



INTELLI 2015

The Fourth International Conference on Intelligent Systems and Applications

ISBN: 978-1-61208-437-4

InManEnt 2015

The International Symposium on Intelligent Manufacturing Environments

October 11 - 16, 2015

St. Julians, Malta

INTELLI 2015 Editors

Ingo Schwab, University of Applied Sciences Karlsruhe, Germany
Leo van Moergestel, Utrecht University of Applied Sciences, The Netherlands
Gil Gonçalves, Faculty of Engineering, University of Porto, Portugal

INTELLI 2015

Forward

The Fourth International Conference on Intelligent Systems and Applications (INTELLI 2015), held between October 11 - 16, 2015 - St. Julians, Malta, continued a series of events on advances towards fundamental, as well as practical and experimental aspects of intelligent systems and applications.

The information surrounding us is not only overwhelming, but is also subject to limitations of systems and applications, including specialized devices. The diversity of systems and the spectrum of situations make it almost impossible for an end-user to handle the complexity of the challenges. Embedding intelligence in systems and applications seems to be a reasonable way to move some complex tasks from user duty. However, this approach requires fundamental changes in designing the systems and applications, in designing their interfaces and requires the use of specific cognitive and collaborative mechanisms. Intelligence becomes a key paradigm and its specific use takes various forms according to the technology or the domain a system or an application belongs to.

The conference had the following tracks:

- Intelligent human-computer interaction systems
- Intelligent applications and systems
- Intelligent agents
- Formal ontology and semantics
- Intelligent robotics
- Hybrid artificial intelligent systems

The conference also featured the following symposium:

- ***InManEnt 2015, The International Symposium on Intelligent Manufacturing Environments***

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the INTELLI 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to INTELLI 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the INTELLI 2015

organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope INTELLI 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of intelligent systems and applications. We also hope that St. Julians, Malta provided a pleasant environment during the conference and everyone saved some time to enjoy the beauty of the city.

INTELLI 2015 Chairs

INTELLI Advisory Committee

Michael Negnevitsky, University of Tasmania, Australia

Roy George, Clark Atlanta University, USA

Pradeep Atrey, University of Winnipeg, Canada

Jerzy Grzymala-Busse, University of Kansas, USA

Daniël Telgen, HU University of Applied Sciences Utrecht, The Netherlands

Zoi Christoforou, Ecole des Ponts-ParisTech, France

Jiho Kim, Chung-Ang University, Korea

Ingo Schwab, Karlsruhe University of Applied Sciences, Germany

Firas B. Ismail Alnaimi, Universiti Tenaga Nasional, Malaysia

Giuseppe Salvo, Università degli studi di Palermo, Italy

Nittaya Kerdprasop, Suranaree University of Technology, Thailand

Susana Vieira, IDMEC/LAETA, Instituto Superior Técnico, Technical University of Lisbon, Portugal

INTELLI Industry/Research Chairs

Antonio Coronato, National Research Council (CNR) & Institute for High-Performance

Computing and Networking (ICAR) - Napoli, Italy

Matjaž Gams, Jožef Stefan Institute - Ljubljana, Slovenia

Haowei Liu, INTEL Corporation, USA

Michael Affenzeller, HeuristicLab, Austria

Paolo Spagnolo, Italian National Research Council, Italy

Pieter Mosterman, MathWorks, Inc. - Natick, USA

Paul Barom Jeon, Samsung Electronics, Korea

Kiyoshi Nitta, Yahoo Japan Research, Japan

Wolfgang Beer, Software Competence Center Hagenberg GmbH, Austria

András Föhrécz, Multilogic Ltd., Hungary

Pierre-Yves Dumas, THALES, France

INTELLI Publicity Chairs

Frederick Ackers, Towson University, USA

Stephan Puls, Karlsruhe Institute of Technology, Germany

Paulo Couto, GECAD - ISEP, Portugal

Yuichi Kawai, Hosei University, Japan

InManEnt 2015 Symposium Co-Chairs

Ingo Schwab, University of Applied Sciences Karlsruhe, Germany

Gil Gonçalves, Faculty of Engineering, University of Porto, Portugal

Juha Röning, University of Oulo, Finland

INTELLI 2015

Committee

INTELLI 2015 Advisory Committee

Michael Negnevitsky, University of Tasmania, Australia
Roy George, Clark Atlanta University, USA
Pradeep Atrey, University of Winnipeg, Canada
Jerzy Grzymala-Busse, University of Kansas, USA
Daniël Telgen, HU University of Applied Sciences Utrecht, The Netherlands
Zoi Christoforou, Ecole des Ponts-ParisTech, France
Jiho Kim, Chung-Ang University, Korea
Ingo Schwab, Karlsruhe University of Applied Sciences, Germany
Firas B. Ismail Alnaimi, Universiti Tenaga Nasional, Malaysia
Giuseppe Salvo, Università degli studi di Palermo, Italy
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Susana Vieira, IDMEC/LAETA, Instituto Superior Técnico, Technical University of Lisbon, Portugal

INTELLI 2015 Industry/Research Chairs

Antonio Coronato, National Research Council (CNR) & Institute for High-Performance Computing and Networking (ICAR) - Napoli, Italy
Matjaž Gams, Jožef Stefan Institute - Ljubljana, Slovenia
Haowei Liu, INTEL Corporation, USA
Michael Affenzeller, HeuristicLab, Austria
Paolo Spagnolo, Italian National Research Council, Italy
Pieter Mosterman, MathWorks, Inc. - Natick, USA
Paul Barom Jeon, Samsung Electronics, Korea
Kiyoshi Nitta, Yahoo Japan Research, Japan
Wolfgang Beer, Software Competence Center Hagenberg GmbH, Austria
András Förhécz, Multilogic Ltd., Hungary
Pierre-Yves Dumas, THALES, France

INTELLI 2015 Publicity Chairs

Frederick Ackers, Towson University, USA
Stephan Puls, Karlsruhe Institute of Technology, Germany
Paulo Couto, GECAD - ISEP, Portugal
Yuichi Kawai, Hosei University, Japan

INTELLI 2015 Technical Program Committee

Syed Sibte Raza Abidi, Dalhousie University - Halifax, Canada
Witold Abramowicz, The Poznan University of Economics, Poland
Michael Affenzeller, HeuristicLab, Australia
Zaher Al Aghbari, University of Sharjah, UAE
Gabor Alberti, University of Pecs, Hungary
Firas B. Ismail Alnaimi, Universiti Tenaga Nasional, Malaysia
Ioannis Anagnostopoulos, University of Thessaly, Greece
Rachid Anane, Coventry University, UK
Andreas S. Andreou, Cyprus University of Technology - Limassol, Cyprus
Ngamnij Arch-int, Khon Kaen University, Thailand
Wudhichai Assawinchaichote, Mongkut's University of Technology -Bangkok, Thailand
Pradeep Atrey, University of Winnipeg, Canada
Paul Barom Jeon, Samsung Electronics, Korea
Daniela Barreiro Claro, Federal University of Bahia, Brazil
Rémi Bastide, Université Champollion, France
Carmelo J. A. Bastos-Filho, University of Pernambuco, Brazil
Bernhard Bauer, University of Augsburg, Germany
Barnabas Bede, DigiPen Institute of Technology - Redmond, USA
Carsten Behn, Ilmenau University of Technology, Germany
Noureddine Belkhatir, University of Grenoble, France
Orlando Belo, University of Minho, Portugal
Petr Berka, University of Economics, Prague, Czech Republic
Félix Biscarri, University of Seville, Spain
Luis Borges Gouveia, University Fernando Pessoa, Portugal
Abdenour Bouzouane, Université du Québec à Chicoutimi, Canada
José Braga de Vasconcelos, Universidade Atlântica, Portugal
Rui Camacho, Universidade do Porto, Portugal
Luis M. Camarinha-Matos, New University of Lisbon, Portugal
Longbing Cao, University of Technology - Sydney, Australia
Sérgio Campello, Escola Politécnica de Pernambuco - UPE, Brazil
Carlos Carrascosa, Universidad Politécnica de Valencia, Spain
Jose Jesus Castro Sanchez, Universidad de Castilla-La Mancha - Ciudad Real, Spain
Marc Cavazza, University of Teesside - Middlesbrough, UK
Kit Yan Chan, Curtin University - Western Australia, Australia
Chin-Chen Chang, Feng Chia University, Taiwan, R. O. C.
Maiga Chang, Athabasca University, Canada
Yue-Shan Chang, National Taipei University, Taiwan
Naoufel Cheikhrouhou, Ecole Polytechnique Fédérale de Lausanne, Switzerland
Rung-Ching Chen, Chaoyang University of Technology, Taiwan
Li Cheng, BII/A*STAR, Singapore
Been-Chian Chien, National University of Tainan, Taiwan
Sunil Choenni, Ministry of Security and Justice, The Netherlands

Byung-Jae Choi, Daegu University, Korea
Antonio Coronato, National Research Council (CNR)& Institute for High-Performance Computing and Networking (ICAR) - Napoli, Italy
Sharon Cox, Birmingham City University, UK
Nora Cuppens, TELECOM Bretagne, France
Arianna D'Ulizia, Research Council - IRPPS, Italy
Chuangyin Dang, City University of Hong Kong, Hong Kong
Suash Deb, IRDO, India
Angel P. del Pobil, Universitat Jaume-I, Spain
Vincenzo Deufemia, Università di Salerno - Fisciano, Italy
Tadashi Dohi, Hiroshima University, Japan
Andrei Doncescu, LAAS-CNRS - Toulouse France
Elena-Niculina Dragoi, "Gheorghe Asachi" Technical University of Iasi, Romania
Marcos Eduardo Valle, University of Campinas, Brazil
Bernard Espinasse, Aix-Marseille Université, France
Shu-Kai S. Fan, National Taipei University of Technology, Taiwan
Aurelio Fernandez Bariviera, Universitat Rovira i Virgili, Spain
Edilson Ferneda, Catholic University of Brasília, Brazil
Manuel Filipe Santos, Universidade do Minho, Portugal
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Juan J. Flores, Universidad Michoacana, Mexico
Gian Luca Foresti, University of Udine, Italy
Rita Francese, Università di Salerno - Fisciano, Italy
Santiago Franco, University of Auckland, New Zealand
Kaori Fujinami, Tokyo University of Agriculture and Technology, Japan
Naoki Fukuta, Shizuoka University, Japan
Matjaž Gams, Jožef Stefan Institute - Ljubljana, Slovenia
Sasanko Sekhar Gantayat, GMR Institute of Technology, India
Leonardo Garrido, Tecnológico de Monterrey - Campus Monterrey, Mexico
Alexander Gelbukh, Mexican Academy of Sciences, Mexico
David Gil, University of Alicante, Spain
Berio Giuseppe, Université de Bretagne Sud, France
Anandha Gopalan, Imperial College London, UK
Sérgio Gorender, UFBA, Brazil
Victor Govindaswamy, University of Texas at Arlington, USA
Manuel Graña, Facultad de Informatica - San Sebastian, Spain
David Greenhalgh, University of Strathclyde, UK
Jerzy Grzymala-Busse, University of Kansas, USA
Christophe Guéret, Free University Amsterdam, The Netherlands
Bin Guo, Northwestern Polytechnical University, China
Sung Ho Ha, Kyungpook National University, Korea
Maki K. Habib, The American University in Cairo, Egypt
Sami Habib, Kuwait University, Kuwait
Belal Haja, University of Tabuk, Saudi Arabia

Sven Hartmann, Technische Universität Clausthal, Germany
Fumio Hattori, Ritsumeikan University - Kusatsu, Japan
Jessica Heesen, University of Tübingen, Germany
Enrique Herrera Viedma, DECSAI - University of Granada, Spain
Pilar Herrero, Universidad Politecnica de Madrid, Spain
Benjamin Hirsch, Khalifa University - Abu Dhabi, United Arab Emirates
Didier Hoareau, University of La Réunion, France
Tetsuya Murai Hokkaido, University Sapporo, Japan
Wladyslaw Homenda, Warsaw University of Technology, Poland
Katsuhiro Honda, Osaka Prefecture University, Japan
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Samuelson Hong, Management School - Hangzhou Dianzi University, China
Bin Hu, Birmingham City University, UK
Yo-Ping Huang, National Taipei University of Technology - Taipei, Taiwan
Carlos A. Iglesias, Universidad Politecnica de Madrid, Spain
Fodor János, Óbuda University – Budapest, Hungary
Jayadeva, Indian Institute of Technology - Delhi, India
Yanguo Jing, London Metropolitan University, UK
Maria João Ferreira, Universidade Portucalense - Porto, Portugal
Diala Jomaa, Dalarna University, Sweden
Janusz Kacprzyk, Polish Academy of Sciences, Poland
Epaminondas Kapetanios, University of Westminster - London, UK
Nikos Karacapilidis, University of Patras - Rion-Patras, Greece
Anthony Karageorgos, Technological Educational Institute of Thessaly, Greece
Panagiotis Karras, Rutgers University, USA
Jung-jae Kim, Nanyang Technological University, Singapore
Sungshin Kim, Pusan National University- Busan, Korea
Abeer Khalid, International Islamic University Islamabad, Pakistan
Shubhalaxmi Kher, Arkansas State University, USA
Alexander Knapp, Universität Augsburg, Germany
Sotiris Kotsiantis, University of Patras, Greece
Ondrej Krejcar, University of Hradec Kralove, Czech Republic
Natalia Kryvinska, University of Vienna, Austria
Satoshi Kurihara, Osaka University, Japan
K.P. Lam, University of Keele, UK
Antonio LaTorre, Universidad Politécnica de Madrid, Spain
Kennerd Lavers, Air Force Institute of Technology - Wright-Patterson
Frédéric Le Mouël, INRIA/INSA Lyon, France
Alain Léger, Orange - France Telecom R&D / University St Etienne - Betton, France
George Lekeas, City Universty – London, UK
Omar Lengerke, Autonomous University of Bucaramanga, Colombia
Carlos Leon, University of Seville, Spain
Haowei Liu, INTEL Corporation, USA
Abdel-Badeeh M. Salem, Ain Shams University - Cairo, Egypt

Giuseppe Mangioni, University of Catania, Italy
Antonio Martin, Universidad de Sevilla, Spain
Gregorio Martinez, University of Murcia, Spain
George Mastorakis, Technological Educational Institute of Crete, Greece
Constandinos X. Mavromoustakis, University of Cyprus, Cyprus
Pier Luigi Mazzeo, Institute on Intelligent System for Automation - Bari, Italy
Michele Melchiori, Università degli Studi di Brescia, Italy
Radko Mesiar, Slovak University of Technology Bratislava, Slovakia
John-Jules Charles Meyer, Utrecht University, The Netherlands
Angelos Michalas, TEI of Western Macedonia, Greece
Veronica S. Moertini, Parahyangan Catholic University, Indonesia
Dusmanta Kumar Mohanta, Birla Institute of Technology - Mesra, India
Felix Mora-Camino, ENAC, Toulouse, France
Fernando Moreira, Universidade Portucalense - Porto, Portugal
Pieter Mosterman, MathWorks, Inc. - Natick, USA
Bernard Moulin, Université Laval, Canada
Debajyoti Mukhopadhyay, Maharashtra Institute of Technology, India
Isao Nakanishi, Tottori University, Japan
Tomoharu Nakashima, Osaka Prefecture University, Japan
Nayyab Zia Naqvi, iMinds - Distrinet | KU Leuven, Belgium
Michael Negnevitsky, University of Tasmania, Australia
Filippo Neri, University of Naples "Federico II", Italy
Mario Arrigoni Neri, University of Bergamo, Italy
Hongbo Ni, Northwestern Polytechnical University, China
Cyrus F. Nourani, akdmkrd.tripod.com, USA
Kenneth S. Nwizege, Swansea University, UK
Joanna Isabelle Olszewska, University of Gloucestershire, United Kingdom
Hichem Omrani, CEPS/INSTEAD Research Institute, Luxembourg
Frank Ortmeier, Otto-von-Guericke Universitaet Magdeburg, Germany
Jeng-Shyang Pan, Harbin Institute of Technology, Taiwan
Endre Pap, University Novi Sad, Serbia
Marcin Paprzycki, Systems Research Institute / Polish Academy of Sciences - Warsaw, Poland
Dana Petcu, West University of Timisoara, Romania
Leif Peterson, Methodist Hospital Research Institute / Weill Medical College, Cornell University, USA
Diego Pinheiro-Silva, University of Pernambuco, Brazil
Alain Pirott, Université de Louvain - Louvain-la-Neuve, Belgium
Agostino Poggi, Università degli Studi di Parma, Italy
Radu-Emil Precup, Politehnica University of Timisoara, Romania
Anca Ralescu, University of Cincinnati, USA
Sheela Ramanna, University of Winnipeg, Canada
Fano Ramparany, Orange Labs Networks and Carrier (OLNC) - Grenoble, France
Martin Randles, Liverpool John Moores University, UK
Zbigniew W. Ras, University of North Carolina - Charlotte & Warsaw University of Technology,

Poland

José Raúl Romero, University of Córdoba, Spain
Danda B. Rawat, Georgia Southern University, USA
David Riaño, Universitat Rovira i Virgili, Spain
Daniel Rodríguez, University of Alcalá - Madrid, Spain
Agos Rosa, Technical University of Lisbon, Portugal
Alexander Ryzhov, Lomonosov Moscow State University, Russia
Gunter Saake, University of Magdeburg, Germany
Ozgur Koray Sahingoz, Turkish Air Force Academy, Turkey
Shigeaki Sakurai, Toshiba Corporation, Japan
Demetrios G. Sampson, University of Piraeus, Greece
Daniel Schang, Groupe Signal Image et Instrumentation - ESEO, France
Ingo Schwab, Karlsruhe University of Applied Sciences, Germany
Florence Sedes, IRIT | Université de Toulouse, France
Amal El Fallah Seghrouchni, University of Pierre and Marie Curie (Paris 6) - Paris, France
Hirosato Seki, Kwansei Gakuin University, Japan
Nikola Serbedzija, Fraunhofer FOKUS, Germany
Timothy K. Shi, National Central University, Taiwan
Kuei-Ping Shih, Tamkang University - Taipei, Taiwan
Choonsung Shin, Carnegie Mellon University, USA
Marius Silaghi, Florida Institute of Technology, USA
Peter Sincák, Technical University of Kosice, Slovakia
Spiros Sirmakessis, Technological Educational Institute of Messolonghi, Greece
Alexander Smirnov, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS), Russia
João Miguel Sousa, Universidade de Lisboa, Portugal
Paolo Spagnolo, Italian National Research Council, Italy
Chrysostomos Stylios, Technological Educational Institute of Epirus, Greece
Adel Taweel, King's College London, UK
Abdel-Rahman Tawil, University of East London, UK
Olivier Terzo, Istituto Superiore Mario Boella (ISMB), Italy
I-Hsien Ting, National University of Kaohsiung, Taiwan
Federico Tombari, University of Bologna, Italy
Anand Tripathi, University of Minnesota Minneapolis, USA
Juan Carlos Trujillo Mondéjar, University of Alicante, Spain
Scott Turner, University of Northampton, UK
Theodoros Tzouramanis, University of the Aegean, Greece
Leo van Moergestel, Utrecht University, Netherlands
Gantcho Vatchkov, University of the South Pacific (USP) in Suva, Fiji Island
Jan Vascak, Technical University of KoSice, Slovakia
Jose Luis Vazquez-Poletti, Universidad Complutense de Madrid, Spain
Mario Vento, Università di Salerno - Fisciano, Italy
Dimitros Vergados, Technological Educational Institution of Western Macedonia, Greece
Nishchal K. Verma, Indian Institute of Technology Kanpur, India

Susana Vieira, University of Lisbon, Portugal
Mirko Viroli, Università di Bologna - Cesena, Italy
Mattias Wahde, Chalmers University of Technology - Göteborg, Sweden
Fang Wang, Brunel University London, UK
Yan Wang, Macquarie University - Sydney, Australia
Zhihui Wang, Dalian University of Technology, China
Viacheslav Wolfengagen, Institute "JurInfoR-MSU", Russia
Mudasser F. Wyne, National University - San Diego, USA
Guandong Xu, Victoria University, Australia
WeiQi Yan, Queen's University Belfast, UK
Chao-Tung Yang, Tunghai University - Taichung City, Taiwan, R.O.C.
George Yee, Carleton University, Canada
Hwan-Seung Yong, Ewha Womans University - Seoul, Korea
Paul D. Yoo, Khalifa University of Science, Technology and Research (KUSTAR), UAE
Slawomir Zadrozny, Systems Research Institute - Polish Academy of Sciences, Poland
Hao Lan Zhang, NIT - Zhejiang University, China
Si Q. Zheng, The University of Texas at Dallas, USA
Jose Jacobo Zubcoff Vallejo, University of Alicante, Spain

InManEnt 2015 Symposium Co-Chairs

Ingo Schwab, University of Applied Sciences Karlsruhe, Germany
Gil Gonçalves, Faculty of Engineering, University of Porto, Portugal
Juha Röning, University of Oulo, Finland

InManEnt 2015 Program Committee Members

Dirk Berndt, Fraunhofer IFF, Germany
Eisse Jan Drewes, AWL, Netherlands
Michael Emmerich, University of Leiden, The Netherlands
Björn Hein, University of Karlsruhe, Germany
Adel Hejaaji, ESM LTD Essex, UK
Martin Kasperczyk, Fraunhofer IPA, Germany
Norbert Link, University of Applied Sciences Karlsruhe, Germany
Niels Lohse, Loughborough University, UK
Giorgio Pasquettaz, CRF, Italy
Marcello Pellicciari, University of Modena and Reggio Emilia, Italy
Marius Pflueger, IPA, Germany
Franz Quint, University of Applied Sciences Karlsruhe, Germany
João Reis, Faculty of Engineering, University of Porto, Portugal
Steffen Scholz, Institute for Applied Computer Science/Karlsruhe Institute of Technology, Germany
Vassilis Spais, Inos Hellas, Greece
Leo van Moergestel, Utrecht University of Applied Sciences, The Netherlands

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Modelling Communicative Space.From Human Communication to Conversational Agents <i>Mare Koit and Haldur Oim</i>	1
Low-level Automation as a Pathway to Appropriate Trust in an Intelligent PED Enterprise: Design of a Collaborative Work Environment <i>Michael Jenkins, Arthur Wollocko, Martin Voshell, and Mike Farry</i>	6
Evaluation of Visual Impression of Delayed Movement of Avatar while Exercising <i>Taeko Tanaka, Hiroshi Hashimoto, and Sho Yokota</i>	10
On the Robustness of Regression Type Classifiers <i>Olgierd Hryniewicz</i>	16
Bio-inspired Design of High-speed Transmission Line <i>Moritoshi Yasunaga and Ikuo Yoshihara</i>	23
Evaluation and Monitoring for Disaster Management <i>Alexander Ryjov</i>	26
G-Form: A New Approach for Visual Interpretation of Deep Web Form as Galaxy of Concepts <i>Radhouane Boughammoura, Lobna Hlaoua, and Mohamed Nazih Omri</i>	32
Experimental Analysis of Black Virus Decontamination by DisJ <i>Jie Cai</i>	40
Multi-Agent Technology in Real-time Intelligent Resource Management Systems <i>Igor Mayorov and Petr Skobelev</i>	49
Modeling the Dynamics of Insulin-Glucose Subsystem Using a Multi-agent Approach Based on Knowledge Communication <i>Sebastian Meszynski and Oleksandr Sokolov</i>	55
Multicast Routing for High-Quality Multimedia Environments: Deployment and New Problems <i>Pavel Troubil, Hana Rudova, and Petr Holub</i>	61
An Iterative Method for Enhancing Text Comprehension by Automatic Reading of References <i>Amal Babour, Fatema Nafa, and Javed Khan</i>	66
Crowdsourcing-Based Multi-Layer Automated Ontology Matching: An approach and Case Study <i>Alexander Smirnov, Nikolay Shilov, Nikolay Teslya, and Alexey Kashevnik</i>	74

Object-Oriented Communication Model for an Agent-Based Inventory Operations Management <i>Rafal Cupek, Adam Ziebinski, Lukasz Huczala, Daniel Grossmann, and Markus Bregulla</i>	80
Granular Meta-Ontology and Extended Allen's logic: Some Theoretical Background and Application to Intelligent Product Lifecycle Management Systems <i>Valery B. Tarassov, Alena V. Fedotova, Rainer Stark, and Baurzhan S. Karabekov</i>	86
A Lightweight Simulator for Autonomous Driving Motion Planning Development <i>Tianyu Gu and John Dolan</i>	94
A NAO-based Intelligent Robotic System for a Word Search-like Game <i>Victor Lobato-Rios, Angelica Munoz-Melendez, and Jose Martinez-Carranza</i>	98
LQG Control of a Two-Wheeled Mobile Pendulum System <i>Akos Odry, Ervin Burkus, and Peter Odry</i>	105
Estimation of Nuclear Reactor Vessel Water Level in Severe Accidents Using Cascaded Fuzzy Neural Networks <i>Man Gyun Na, Dong Yeong Kim, Kwae Hwan Yoo, and Geon Pil Choi</i>	113
Prediction of Golden Time Using SVM for Recovering SIS in Severe Post-LOCA Circumstances <i>Man Gyun Na, Kwae Hwan Yoo, Dong Yeong Kim, and Ju Hyun Back</i>	118
Automatic Trigger Speed for Vehicle Activated Signs using Adaptive Neuro fuzzy system and ClassificationRegression Trees <i>Diala Jomaa, Siril Yella, and Mark Dougherty</i>	124
Process Chain Optimization using Universal State and Control Features <i>Melanie Senn, Ingo Schwab, and Norbert Link</i>	126
Application of Tast-to-Method Transform to Laser Seam Welding <i>Jurgen Pollak</i>	128
Globally Optimized Production by Co-operating Production Agents Based on Bellmans Principle <i>Norbert Link</i>	134
SMARTLAM - A Modular, Flexible, Scalable, and Reconfigurable System for Manufacturing of Microsystems <i>Steffen Scholz, Matthias Plasch, Hannes Limbeck, Tobias Iseringhausen, Markus Dickerhof, Andreas Schmidt, Tobias Muller, and Christian Woegerer</i>	140
Process State Observation Using Artificial Neural Networks and Symbolic Regression <i>Susanne Fischer</i>	142
Efficient Implementation of Network-enabled Devices into Industrial Environment	148

Martin Kasperczyk and Eileen Ridders

Optimizing Product Paths in a Production Grid <i>Leo van Moergestel, Erik Puik, Daniel Telgen, and John-Jules Meyer</i>	150
Self-organising Smart Components in Advanced Manufacturing Systems <i>Rui Pinto, Joao Reis, Ricardo Silva, Vitor Sousa, and Gil Goncalves</i>	157
Self-Diagnosis and Automatic Configuration of Smart Components in Advanced Manufacturing Systems <i>Rui Pinto, Joao Reis, Vitor Sousa, Ricardo Silva, and Gil Goncalves</i>	164
Comparing Knowledge Representation Forms in Empirical Model Building <i>Hao Wang, Ingo Schwab, and Michael Emmerich</i>	170
Test Platform for the Performance Evaluation of OPC-UA Servers for Fast Data Transfer Between Intelligent Equipment <i>Flavio Gonzalez Vazquez</i>	179

Modelling Communicative Space

From human communication to conversational agents

Mare Koit

Institute of Computer Science
University of Tartu
Tartu, Estonia
email: mare.koit@ut.ee

Haldur Õim

Institute of Estonian and General Linguistics
University of Tartu
Tartu, Estonia
e-mail: haldur.oim@ut.ee

Abstract—The paper introduces a work in progress on modelling one aspect of natural human communication – communicative space. Participants of a communication event place themselves at certain points of communicative space which characterizes such relevant features of their relations in the event as communicative distance, dominance, politeness, personal relatedness, etc. Examples of human-human dialogues demonstrate how participants pass different points in communicative space during an event. Our aim is to include such a model of communicative space in our experimental system for modelling conversational agents in order to make interaction with the system more human-like.

Keywords—*dialogue; communicative space; human-human communication, human-computer interaction, conversational agents.*

I. INTRODUCTION

Communication between people can take various forms depending on a lot of circumstances – participants’ individual characteristics, their social roles, subject of conversation, etc. When a human talks with other people, (s)he evaluates them not only from their words but also from their facial expression, body movement, and gestures. These nonverbal aspects help to convey the ‘tone’ of the conversation [9]. The lessons learnt from the study of human-human communication can be used when modelling interaction with the computer. Different features have to be taken into account in order to make it possible for a user to interact with the computer in a natural way, i.e., in a natural language and following norms and regulations of human-human communication.

There is one relevant aspect of human communication, which we will center on – communicative space [10]. For a general description see the work in [3].

Healey et al. [6] declare that “there are important differences in the quality of human interaction – in degrees of interpersonal, as opposed to physical, closeness – that are important for the organization of human activities and, consequently, for design”. The concept of communicative space provides a useful approach to thinking about the basic organization of human interaction.

Communicative space is a mental space where a communication participant places himself/herself with respect to other ones and where (s)he is ‘moving’ during a communication event. Communicative space can be

characterized by different features, e.g., (social) closeness of a communication participant with the partner, collaboration, politeness, etc. These features of communication can be conveyed by language use as well as by different nonverbal means (body movement, facial expressions, etc.). Still, in this paper we limit ourselves with considering interaction in natural language, without taking into account nonverbal means. In order to model communicative space, we analyze transcripts of human-human spoken dialogues.

The paper is structured as follows. Section 2 introduces a model of communicative space. Section 3 analyzes some dialogue examples, which demonstrate different points in communicative space. Section 4 discusses how communicative space can be used when developing a dialogue system and Section 5 draws conclusions.

II. DIMENSIONS OF COMMUNICATIVE SPACE

Communicative space can be specified by a number of dimensions that characterize the relationships of participants in a communicative encounter. Communication can be collaborative or confrontational, personal or impersonal; it can also be characterized by the social distance of participants (near, far), by the modality of communication (friendly, hostile), by the intensity (peaceful, vehement), etc. [10]. Together, these dimensions bring the social aspect of communication into the model [2]. They represent a subsystem of human communicative competence with deep evolutionary roots, the basic function of which is to regulate the communication process. People have an intuitive, ‘naïve’ theory of these coordinates; the values of the coordinates can be expressed by specific words [11]. Instead, at present we use numerical values as approximations in our model.

We determine communicative space as an n -dimensional ($n > 0$) space with the following coordinates:

- communicative distance to the partner (on the scale from familiar to remote)
- cooperation (on the scale from collaborative to confrontational)
- politeness (from polite to impolite)
- personality (from personal to impersonal)
- modality (from friendly to hostile)
- intensity (from peaceful to vehement).

The concept of communicative space is thus related to approaches where the concept of social attitude or

interpersonal stance in interaction (e.g., being polite, distant, cold, warm, supportive, contemptuous) are dealt with [4], or where the interaction space is represented by two dimensions – dominance and liking [17].

The social role of a communication participant (e.g., boss vs subordinate, salesman vs customer, etc.) influences the choice of a point in communicative space. For example, we usually expect that a salesman politely, impersonally and peacefully interacts with a customer at the same time when some customers are impolite, hostile and vehement. A communicative distance is small between friends but it is big between adversaries, etc.

We use the numbers +1, 0 and -1 for the values of the coordinates of communicative space. For example, the value +1 on the scale of intensity means peaceful and the value -1 means vehement interaction. Communicative distance is +1 if a person is close to his or her communication partner and -1 if (s)he is far from the partner. 0 is the neutral value on any scale. Still, it would be possible to consider a bigger number of values on every scale.

It is especially important to stress two moments related to the location(s) of participants in the communicative space during a communication event. First, the participants can be located at different points of communicative space. For example, a good clerk remains polite also when communicating with an impolite customer; one communication participant can feel closeness to his/her partner whereas the partner has different feeling, etc. Secondly, the participants can also ‘move’ from one point to another during the encounter. For instance, conversation, which started peacefully can become vehement, or *vice versa*; participants who were on confrontational positions at the outset can reach the collaborative one at the end (and also *vice versa*), etc. It is just this latter moment where the function of communicative space as a regulator and reflector of the dynamics of communication encounters as social events reveals itself (see Section 3, comments to Figure 1).

III. PEOPLE IN COMMUNICATIVE SPACE: EXAMPLES

With the aim to model human-computer interaction we start with considering human-human communication. Where do people place themselves in communicative space when communicating and how do they ‘move’ there? We are especially interested in linguistic means, which help us to recognize the points of communicative space on the basis of texts of communication participants in a natural language. Let us consider some examples from the Estonian dialogue corpus [7]. The corpus includes mainly audio recordings of human-human dialogues in authentic situations, which are transcribed by using a transcription of Conversation Analysis [8]. Each transcription is provided with a header that lists situational factors, which affect language use, e.g., participants names, social characteristics, relations between participants in the situation, specification of situation (private/public place, private/institutional conversation), etc. We will present examples of two types of conversations: institutional (more concretely, conversations with an information clerk) and everyday conversations between acquaintances.

A. Institutional Conversation

Let us start with considering transcripts of directory inquiries. Customers call an answering service and request some information (phone numbers, addresses, institution names, etc.). A clerk (answerer) is an official person and she has to place herself at a certain point in communicative space: to keep a neutral communicative distance, to be polite (or neutral but not impolite), collaborative (or neutral but not antagonistic), etc. Customers have more freedom. In the following examples, *A* is a customer and *B* is a clerk.

Transcription marks used in the examples can be found in [7]. Let us only point out that a number in parentheses marks duration of a pause, e.g., (3.5) marks a break for 3.5 seconds; (.) marks a micro-pause with duration of 0.2 seconds or less. Comments are given in double parentheses.

When annotating the points of communicative space we present the values of the coordinates in the following order: 1) communicative distance, 2) cooperation, 3) politeness, 4) personality, 5) modality, 6) intensity. All the values can be +1, 0 or -1.

In the first example (1), a customer requests a phone number and a clerk gives it him. Both the customer and the clerk have chosen the same communication point (0,0,0,0,0,0) – the values of all coordinates are 0 (neutral).

(1)

A: paluks Asa `kindlustuse `Tartu `osakonda telefoni`numbrit.

I'd like to get a phone number of Asa insurance in Tartu (0,0,0,0,0,0)

(3.5)

B: neli kolm `kaks (.) `seitse kuus `iiks.

Four three two seven six one (0,0,0,0,0,0)

In the case if a clerk does not have the requested information in her data base, she can express her emotion (regret) when answering like in example 2. The value on the scale of cooperation is +1.

(2)

B: sellist `baari ei ole `antud meie andmebaasi kahjuks.
Sorry, we don't have such bar in our data base (0,+1,0,0,0,0)

Still, we can concentrate also on one single dimension. Example 3 demonstrates, how the values are changing during a conversation on the scale of collaboration (the second coordinate). The comments start with '//'. Adjacency pairs of utterances [8] are numerated.

(3)

/--/

1 A: ma paluks `Maarjamõisa `kõõki

May I get the kitchen in Maarjamõisa? // a neutral information request; communication point (0,0,0,0,0,0)

(2.2)

B: `haigla juures või poli`kliinikus.

In the hospital or outpatients' office? // the clerk expresses cooperation by asking an adjusting question; (0,+1,0,0,0,0)

2 A: `haigla.

The hospital // the customer similarly expresses cooperation by giving information; (0,+1,0,0,0)

(0.5)

B: *e`kõogi numbrit ei ole meil`antud.*

We don't have the phone number of the kitchen // the clerk refuses to give information; (0,-1,0,0,0)

(0.5)

3 A: *sääl neil`on telefon peal.*

But they have a phone there // the client expresses protest; (0,-1,0,0,0)

B: *jah nendel võib`olla, aga meil ei ole`antud kõögi`numbrit.*

Yes they may have but we don't have the phone number of the kitchen // the clerk again refuses, and she is excited; (0,-1,0,0,-1)

(0.5)

4 B: *ma saan teile`anda`üldinfo`numbri.*

I can give you the information number // the clerk expresses cooperation, proposing information; (0,+1,0,0,0)

(1.0)

5 A: (1.0) *jah, (.)`olge pai,`andke.*

Yes, be so kind as to give it // the client accepts cooperation, he answers friendly and personally; (0,+1,+1,0,+1,0)

B: *neli neli kaheksa,*

Four four eight // the clerk gives information; (0,+1,0,0,0,0)

/---/

In example 3, the clerk is moving from one communication point to another when answering. Plural form of the pronoun 'you' (in Estonian *teile/you* [plural] vs *sulle/you* [singular]) indicates politeness (value +1).

When analyzing the dialogues with the same clerk we can draw her 'portrait' taking into account the communication points she passes in communicative space. We evaluate an information provider as a good clerk if she keeps neutral values of coordinates or at least avoids negative values and she is collaborative in the sense that if she doesn't have the data requested by a customer then she attempts to offer substituting information like in example 3.

Similarly, we can draw the 'portrait(s)' of a participant (or both participants) of a dialogue regarding any coordinate of communicative space. Figure 1 represents the 'portraits' of A and B in relation to collaboration in conversation (Example 3).

In the same way, it is possible to analyze and compare the changes of values of more than one (selected) dimensions, during a certain dialogue encounter or in some interaction type in general, e.g., in order to investigate possible 'dependency patterns' between different dimensions in different kinds of interaction types. That is, it is possible not only to fix general (static) relationships between dimensions but also to establish dynamic patterns of changes of the values of certain dimensions in the development of communicative encounters we are interested in (e.g., formal negotiations vs buying-selling situations vs quarrels about personal matters).

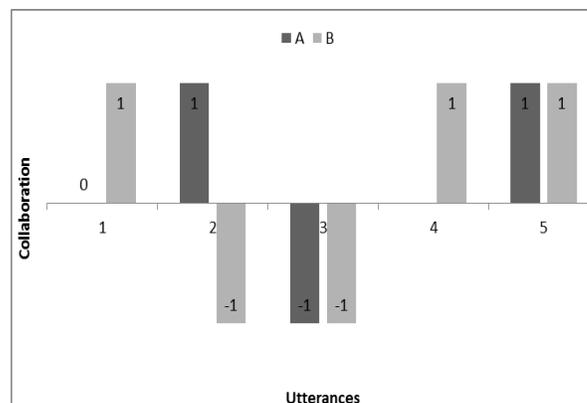


Figure 1. The 'portraits' of A and B on the scale of collaboration (Example 3; the values on the scale are -1, 0 or +1).

B. Everyday Conversation

Everyday conversation is different as compared with institutional conversation. The participants are not obliged to stay in a certain communication point predetermined by their role in the communication event but the values of coordinates may vary in a wide range.

In the following examples, the participants are friends (the value of communicative distance is fixed as +1).

In example 4, A makes a proposal but B doesn't agree and answers angrily as indicated by a comment (in double parentheses). The value on the personality scale is +1 because singular imperative is used (*helista/call* [singular imperative] vs *helistage/call* [plural imperative]).

(4)

A: *'helista`talle.*

Call [singular] him (+1,0,0,+1,0,0)

(.)

B: *helista`ise. ((angrily))*

Call [singular] yourself (+1,0,-1,+1,-1,-1)

In example 5, A expresses protest. The value on the personality scale is +1 because singular is used (*sa käisid kolamas/you have nosed* [singular] vs *te käisite kolamas/you have nosed* [plural]). The values are -1 on the modality and intensity scales.

(5)

A: *se=tändab`seda et sa käisid minu`sahtlites`kolamas.*

It means that you have nosed [singular] around my lockers (+1,0,0,+1,-1,-1)

In example 6, A is surprisingly calling his girlfriend B. The comments (in double parentheses) help to follow the 'tone' of the conversation.

(6)

B: *'tsau musi. ((surprised))*

Ciao darling (+1,0,0,+1,+1,0)

(0.6)

A: *tsau ((dearly))*

Ciao (+1,0,0,+1,+1,+1)

/---/

tuled mulle`külla=vä.

Do you [singular] come to me (+1,0,0,0,+1,+1)

B: mmmmmmm=> ma=i 'saa {praegult} 'tulla. < ((apologizing))

I can't come at the moment (+1,-1,0,0,+1,0)

After presenting these examples, which represent typical data we are working with, and before proceeding to the discussion of specific aspects and problems of our model, let us make here a general comment concerning the model. The properties of communicative space make it possible to represent agent's intellectual states, by changing the values of dimensions during a communication event.

In order to control these properties when modelling real interaction, one needs to take into account also the emotional aspect of communication, i.e., to relate the model of communicative space somehow to the emotional models. This need is especially clear in the case of everyday informal interactions (as one can see also in the case of above examples). There exist some approaches using emotional models. Thus, communication-driven models select an emotional display for its communicative effect. Simulation-based approaches simulate aspects of emotion processes, essentially giving the agent true emotions [5]. But in trying to incorporate such emotional models into a general model of communicative space (in our sense) several critical problems arise, first of all, the problem of delimiting the concept of 'emotional aspect' in this context. In human communication, it includes not only 'pure' feelings and moods, but also attitudes, opinions, (psychological/social) dispositions and stances, which involve also intellectual component in the sense that they can be debated about by using rational arguments. In our present model of communicative space these 'emotional aspects' are implicitly accounted for by different dimensions (modality, politeness, intensity); see also the next Section. But making their role and interdependencies explicit in different types of communicative interaction needs more investigations into deeper levels of human motivational sphere.

IV. DISCUSSION

We are using the values +1, 0, and -1 for the coordinates in communicative space. Actually, all scales could be divided into a bigger number of values and – as said before – a word in a natural language can be used for every value. For example, modality of communication can be *friendly*, *ironic*, *hostile*, etc. Still, the words can be substituted with numbers in the model as we do.

The dimensions that we are using for characterization of communicative space are not fully independent on each other. For example, the length of communicative distance is related to personality – a shorter distance implies a bigger value on the personality scale; the impoliteness implies small values of the modality and intensity, etc. Further research is needed in order to elaborate the list of dimensions (some dimensions could be removed and new dimensions added) and the borderlines between different dimensions and different values.

Of course, there are other possible approaches than the one offered by our model, which operates with a predefined system of dimensions and their values (even when not independent). For instance, the reasoning methods in fuzzy rule-based classification systems have been studied by Mesiarová-

Zemánková [13][14]. Such systems deal with noisy, imprecise, or incomplete information while keeping a satisfactory level of approximation and a good interpretability of the system. It is shown that reasoning methods and derivation of fuzzy rule consequents are based on multipolar aggregation operators. Nevertheless, such an approach hardly suits for interpreting dialogical texts with the aim of recognizing the covert intentions of the interacting agents and explaining the choice of current turns and their verbal realizations made in a concrete communication situation.

There are some linguistic keys, which help to recognize some parameters of communicative space. For example, if a person uses singular form of a verb or of a pronoun (in Estonian *sa/you* [singular] vs *te/you* [plural]) in his/

her utterance addressed to the partner then it indicates a short communicative distance (value +1) and a big personality (value +1) like in Examples 4 to 6. When communicating with an unfamiliar person Estonians usually use the plural form (Example 3). Still, young people are discarding this tradition.

Feeling words can signal some values, e.g., *please* and *thank* indicate politeness. Some research has been done for detection of emotions in Estonian texts using both lexicon-based and statistical methods [1][15].

Comments in transcripts of spoken recordings can help to determine the 'tone' of conversation like in Example (6).

In order to do automatic recognition of values of coordinates of communicative space, opinion (or sentiment) analysis can be used, which allows to determine the contextual polarity of a text [16]. However, this line of investigations remains for the further research.

Our examples demonstrated that people behave differently in different situations. Dialogue participants have different expectations when communicating, e.g., with a near friend or with an official person. The path covered by a communication participant in communicative space characterizes his/her attitudes regarding the partner.

How to use the notion of communicative space when developing human-computer dialogue systems? A number of possible applications could be offered but here we will simply point at two general directions of research where the need for some kind of such conceptual mechanism should be obvious.

First, the systems can be created, which analyze the protocols of certain interaction sessions and 'reconstruct' the placements of participants and the changes of the locations during a session. The second direction (chosen also by us) is more interesting for research. It is related to intelligent agents with two constraints. First, such agents are interacting with human users in a natural language, and secondly, they are planned to play a certain 'social role' in interaction. Such interaction systems have been created for a long time but the aspect of their social role has usually not been explicated. The conversational agents [12] and especially, the conversational characters, which have recently become popular, take into account only the features of a limited field (e.g., a virtual guide of an art exhibition). At the same time, the agents can be created, which could be 'tuned' to behave according to certain locations in communicative space depending on the user. For example, a travel agent gives information about a trip but it can also add various advices being neutral, advertising or even

intrusive. An advisor system in negotiation can take in the coordinates of the location of the user in communicative space related to his/her partner and then recommend suitable data to use (facts, arguments).

V. CONCLUSION AND FURTHER WORK

We analyze human-human dialogues with the aim to develop a dialogue system, which interacts with a user in a natural language following norms and rules of human communication. This paper considers a model of communicative space – a mental space where conversation participants are situated and where they are ‘moving’ during a conversation. We are modelling communicative space as an n -dimensional space with such dimensions as communicative distance of a participant to his/her partner, cooperation, politeness, personality, modality, and intensity. We assign the values +1, 0, or -1 to the coordinates. The analysis of human-human dialogues demonstrates how different points in communicative space are visited during conversation. Using the path covered by a communication participant in communicative space we can create his/her ‘portrait’ and implement it in a dialogue system.

We have implemented an experimental conversational agent, which argues for doing an action interacting with the user in written Estonian. We believe that including the model of communicative space into the system will make the interaction more natural. This remains for the further work.

ACKNOWLEDGMENT

This work is supported by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS) and the Estonian Research Council (grant IUT20-56).

REFERENCES

- [1] R. Altrov, “Aspects of cultural communication in recognizing emotions,” *Trames: Journal of the Humanities and Social Sciences*, 2013, 17(67/62), pp. 159–174.
- [2] G. Boella and L. van der Torre, “BDI and BOID argumentation,” *Proc. of CMNA-03. The 3rd workshop on computational models of natural argument at IJCAI-2003, Acapulco*, 2003, 4p. Available from www.cmna.info [retrieved: August, 2015]
- [3] P. Brown and S. Levinson, “Universals in language usage: Politeness phenomena,” E. Goody (Ed.), *Questions and politeness: Strategies in social interaction*, pp. 56–289, Cambridge: Cambridge University Press, 2008.
- [4] V. Carofiglio, B. De Carolis, I. Mazzotta, N. Novielli, and S. Pizzutilo, “Towards a Socially Intelligent ECA,” *IxD&A*, 2009, 5-6, pp. 99-106.
- [5] J. Gratch and S. Marsella, “Some Lessons from Emotion Psychology for the Design of Lifelike Characters”, *Applied Artificial Intelligence*, 2005, 19, 35 pp.
- [6] P. G. T. Healey, G. White, A. Eshghi, A. J. Reeves, and A. Light, “Communication spaces,” *Computer Supported Cooperative Work*, 2008, 17, pp.169–193, Springer. doi: 10.1007/s10606-007-9061-4
- [7] T. Hennoste, O. Gerassimenko, R. Kasterpalu, M. Koit, A. Rääbis, and K. Strandson, “From human communication to intelligent user interfaces: Corpora of spoken Estonian,” *Proc. of the sixth international language resources and evaluation (LREC’08), European Language Resources Association (ELRA), Marrakech, Morocco*, 2008, pp. 2025–2032. Available from www.lrec-conf.org/proceedings/lrec2008 [retrieved: August, 2015]
- [8] I. Hutchby and R. Wooffitt, *Conversation Analysis*, Polity Press, Cambridge, 2008.
- [9] M. Knapp and J.A. Hall, *Nonverbal Communication in Human Interaction*. (5th ed.) Wadsworth: Thomas Learning, 2007. ISBN 0-15-506372-3
- [10] M. Koit, “Communicative strategy in a formal model of dispute,” *Proc. of the International Conference on Agents and Artificial Intelligence: 7th International Conference on Agents and Artificial Intelligence (ICAART)*, 2015, pp. 489–496. Lisbon, Portugal, SciTePress.
- [11] M. Koit and H. Õim, “A computational model of argumentation in agreement negotiation processes,” *Argument & Computation*, 2014, 5 (2-3), pp. 209–236. Taylor & Francis Online. doi: 10.1080/19462166.2014.915233
- [12] J. Lester, K. Branting, and B. Mott, “Conversational Agents”, *The Practical Handbook of Internet Computing*, Chapman & Hall, 2004.
- [13] A. Mesiarová-Zemánková, “Sensitivity analysis of fuzzy rule-based classification systems by means of the Lipschitz condition”, *Soft Comput.*, 2015, 9 pp. Published Online 20 June 2015, Springer-Verlag Berlin Heidelberg. doi: 10.1007/s00500-015-1744-z
- [14] A. Mesiarová-Zemánková, “Multipolar Aggregation Operators in Reasoning Methods for Fuzzy Rule-Based Classification Systems”, *IEEE Transactions On Fuzzy Systems*, 2014, vol. 22, No 6, pp. 1569-1584. doi: 10.1109/TFUZZ.2014.2298878
- [15] H. Pajupuu, K. Kerge, and R. Altrov, “Lexicon-based detection of emotion in different types of texts: preliminary remarks,” *Eesti Rakenduslingvistika Ühingu aastaraamat = Estonian Papers in Applied Linguistics*, 2012, 8, pp. 171–184.
- [16] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval*, 2008, vol. 2, No 1-2, pp. 1–135.
- [17] B. Ravenet, M. Ochs, and C. Pelachaud, “A computational model of social attitude effects on the nonverbal behavior for a relational agent,” *Proc. of WACAI, Grenoble, France*, 2012, pp. 94–101.

Low-level Automation as a Pathway to appropriate Trust in an Intelligent PED Enterprise: Design of a Collaborative Work Environment

Michael P. Jenkins, Arthur Wollocko, Martin Voshell, & Mike Farry

Charles River Analytics, Inc.

Cambridge, MA USA

Email: mjenkins@cra.com

Abstract—In Military Intelligence, Processing, Exploitation, and Dissemination (PED) functions are critical to success. These functions provide an array of capabilities that support the entire lifecycles of intelligence requests. Advanced PED capabilities are becoming increasingly available to smaller, more-centralized teams supporting multiple battlespace operators. As the PED domain evolves and more distributed information requests are made relying on an increasing volume of Multiple-Intelligence (MultiINT) information, automation support has become critical to success. However, automated support and cognitive incongruence between existing automated solutions and the support required by analysts, resulting in a lack of trust in these “black box” capabilities. To overcome this and other current and future PED challenges, we present a Collaborative Work Environment, serving as a central software platform providing communication channels and tailored workflow support tools for PED operations. Integrated within these capabilities is automation support in the form of decision-centered analytics, that carry out low-level tasks in a transparent manner, reducing workloads and establishing the intelligent human-machine dialogues required to form appropriate attitudes of trust towards the system (e.g., avoiding overreliance). This approach has shown promise in supporting trust in the overall joint human-automation system, enabling the PED enterprise to roll out higher-level, planned automation capabilities to further offload PED tasks.

Keywords- PED; military intelligence; intelligence analysis; automation; intelligent HCI; trust in automation .

I. INTRODUCTION

Maintaining military superiority in the 21st century is of utmost importance to the United States armed forces, but this preservation does not come without cost and significant changes to doctrine, ideology, and process. In order to dominate the 21st century battlefield, the Army has the need to transform from the premier land force of the past, and enhance itself for the evolving conflicts ahead. Warfare and battlefield operations are evolving at a previously unseen rate, causing an increased emphasis on decision dominance and speed. This emphasis is appropriate, because it drives and informs the decision making process at every echelon. Decision management is enabled by information superiority, which can be defined as the speedy generation, collection, and effective use of information to inform Commander’s battlefield intent. The battle rhythm is shaped by this intent, which is passed through various echelons and different units through “information requirements” (IRs) about the environment [1].

For system developers looking to provide technology-based support in the form of automated capabilities, it is still unclear how this information collection is conducted, how it is converted into useful and manageable intelligence, and how it is distributed throughout the tiers of military command. Processing, Exploitation, and Dissemination (PED) is collectively defined as the conversion of collected information into forms suitable for the production of intelligence [1]. Overall, it is the process where analysts receive Commander’s intent from IRs, and set about collecting and analyzing raw intelligence, converting it into usable and command-actionable forms.

PED has been conducted for decades in continually evolving formats, but its key functional components are getting increasingly difficult to define as new technologies and missions continue to outpace force structure changes and blur the lines between individual roles, responsibilities, and authorities [2]. System designers looking to provide relief for PED operations with shrinking manpower and rapidly expanding volumes of MultiINT data need guidance to ensure developed capabilities will succeed. This guidance must be grounded in a robust and deep understanding of the PED force structure, capability gaps, existing tactics, techniques and procedures, and where PED is evolving to ensure novel systems and capabilities succeed within the challenge PED landscape.

Section II of this paper presents a brief overview of the current state of PED and where the domain is trending, based on a series of knowledge elicitation interactions that our team has had with the PED community. Section III then covers the need for providing automated support capabilities to meet the challenges of current and future PED. Section IV provides a set of guidelines for establishing and maintaining trust in automation as a critical requirement for successful system design and deployment within the PED domain. Finally, Section V presents a brief overview of our ongoing efforts to design, evaluate, and deploy a collaborative work environment to meet a number of challenges facing current and future PED, referred to as PEDX.

II. PROCESSING, EXPLOITATION, & DISSEMINATION

PED (see Figure 1) is one of the most essential pillars of intelligence collection today, with the Army’s focus currently being aimed towards developing ISR capabilities in support of PED. As MultiINT and multi-payload platforms have become increasingly utilized in current operations, they have not automatically reduced sensor operator workload or reduced manpower requirements. Instead these technologies

have led to both a growth of personnel requirements for their operation and, because they require operators to work at new intensities and new tempos of activity, they have created new complexities across military intelligence (MI) systems. While the Army is investing in novel sensors and automation capabilities in an attempt to reduce the burden placed on operators, acceptance of novel capabilities has been slow, delaying any significant enhancements to efficiency or performance at the forefront of the PED process. This results in analysts wedged between increased demands and reliance on antiquated technologies to perform their tasks.

The challenge for designers looking to provide new automated capabilities is that automation typically extracts humans from core processes that help them to better understand the context of a situation or analysis. In the case of multiple distributed operators (which is becoming commonplace as the Army transitions forward-deployed PED nodes connected to a centralized reachback PED center of excellence), if automation contributes to some small piece of a single contributor or cell of contributors workflow – the resulting impact on shared situational awareness can rapidly propagate to other collaborators as well. If this has significant impacts on collaborators productivity, the automated capabilities are likely to fail. This is evident by the many systems developed to aid PED analysts, which have fallen by the wayside in favor of antiquated, but proven technologies that are heavily relied upon throughout the PED community. While there is no doubt that automation has a significant role in both current and future PED workflows, introducing automation to those workflows in a way that will be adopted and enhance mission efficiencies is a challenging task that requires a strategic approach to planning, design, development, and deployment of system capabilities.

III. PED AUTOMATION BENEFITS

Ever increasing amounts of data are being generated by newly developed MultiINT sensor platforms, and demands for analysis of this data are increasing, all while the PED personnel footprint has remained stagnant and even reduced in many situations. In order for PED to continue to be successful in generating valuable and timely intelligence information with any sort of analytical rigor, automation is necessary to assist overburdened and overtasked analysts.

For example, the vast majority of collaboration across PED stakeholders takes place via online chat. As a result, part of dissemination activities often involves generation of a communication log. This requires analysts to manually compile chat logs that typically involve 20+ individual and group conversations that take place over the course of an analysis. It is not hard to envision an automated capability to log and compile these chat dialogues. In fact, many procedural and analytical tasks throughout the PED cycle lend themselves towards automation, if capabilities are employed and integrated into existing workflows correctly.

While there is a clear need for automation in the PED process, even in cases where automated capabilities exist there is failure to adopt and rely on them by analysts. Often it is the case where more seasoned analysts who have a reliable workflow simply do not trust automated processes enough to learn or rely on them. This is in part because the majority of existing capabilities fail to integrate with existing workflows, requiring time-stressed and overloaded analysts to blindly abandon their proven methods and rely on a new workflow. PED analysts suffer from the same problems of introducing new automation that common across domains, including lack of understanding of the automated techniques that have already been developed, the availability and usability of automated software to help with simple or repeated tasks, and their lack of faith in these capabilities leading to a mistrust and lack of familiarity with the tools with potential to increase operating efficiencies and thereby increase the robustness of PED analyses and artifacts.

IV. TRUST IN AUTOMATION

For any automated capability to succeed in the PED environment, it must encourage analysts to trust the capability to improve their performance and to consistently meet their expectations with respect to what it can provide. While many system attributes (e.g., reliability, performance, predictability, availability, explication of intentions) are known to influence trust in automation, it is important to note that it is not the actual state of these attributes that influences attitudes of trust, but rather the perceived state of these attributes (which may not align with the true automation capabilities). However, a prerequisite for any change in analyst trust to occur is the decision of the analyst

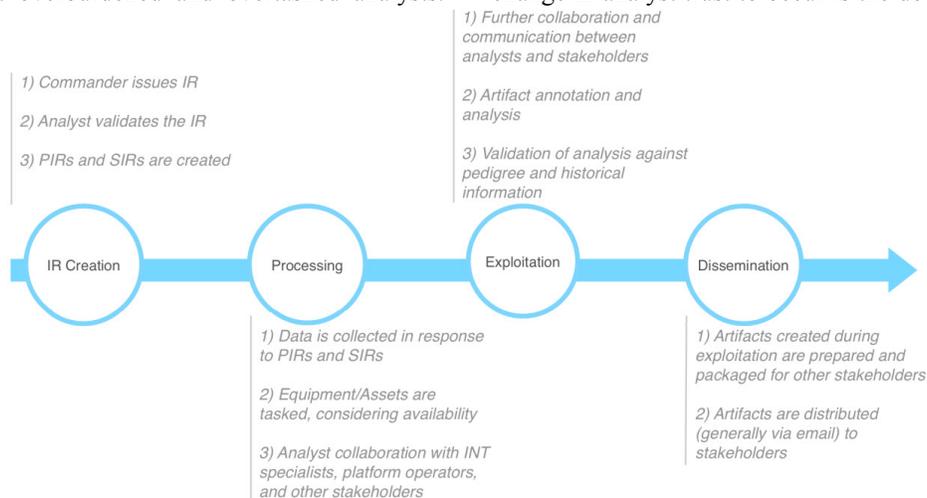


Figure 1. High-level PED workflow overview

to rely on the automated capability.

The decision to rely on any automated capability will result from situations where the analyst's reliance threshold (i.e., the point at which they decide to rely on an automated capability based on their system perception, workload, perceived risks, etc.) for the automated system is exceeded by the combination of changes in their trust in the system, self-confidence, and trust in alternative options. Consequently, any circumstances that result in decreases in the analyst's self-confidence or their trust in alternative options will create the potential for this prerequisite to be met and in turn for trust in the system of interest to evolve. As previously discussed, the real-world dynamics of the PED domain can create these circumstances based on changes to the overall task uncertainty, resulting from changes to data sets, problem structure and/or organizational uncertainty, or changes in the availability of the alternative options. Another set of circumstances that can result in the shifting of analyst trust based on differences in perceived capabilities would be situations where the analyst relies on the automated system and it meets/exceeds expected capabilities or fails to meet its expected capabilities (which would require a system feedback mechanism) causing the analyst to update the

perceived state of the various automated capability attributes. While there is a vast collection of existing literature on trust in automation (see [3-5]), the implication for system designers is the need to design automated capabilities that facilitate initially establishing trust, while also providing system feedback elements that can help maintain it over time through an appropriate man-machine dialogue.

A. Designing for Successful Automation

One of the most critical characteristic of an automated system that will facilitate appropriate attitudes of trust is the ability of that system to effectively communicate its ability to perform as designed and expected [3,5]. While this may seem a simple requirement to system designers, effectively building out a system that can proactively recognize system shifts and effectively communicate self-health in a timely manner so as to calibrate operator expectations is a significant challenge. For the PED domain, this issue is compounded by the reluctance of PED operators to deviate from existing proven workflows and technologies given minimal resources to dedicate to learning nuances of new systems and adapting their procedures on mission critical operations. For this reason, we have established a series of

TABLE 1. GUIDELINES FOR SUCCESSFUL PED AUTOMATION SUPPORT DEVELOPMENT

Guideline	Justification
Integrate with existing workflows and workflow support systems	The engrained reliance on existing tools and systems is unlikely to be severed given fast tempos and constrained resources for PED analysts. Without buffers to enable analysts to experiment with new tools and augmented workflows to justify changing practices, it is unlikely new technologies will succeed over existing, proven systems and practices. By integrating with existing systems and workflows, analysts will have the ability to rely on novel automation capabilities to start calibrating their expectations of how those capabilities can enhance their productivity leading to the establishment of an initial attitude of trust.
Focus initial system capabilities on low-level automation	The performance variability of an automated capability is most often dependent on the variability of the inputs it must act upon. If available inputs are consistently changing or are of questionable pedigree, there is an increased potential for system performance to suffer. For new systems being deployed to the PED domain, if performance is highly variable, then it will be more challenging for analysts to appropriately calibrate expectations for a given system interaction. This creates the potential for inappropriate attitudes of trust that can lead to further mismatches in system performance and expectations, or simply analysts deciding to not rely on the capabilities at all. Instead, novel systems should deploy a set of low-level automated capabilities that do not require inputs with varying degrees of reliability. These types of capabilities often target highly redundant tasks. The benefit of this approach is that it facilitates appropriate calibration of performance expectations (given that performance is unlikely to significantly change) and fosters initial establishment and ongoing maintenance of positive attitudes of trust in the system. This initial calibrated attitude of trust then serves as a foundation for deploying more volatile automation capabilities such that system designers will be able to leverage the existing trust to encourage reliance on new system features to in turn foster appropriately calibrated expectations – without resulting in a significant enough decrement of trust in cases where expectations are not met to lead to abandonment of system reliance altogether.
Maintain transparency in automation	This is a standard piece of guidance for any automated system, and it applies to PED automation as well. For analysts trying to maintain context of multiple parallel analyses, processing data or allocating workflow tasks to automated capabilities will create a degree of separation for analysts and degrade their frame of reference unless automated processes are made transparent.
Utilize an ongoing dialogue to calibrate expectations	Similar to the third recommendation, automated capabilities should maintain an ongoing man-machine dialogue to foster appropriately calibrated expectations of system performance – this will ensure that attitudes of trust do not degrade when analysts make the decision to rely on automated capabilities.
Do not force reliance	Related to the first recommendation, integrating any new capability into the PED workflow and forcing reliance is likely to result in an extreme prejudice towards the system if it fails in any way to meet analyst expectations. Forcing analysts to rely on automation, and abandon their already proven and trusted systems will create potential for analysts to enter into system reliance with a negative view of the system and a significant desire to validate the system as faulty or not meeting requirements as justification for reverting to their trusted systems. Instead, novel capabilities should be provided that the analyst can optionally rely upon, enabling them to depend on capabilities when they see the benefit so they enter into the interaction with positive expectations for capabilities that will benefit their own effectiveness. This will remove the bias from situations of reliance and improve (with properly calibrated expectations) the likelihood of the analyst's attitude of trust towards the system improving based on the interaction.

guidelines (see Table 1) for system developers seeking to design successful automated capabilities to assist analysts in the PED or other similarly characterized domains.

The underlying assumption with these guidelines is that successful deployment of novel technology-driven capabilities relies on the establishment and maintenance of an appropriately calibrated attitude of trust and pre-existing faith towards the introduced system's true capabilities. In order to establish and maintain this attitude of trust, it requires operators to make the decision to rely on the capabilities so they can benefit from the various feedback elements afforded by supporting the guidelines provided in Table 1. Therefore, these guidelines are intended to be applied in a stepwise manner, initially motivating decisions to rely on the automated capabilities to establish an initial attitude of trust and then incorporating the feedback dialogue layers to foster appropriate calibration and maintenance of this trust to motivate continued reliance on system capabilities when appropriate. This approach is beneficial as it encourages rapid deployment of relatively simplistic automated capabilities prior to their creation of more complex capabilities that may have a greater degree of performance variance (based on factors such as the quality of inputs made available to the system).

V. PEDX COLLABORATIVE WORK ENVIRONMENT

Through our numerous Knowledge Elicitation sessions with both reachback analysts and PED training experts, we have identified the central pillar of the PED cycle to be based around the efficient communications needed to support distributed (temporally and geographically) collaboration, with a particular focus on chat based collaboration. To this end we have begun prototyping an asynchronous chat client (PEDX; see Figure 2), modeled after a standard email client to increase familiarity and communication robustness by allowing for multiple concurrent conversational threads and in-line responses among other features. Chat is the main medium by which IRs are communicated, requirements for collection and exploitation are updated, and different stakeholders are informed of analysis. However, there is a severe lack of automation or utilization of new technologies employed today, with PED personnel continuing to rely on antiquated chat clients that become overwhelming as the number of conversations occurring at once increases.

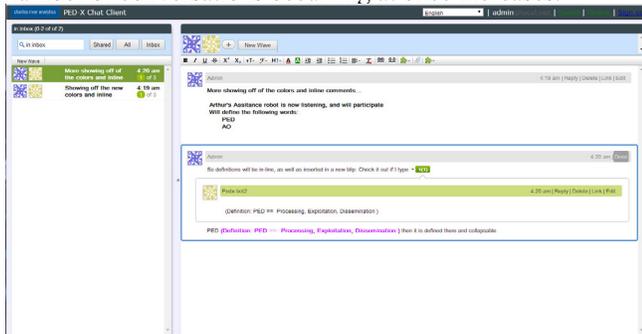


Figure 2. PEDX asynchronous thin-client chat

Following the guidelines presented in the previous section, we are developing and integrating automated

capabilities (e.g., alerting and attention direction systems) that maintain transparency and do not force reliance, we hope to raise awareness, acceptance, and reliance of these tools as a pathway to support higher level automated features. Currently, the list of low-level automated capabilities that PEDX supports includes the following, and while they may seem simplistic, they offer significant enhancements to the PED process by offloading mental workload and providing added support for shared awareness between geographically and temporally distributed collaborators:

- Message date and time stamping
- Multi-thread chat log compiling / output generation
- New or revised information and content alerting
- Information prioritization and organization
- Acronym definition
- Platform asset capabilities and availability indexing
- Automated chat agents (e.g., weather request bots)
- Integrated file sharing
- Asynchronous multiple user geospatial layout editing
- Automated database queries
- Thread / chat log search

VI. CONCLUSIONS

This paper provided an overview of our attempts to integrate novel automated capabilities into existing and evolving PED workflows. We present guidelines for successfully deploying new technology to PED (and similarly characterized domains) centered on the idea of facilitating initial establishment and ongoing maintenance of trust to motivate appropriate decisions to rely on new capabilities. Our strategy focuses on building initial trust through low-level automation that integrates with existing workflows and systems, and without a dependence on highly variable system inputs to ensure consistent performance. This reduces the need for analysts to frequently calibrate system performance expectations, helping the system to meet expectations. Once initial trust is established, it can be leveraged to support deployment of more complex automation with a higher degree of performance variability. The initial attitude of trust serves as a buffer that will encourage decisions to rely on the automated capabilities so expectations can be properly calibrated and appropriate decisions to rely can be made.

REFERENCES

- [1] M. Voshell et al., "Application of emerging technologies to support Army Processing Exploitation and Dissemination (TAPED)," Charles River Analytics, Cambridge, MA, USA, Tech. Rep. R0901542, 2013.
- [2] J. Kraiman, B. Kingston, and J. Muccio. "Modeling and simulation of the ISR tasking, processing, exploitation, and dissemination (TPED) process." *AeroSense 2000*. Int. Soc. for Optics and Photonics, 2000.
- [3] J. D. Lee, and K. A. See, "Trust in automation: Designing for appropriate reliance." *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46.1, pp 50-80, 2004.
- [4] R.R. Hoffman, M. Johnson, J.M. Bradshaw, and A. Underbrink, "Trust in automation," *Intelligent Systems, IEEE*, vol. 28(1), pp 84-88, 2013.
- [5] R. Parasuraman, Raja, and V. Riley. "Humans and automation: Use, misuse, disuse, abuse." *Human Factors*, 39.2 (1997): 230-253.

Evaluation of Visual Impression of Delayed Movement of Avatar while Exercising

Taeko Tanaka and Hiroshi Hashimoto

Master Program of Innovation for Design and Engineering
Advanced Institute of Industrial Technology
Tokyo, JAPAN
Email: { b1315tt, hashimoto }@aiit.ac.jp

Sho Yokota

Department of Mechanical Engineering
Toyo University
Saitama, JAPAN
Email: s-yokota@toyo.jp

Abstract— This paper considers psychological evaluation of the visual impression of the delayed movement of an avatar that performs interaction where actions of human are imitated by using the skeleton model obtained from Kinect sensor. In the interaction, the perception of the level of delay, impression of delay and habituation to the delayed movement of the avatar are investigated through some exercising experiments. From the results of the questionnaire for subjects who experience the delayed movement of the avatar, those visual impression are analyzed and the novel habituation based on a certain level in the experience is discussed.

Keywords-*delayed movement; avatar; visual impression; habituation.*

I. INTRODUCTION

Avatar can be projected on a screen in real time by applying humanoid Computer Graphics (CG) on the skeleton model extracted from the human motion capture. By seeing the avatar, the user can evaluate one's own motion in real time while moving.

However, in the real-time display of the avatar, in fact, time delay occurs during the process of extracting information from body motion and information process of applying it to humanoid CG. In other words, time delay occurs while the movement of user is reflected and displayed in the avatar.

Time delay is known to affect the human psychology. Many research works have been undertaken regarding this mainly focusing on the interaction between humans and artifacts. It was pointed out that the delay of the computer response time adversely affects psychology [1]. The psychological influence in the utterance delay was studied well [2],[3], and it was found that delay of one second or more has adverse impact, and voice of the conversation tends to increase. The effect of appearance of an artificial agent and utterance time on psychology was studied [4],[5], and it was shown that higher is the delay, worse are the psychological changes. In the conversation between humans and robots, it was investigated the effect of starting time of utterance by Robot and timing of nodding on the psychology, and revealed that delay gives bad feelings [6],[7].

In these studies, it is stated that in the interaction between humans and artifacts, delayed reaction of artifacts to a stimulus from the outside world has a negative impact on the psychology of humans. This impact pertains to usability

when a human uses the artifacts, and it must be treated as an important problem. However, these studies consider the cases while verbal communication is taking place, and they do not discuss the effect of time delay in the body motion interaction between humans and artifacts on the psychology.

In this paper, we will discuss about psychological evaluation of the visual impression of the delayed motion of the avatar as an artifact that performs interaction where actions of human are simulated. Motion considered in this paper is swing movement often seen in exercising, where the human raises and lowers both arms. We will have this discussion about perception of the level of delay, impression of delay, and habituation to delay. Here, the level of delay means a quantitative expression of how much the delay a human feels.

We will explain the details of the experiment conducted in this paper for psychological evaluation. Recent software systems of artifacts can adjust the delayed degree of movement with digital filter functions. In other words, it is possible to adjust the delay time in the process of displaying avatar with human motion capture. Using this, in our experiment, we measured the stage from when the human clearly recognizes it as delay when the delay time is changed in a stepwise manner. Measuring this delayed degree should be useful in offering guidance for improving the avatar system.

Next, we administered a questionnaire survey about the impression the subject got when seeing the avatar that moves according to the movement of the subject. We studied the impression the subject got when he saw that movement of the avatar is slower than him (hereinafter referred to as the delayed movement) while the subject does the swing movement.

In Section 3, in order to obtain the characteristics of this impression, we use different movements than the delayed movement. These are two types of movements, namely, state where movements of the subject and the avatar seem to be matching (hereinafter referred to as the synchro movement) and the state where movement of the avatar seems to have progressed than that of the subject (hereinafter referred to as the lead movement). The reason why these two movements are conducted is that the synchro movement is used for the bench mark and the lead motion is used to highlight the visual impression for the delay movement. Then, we will compare impression evaluation and consider habituation to delay.

II. EXPERIMENTAL ENVIRONMENT AND METHOD

Hardware used in the experiment is comprised of Microsoft Kinect for measuring the movement of the subject, PC that creates movement of the avatar based on the movement of the subject, and projector and screen for displaying the avatar to the subject.

Figure.1 shows the hardware configuration for measurement of the human motion and the avatar display system. As software, we used Kinect for windows SDK [8], which is the library for obtaining the human motion from the depth data photographed with Kinect and Microsoft XNA [9] ,for drawing the avatar. With Kinect for Windows SDK, we can obtain the subject's movement data, and by transferring this data to Microsoft XNA, avatar can display identical movement as the subject.

Figure. 2 shows an image of the avatar displayed to the subject during the experiment. Figure. 3 shows the experiment in progress where the subject is moving his body while watching the avatar. Strictly speaking, movements of the avatar and the subject are not synchronized. Rather, after measuring the movement of the subject with Kinect, movement is created in the avatar and after that the avatar will act. Therefore, irrespective of whether the subject

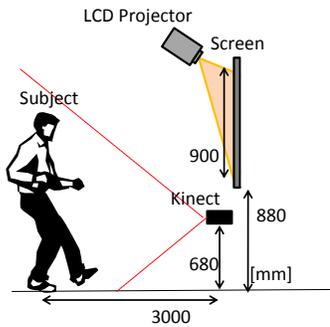


Figure. 1. Outline drawing of the experiment setup.

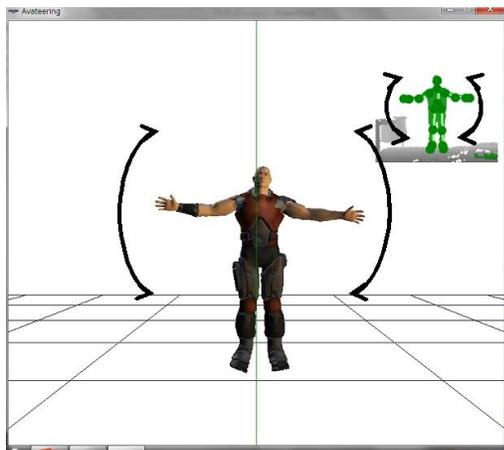


Figure. 2. Avatar (in the middle of the Figure) doing the swing movement

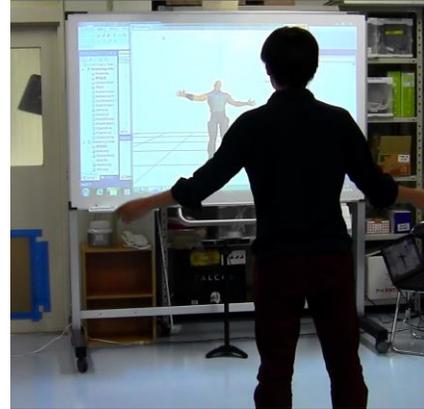


Figure. 3. Experiment in progress.

realizes or not, movement of the avatar starts with delay. In Kinect, with the filter process, by delaying and advancing the subsequent movement, it is possible to control the delay time from the actual movement.

In this experiment, in the first place, we will measure the level of delay where the subject feels that delay has occurred while gradually increasing the delayed degree of the movement of the avatar. For implementing the delayed movement in the avatar, in this paper, we adjust the parameters of digital filter included in Kinect for Windows SDK. This filter is equipped with the smoothing function, and it is used for removing the errors in the measurement data where such errors have occurred due to disturbance of shooting conditions and the like in Kinect. By changing the parameters of this filter, it is possible to have smoothing effect and cause delay in the movement data of the subject obtained with Kinect. Then, by implementing this movement data with delay on the avatar, movement of the avatar will be delayed than the actual movement of the subject.

As the filter parameters, Prediction [≥ 0.0] and Smoothing [0.0 , 1.0] are available. Value of Prediction is the number of frames that predicts the movement. Its default value is 0.0, and it tends to overshoot from around 0.5. So, we used the values in the range of 0.4 and below. Value of Smoothing is the smoothing index. When it is 0.0, there is no delay, and when it is 1.0, delay is at the maximum. Default value is 0.5, and based on our experience, we selected values equal to or higher than this value.

Table I shows the perceived level of delay in five stages as the level of delay with respect to the delayed movement in this experiment.

We will now explain about the method of creating this perceived level. During the course when the subject moves his body while watching the avatar displayed on the screen, we gradually changed the parameter value using the Smoothing function of Kinect. When increasing the parameter of Smoothing, the movement of the avatar will not be able to keep up with the movement of the subject, and movement will become sluggish. This state where the avatar hardly move is considered as maximum delay. Against this, the state where the movement of avatar appears to be

TABLE I. PERCEIVED LEVEL WITH RESPECT TO DELAY

Parameter set #	Prediction, Smoothing
1 (minimum delay)	Prediction=0.4, Smoothing=0.5
2	Prediction=0.3, Smoothing=0.6
3	Prediction=0.2, Smoothing=0.7
4	Prediction=0.1, Smoothing=0.8
5 (maximum delay)	Prediction=0.0, Smoothing=0.9

TABLE II. PARAMETER SET OF LEAD AND SYNCHRONIZATION

Parameter set	Prediction, Smoothing
Lead	Prediction=1.0, Smoothing=0.5
Synchronization	Prediction=0.5, Smoothing=0.5

synchronous with the subject himself is considered as minimum delay, and this interval was divided into 5 stages.

Subject's movement of raising and lowering arms while watching the avatar was aligned too to the metronome of 100BPM (Beat Per Minute). Parameter set #1 through #5 shown in Table I was changed every 5 seconds. The subject would move his body for every parameter set. After that the subject was asked "Do you think that the avatar you just saw was delayed compared to your movement?" Subject's response was collected in Yes / No or Possibly as shown in Table III. This was repeated 5 times, and response data was collected and summarized.

For verifying impression evaluation with respect to delayed movement, we thought that it is necessary to have another comparison target. Based on this, we designed "synchro movement" and "lead movement". The former one synchronizes with the movement of the subject, while the latter one advances the phase of movement using differential operation. This was implemented using the parameters shown in Table II.

In this experiment, we used three movements, namely, "delayed movement", "synchro movement", and "lead movement". The "synchro movement" is placed as a benchmark to measure objectively, to compare, and to evaluate the difference of the impression.

The following experiment was carried out for impression evaluation.

[Step 1] In the first place, in order to have the experience of the delayed movement of the avatar, while watching the avatar moving as per the settings of #3 in Table I, the subject moved his body for about 5 seconds along with the sounds of metronome and experienced the delayed movement of the avatar. Similarly, the subject moved his body for about 5 seconds for the lead movement (Lead in Table 2) and the synchro movement (Synchronization in Table 2) and experienced these movements.

[Step 2] In order to find out perception of the level of delay, we changed the parameter set in Table I from #1 to #5 at

every 5 seconds. Every time when changing the parameter, we asked the subject whether the movement is delayed or not.

[Step 3] Next, we find out how the impression regarding delayed differs from synchro movement and lead movement. For that, for each subject, we run the delay movement using the parameter sets in Table I for which the subjects felt the delay, and we changed the movements of avatar as per the following patterns.

[Pattern 1] delayed movement (10 seconds) → synchro movement (10 seconds) → delayed movement (10 seconds)

[Pattern 2] synchro movement (10 seconds) → synchro movement (10 seconds) → synchro movement (10 seconds)

[Pattern 3] lead movement (10 seconds) → synchro movement (10 seconds) → lead movement (10 seconds)

These patterns were created based on the concept of placing the synchro movement at the middle position, and placing three types of movement patterns on both sides.

In this experiment, there was 14 subjects, all males in their 20s. As for the sequence of the experiment, after completing [Step 1], subjects went to [Step 2], and after that they went to [Step 3]. Step 1 is preparation for the experiment to be conducted here onwards.

III. ANALYSIS OF IMPRESSION EVALUATION

A. Evaluation Items

Restating the explanation given in Chapter 2, the following are the evaluation items in impression evaluation.

P1: From what stage does the subject sense "delay" in the movement of the avatar? This leads to perceptual evaluation of the level of delay.

P2: What kind of the impression the subject forms regarding delayed movement of the avatar?

P3: Look into impression of each movement of the avatar, and see if there are any differences in the evaluation of each pattern. This leads to finding out habituation to delay.

For investigating P1, we conducted the experiment mentioned in [Step 2] in the preceding section. For investigating P2 and P3, we conducted the experiment mentioned in [Step 3].

B. Analysis of P1

Response data for three perceptions, namely, the movement of the avatar is "delayed", "possibly delayed", and "not delayed", was summarized for each parameter set. Table III shows the results of this.

From the results in Table III, for parameter set 3 and above, about 40% of the subjects responded that the movement of the avatar is "delayed". For parameter set 2 and above, about half of the subjects responded that the movement of the avatar is "possibly delayed". Smoothing of parameter set 2 is set only slightly higher than the default

TABLE III. RESPONSES WHERE THE SUBJECTS FELT THAT THE MOVEMENT IS DELAYED WITH RESPECT TO THE PARAMETER SET IN TABLE I

		delay	possibly delay	not delay	total
Parameter set 1 Prediction=0.4, Smoothing=0.5	number	0	0	14	14
	rate (%)	0.0	0.0	100.0	100
Parameter set 2 Prediction=0.3, Smoothing=0.6	number	0	8	6	14
	rate (%)	0.0	57.1	42.9	100
Parameter set 3 Prediction=0.2, Smoothing=0.7	number	6	6	2	14
	rate (%)	42.9	42.9	14.3	100
Parameter set 4 Prediction=0.1, Smoothing=0.8	number	6	6	2	14
	rate (%)	42.9	42.9	14.3	100
Parameter set 5 Prediction=0.0, Smoothing=0.9	number	10	4	0	14
	rate (%)	71.4	28.6	0.0	100

value, and it resulted in somewhat ambiguous perception. In the case of parameter set 4 and 5, the subjects are divided into two groups, namely, group that clearly recognized that the movement is "delayed" and the group that vaguely sensed the delay. However, this excludes a small number of subjects who responded that the movement is "not delayed". In the case of parameter set 5, about 70% of the subjects recognized that the movement of the avatar is clearly "delayed".

Based on these results, it came to light that the subjects sense the "delayed movement" of the avatar from parameter 3 onwards. At the stage of parameter set 2, the subjects may not sense that the movement is delayed.

C. Analysis of P2

In P2, we administered a questionnaire survey to find out the kind of impression with respect to the "delayed

movement" of the avatar. Simultaneously, apart from the "delayed movement", we also studied the "synchro movement" and the "lead movement". With regard to the pair of adjectives used in this method, we referred to the previous studies [10], [11], related to impression evaluation of the movement of robot, and we prepared 13 pairs of adjectives shown in Table IV and we conducted evaluation in 7 stages.

Table V shows the average value of response data obtained from 14 subjects for three movements. Next, from the data group of each movement, in order to find out relationship with the respective impression evaluation, we conducted correspondence analysis [12], on the results shown in Table V. Figure. 4 shows the outcome of this analysis.

The following can be concluded from the results shown in Figure. 4.

- For the "delayed movement", the subjects formed the impressions such as "like other human", "unexpected", and "unfriendly", and other impressions such as "fast and slow" and "moderate" based on the speed of movement
- For "synchro movement", the subjects formed the impressions such as "smooth", "natural", "like oneself", "enjoyable", "soft", and "comfortable".
- Subjects formed the impression that the "lead movement" was "hard" and "intense". However, some of the subjects responded that they formed the impressions such as "interesting", "as expected", and "pleasant".
- As compared to the "synchro movement", the subjects clearly realized the difference in the movement in the "delayed movement". The subjects felt uncomfortable that the movement of avatar didn't match with their movement.
- There were some subjects who favorably treated the "delayed movement" as smooth movement. However,

TABLE IV. IMPRESSION QUESTIONNAIRE ITEMS FOR P2

Evaluation Items			
1	Fast	↔	Slow
2	Smooth	↔	Awkward
3	Like oneself	↔	Like other human
4	As expected	↔	Unexpected
5	Comfortable	↔	Uncomfortable
6	Soft	↔	Hard
7	Sudden	↔	Not sudden
8	Pleasant	↔	Unpleasant
9	Interesting	↔	Boring
10	Intense	↔	Moderate
11	Susceptible	↔	Insusceptible
12	Amiable	↔	Unfriendly
13	Natural	↔	Unnatural

TABLE V. IMPRESSION EVALUATION RESULTS USING THE SD METHOD

Evaluation items	Average Value of Response Data		
	delay	synchro	lead
1	4.86	3.57	3.57
2	4.00	3.14	4.00
3	4.57	3.43	3.29
4	4.57	4.00	3.71
5	4.14	3.00	3.71
6	4.14	3.71	4.43
7	4.86	4.14	4.00
8	4.14	3.29	2.71
9	3.86	3.29	3.00
10	4.86	3.71	3.71
11	4.00	4.14	4.00
12	4.71	3.43	3.14
13	4.71	3.43	3.57

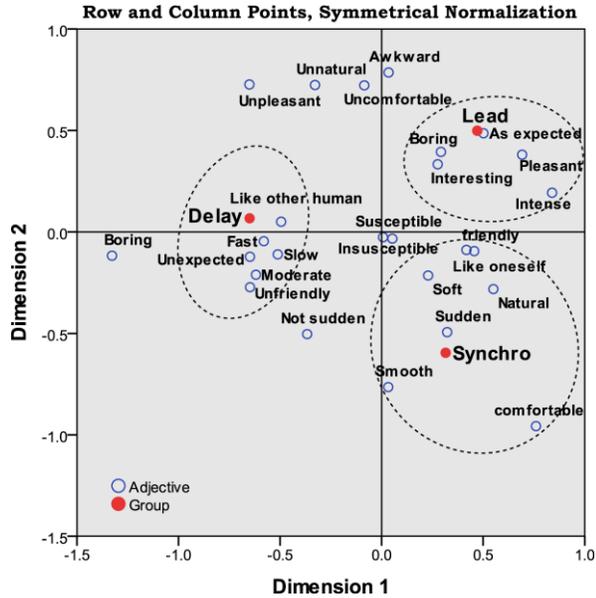


Figure. 4 Results of correspondence analysis.

in terms of the overall trend, subjects had a negative impression of the “delayed movement”.

- Impression became positive in the case of the “synchro movement”.
- In the “lead movement”, while there was negative impression, simultaneously, the subjects also found it “interesting” and “pleasant”.
- In the settings of the "lead movement", in the present Kinect, the avatar reacted acutely to the speed of exercising in the subjects, which formed the impression such as "hard" and "intense". However, there were opposite responses to this impression such as "interesting", "as expected" and "pleasant".

D. Analysis of P3

In the preceding paragraph, we mentioned that apart from "hard" and "intense" that was the impression evaluation with respect to the "lead movement", subjects formed the impression of "interesting", "as expected" and "pleasant" as in the case of "synchro movement". Because it was found that the subjects formed similar impression in these two movement patterns, we will verify whether there are any differences in impression between the "synchro movement" and the "lead movement". From the point of view, P3 was designed.

We carried out impression evaluation for the experiment [Step 3] where three types of movements, namely, "delayed movement", "synchro movement", and "lead movement" are combined. Here, data group for each of three types of movements of avatar were named as data group of movements. We set the hypothesis that "there is no difference between levels due to the data group of movement", and we carried out corresponding one-way analysis of variance (Repeated measures ANOVA)[13].

TABLE VI. TEST RESULTS OF EFFECT BETWEEN SUBJECTS

Source	Type III Sum of Squares	Degree of freedom	Mean square	F value	Significance level	
Intercept	Hypothesis	8138.579	1	8138.579	624.110	.000
	Error	169.524	13	13.040 ^a		
Subject No.	Hypothesis	84.806	2	42.403	17.757	.000
	Error	1179.619	494	2.388 ^b		
Data group	Hypothesis	42.183	12	3.515	1.472	.131
	Error	1179.619	494	2.388 ^b		
Evaluation items	Hypothesis	169.524	13	13.040	5.461	.000
	Error	1179.619	494	2.388 ^b		
Data group * Evaluation items	Hypothesis	41.289	24	1.720	.720	.832
	Error	1179.619	494	2.388 ^b		

a. Mean square(id), b. Mean square (Error)

TABLE VII. RESULTS OF MULTIPLE COMPARISON

Dependent Variable: Bonferroni

Movement		Difference in average value (a) - (b)	Standard error	Significance level	95% Confidence Interval	
					Lower limit	Upper limit
Delay	Synchro	.86*	.162	.000	.47	1.25
	Lead	.81*	.162	.000	.42	1.20
Synchro	Delay	-.86*	.162	.000	-1.25	-.47
	Lead	-.04	.162	1.000	-.43	.35
Lead	Delay	-.81*	.162	.000	-1.20	-.42
	Synchro	.04	.162	1.000	-.35	.43

Based on the observed average value. Error value is mean square (error) = 2.388.

* Difference in average value is significant at 0.05 level.

Table VI shows the results of this analysis. The one-way analysis of variance (Repeated measures ANOVA) is used to determine whether there are any significant differences between the means of two or more groups.

Results in Table VI showed statistical significant in the class of movement from the significance level (p < 0.01). Accordingly, the hypothesis "There is not difference between the levels" was rejected, and it can be said that the impression formed in the subjects for three movements of the avatar are different.

Furthermore, in order to shed light on the difference between movements of different phase, we used the Bonferroni's method [14], and conducted multiple comparison. Table VII shows the results of this comparison.

Based on these results, it is evident that in the movements of the avatar, "delayed movement" and "synchro movement", and "delayed movement" and "lead movement" are statistically significant (p<0.05). In other words, the impression formed for "delayed movement" is different from that for "synchro movement" and "lead movement". On the other hand, it cannot be said that impression differs for "synchro movement" and "lead movement".

IV. DISCUSSION AND CONCLUSION

In the present paper, we shed light on the numerical value of the level of delay based on the experiment where the subject recognizes that the movement of the avatar is "delayed" from his movement, and we verified its stage. Next, we conducted a survey about impression formed by the subject regarding the avatar that moves out of synchronization with the subject.

To start with, as the results of P1 stated in this paper, we conducted experiment and quantitatively define the level of delay where the subjects recognize that the movement of the avatar is "delayed" than their movement, and we ascertained the stage of this level. As a result, it became clear that at parameter 3 and above, about 40% of subject sensed "delayed movement" where the movement in the avatar was delayed compared to the subjects' movement.

Next, as the result for P2, it came to light that the subjects feel uncomfortable with the "delayed movement". In the "synchro movement" experienced by the subjects after the delayed movement, they formed the impressions such as "natural", "like oneself", and "amiable", and in the "lead movement", the subjects formed the impressions such as "hard" and "intense", as well as "interesting", "as expected", and "pleasant".

In P3, we verified whether there is any difference in the impression evaluation of each of three types of movements of the avatar confirmed in P2, namely, "delayed movement", "synchro movement", and "lead movement". Here, we found that while the "delayed movement" gave a different impression than the "synchro movement" and the "lead movement", it cannot be said that impression differed in the "synchro movement" and the "lead movement". As for the impression of the "lead movement", the impression evaluation was "interesting", "as expected" and "pleasant", which was most likely because of habituation [15], in perception in terms of mitigation of the sense of discomfort to time delay and adverse psychological effect (becoming insensitive). This habituation differs from simple stimulation [16], mentioned in the preceding studies and reactive habituation [17], that occurs due to iterative presentation of irritation. Habituation showed by these results are similar to habituation explained by [18], [19], in terms of order effect where after experiencing the "synchro movement", the subjects become insensitive to the delay of the movement. However, we think it is a new finding that the order of movement patterns affects psychology. Nonetheless, we have used only three patterns of order in this experiment, and our next challenge is to study and discuss changes in impression and habituation for different order of movements.

ACKNOWLEDGMENT

This work was in part supported by JST RISTEX Service Science, Solutions and Foundation Integrated Research Program and by JSPS KAKENHI Grant Numbers 25280125.

REFERENCES

- [1] J. Klein, Y. Moon and R. W. Picard, "This computer responds to user frustration, Theory, design, and results," *Interacting with Computers*, ELSEVIER, vol.14, Issue 2, 2002,pp. 119-140.
- [2] A. R. Pearson, T. V. West, J. F. Dovidio, S. R. Powers, R. B. and R. Henning, "Divergent Effects of Delayed Audiovisual Feedback in Intergroup and Intragroup Interaction, vol.19, no.2, 2008, pp. 1272-1279.
- [3] S. R. Powers, C. Rauhb, R. A. Henning, R. W. Bucke and T. V. Weste, "The effect of video feedback delay on frustration and emotion communication accuracy," *Computers in Human Behavior*, ELSEVIER, vol.27, no.5, 2011,pp. 1651-1657.
- [4] H. Prendinger, J. Mori and M. Ishizuka, "Recognizing, Modeling, and Responding to Users' Affective States," *User Modeling, Lecture Notes in Computer Science*, vol.3538, 2005,pp. 60-69.
- [5] N.C.Kramer, N.Simons and S.Kopp, "The Effects of an Embodied Conversational Agent's Nonverbal Behavior on User's Evaluation and Behavioral Mimicry," *Intelligent Virtual Agents Lecture Notes in Computer Science*, Springer Berlin Heidelberg, vol.4722, 2007,pp. 238-251.
- [6] S.Takasugi,S.Yoshida, K.Okitsu, M.Yokoyama, T.Yamamoto and Y.Miyake, "Influence of Pause Duration and Nod Response Timing in Dialogue between Human and Communication Robot," *Trans. of the Society of Instrument and Control Engineers*, vol.46, no.1, 2011,pp. 72-81.
- [7] M.Yamamoto, T.Watanabe, "Timing Control Effects of Utterance to Communicative Actions on Embodied Interaction with a Robot and CG Character," vol.24, no.1, 2008,pp. 87-107.
- [8] Kinect for windows SDK, <http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx>.
- [9] Microsoft XNA, <http://msdn.microsoft.com/en-us/centrum-xna.aspx>.
- [10] Y. Suzuki and R. Ohmura, "Impression Evaluation of Pointing Prediction Based on Minimum-Jerk Model," *IPSJ Interaction2013*, 2013,pp.249-254.
- [11] T. Kanda,H. Ishiguro,T. Ono,M. Imai and R. Nakatsu, "An evaluation on interaction between humans and an autonomous robot Robovie" *J. of the Robotics Society of Japan*, vol.20, no.3, 2002,pp. 315-323.
- [12] M.J.Greenacre, *Theory and Applications of Correspondence Analysis*, Academic press, 1984.
- [13] Dr. Andy Field, *Repeated Measures ANOVA*, *Research Method of Psychology*, 2008.
- [14] O. J. Dunn, "Multiple Comparisons Among Means," *J. of the American Statistical Association*, vol. 56, no. 293,1961,pp.52-64.
- [15] R. B. Zajonc, "Attitudinal Effects of Mere Exposure," *the American Psychological Association, Inc.*, vol.9, no. 2, Part2, 1968.
- [16] R.L.Moreland and R. B.Zajonc, "A Strong Test of Exposure Effects," *J. of Experimental Social Psychology*, vol.12, 1976,pp.170-179.
- [17] E. H. Jones and J. J.B. Allen, *The role of affect in the mere exposure effect: Evidence from psychophysiological and individual differences approaches*, *Personality and Social Psychology Bulletin*, 2001,pp.889-898.
- [18] H. Schuman and S. Presser, *Questions & Answers in Attitude Surveys*, Academic Press., 1981.
- [19] D.W. Moore, *Measuring new types of question-order effects*, *Public Opinion Quarterly*, no.66, no.1, pp.80-91, 2002

On the Robustness of Regression Type Classifiers

Olgierd Hryniewicz

Systems Research Institute
Polish Academy of Sciences
Warszawa, Poland

Email: hryniewi@ibspan.waw.pl

Abstract—Six regression type binary classifiers based on linear and logistic models have been evaluated using a complex simulation experiment. The classifiers were compared with respect to the robustness to unexpected changes of the models that describe data in training and test sets. The simple logistic regression has appeared to be the best one in these circumstances.

Keywords—Binary classification; Regression type classifiers; Robustness.

I. INTRODUCTION

Classification algorithms are probably the most frequently used tools of data mining. The methods of their construction in the Artificial Intelligence (AI) community is known under the name of *supervised learning*. There are thousands of books and papers devoted to their theory and applications. Thomson Reuter’s scientific database Web of Science displays information (as on 2015 April 22th) on over 96,000 papers with a topic related to the query “classification algorithms”. The information about theoretical foundations of classification algorithms can be found, e.g., in books by Duda et al. [1] and Hastie et al. [2]. Comprehensive description of application aspects of classification algorithms can be found in the book by Witten et al. [3].

The main problem with the evaluation of each, from among hundreds of already proposed, classifier is estimation of its quality characteristics. Japkowicz and Shah in their excellent book [4] write about two general approaches to this problem: *de facto* approach based on computing of many different quality characteristics, and *statistical* approach in which unavoidable randomness of classification results is taken into account. The *de facto* approach can be used for any type of testing procedure, and is predominately used by the AI community. The applicability of the statistical approach is somewhat restricted, as the analyzed data should fulfill some requirements precisely described in terms of the theory of probability. These requirements are easily verified if we use for testing purposes artificially generated data. However, the usage of such data is not appreciated by the AI community, who prefers to use real-life benchmarks for evaluation purposes. When we use benchmark data for evaluation, the data used for the construction of an algorithm and the data used for its evaluation come from the same set of real-life values. In order to assure validity of comparisons different schemes of randomization, e.g., cross-validation techniques, are used. This approach is commonly accepted, and valid for the great majority of potential applications. It is, usually rightly, assumed that a classifier (in fact, the method of its construction) is of good quality if it performs well on many

different benchmarks. However, in nearly every case (see, e.g., Hand [5]) it is assumed that the classifier is constructed and further used on *the same population* of classified objects. In some cases, however, this assumption may be questioned.

Robustness is well defined in statistics. According to Wikipedia, robust statistics “is a statistical technique that performs well even if its assumptions are somewhat violated by the true model from which the data were generated”. This definition of robustness can be directly applied to these methods of classification which are based on well established statistical methodology, such as, e.g., regression. In general, however, many classification methods, such as, e.g., neural networks or decision trees, are not based (at least, directly) on statistical models. Therefore, in the machine learning community robustness is often understood somewhat differently, as the ability to perform well for many different sets of real data. David Hand, one of the most renowned researchers in the area of machine learning, in his overview paper [5] discusses consequences of breaking the assumption that the data in the design (training) set are randomly drawn from the same distribution as the points to be classified in the future. He gives references to some works related to this problem, and presents examples of problems encountered in the area of the credit scoring and banking industries. It has to be noted, however, that the number of papers devoted to the problem of robustness, understood as in [5], is rather small. For example, Japkowicz and Shah [4], while discussing this type of the robustness of classifiers, cite only the paper by Hand [5].

Hryniewicz [6] [7] considers the case when binary classifiers are used for quality evaluation of items in production processes. In many cases of such processes, quality characteristics cannot be directly evaluated during production time. Sometimes it is impossible, when a testing procedure is destructive or impractical, or when a testing procedure is costly or lasts too long. In such cases, an appropriate classifier which labels monitored items as “good” or “bad” is constructed using the data coming from specially designed (and usually costly) experiments, and then used in production practice. The situation does not rise any objections if the process from which items used in the construction phase of a classification algorithm are taken is *the same* as a process in which obtained classifiers are used. Hryniewicz [6], [7] has demonstrated that deterioration of such process may have detrimental effects on the quality of classification. Similar problems may be also encountered in other fields of applications. Consider, for example, a classifier which is used for the prediction of cancer recurrence who may change its quality characteristics when

future patient will undergo a treatment which was not used at the moment when this classifier was built.

The problems described in the previous paragraph may suggest that in the evaluation of classifiers we should add another dimension, namely the robustness to the change of population understood as the change of probability distributions that describe input variables in the classification process. At the moment this can be achieved using artificially generated data, as appropriate, and widely known, benchmarks seem not to exist. In the research described in this paper, we have used software designed for the generation of complex nonlinear processes with statistically dependent data described in Section II. We have evaluated binary classifiers whose construction is based on generalized linear models and regression techniques. In particular, we analyzed classifiers based on

- Simple linear regression,
- Linear regression with interactions,
- Simple logistic regression,
- Logistic regression with interactions,
- Linear Discrimination Analysis with a symmetric decision criterion,
- Linear Discrimination Analysis with an asymmetric decision criterion.

We have assumed that the dependence between variables in our simulation model may be described by different copulas, characterized by different strength of dependence. The main goal of the research was twofold. First, we have tried to evaluate the robustness of the considered classifiers to shifts of the expected values of input variables (attributes). Second, we have tried to find if such robustness depends upon the type of dependence and its strength. Because of limited volume of this paper only few results will be presented in details. In contrast to the results published by other authors, we present the results of experiments performed in a strictly controlled environment that simulates conditions which are significantly different from those usually assumed for the considered classification models.

The paper is organized as follows. In Section II we shortly describe the simulation software and considered classifiers. Then, in Section III we describe some methods of evaluation. The most important results of experiments will be illustrated with examples in Section IV. Finally, in Section V we will conclude the experiments taking also into account the results that have not been presented in details in this paper.

II. DESCRIPTION OF SIMULATION EXPERIMENTS

A. Simulation software

Realization of the task formulated in Section I requires an implementation of complex mathematical model in a form of simulation software. On the most general level, let us assume that a general mathematical model that describes dependence of input variables (predictors) with an output binary variable is a simple one. Let Z_1, \dots, Z_p be p output characteristics whose values are not directly observed in an experiment. Assume now that these values should be predicted using observations X_1, \dots, X_k of k predictors. This problem is easy to solve if we assume that we know the joint probability distribution of input and output variables, i.e., the probability distribution of a combined vector $(Z_1, \dots, Z_p, X_1, \dots, X_k)$. According to the famous Sklar's theorem this distribution is unequivocally

described by a $(p + k)$ -dimensional copula, and marginal probability distributions of Z_1, \dots, Z_p and X_1, \dots, X_k . Such a general model is hardly applicable in practice. Therefore, our simulation software should be based on a model which is simpler and more easy for practical interpretation. In this research we have used a hierarchical 3-level model, originally proposed in [6]. On the top level of this model there is an auxiliary one-dimensional real-valued variable T . This value is transformed to a binary one (in which we are interested in) by means of the following transformation

$$Z_t = \begin{cases} 0 & , T \geq t \\ 1 & , T < t \end{cases} \quad (1)$$

The instances with the value $Z_t = 1$ we will call "positive cases" or "positives", and the instances with the value $Z_t = 0$ we will call "negative cases" or "negatives". This model has a direct interpretation in the case considered by Hryniewicz [6] who modeled a monitoring of a production process with indirectly observable quality characteristic. The first level of our model describes the predictors X_1, \dots, X_k . In order to simplify simulations we assume that consecutive $k - 1$ pairs of predictors $(X_i, X_{i+1}), i = 1, \dots, k - 1$ are described by $k - 1$ copulas $C_i(F_i(X_i), F_{i+1}(X_{i+1})), i = 1, \dots, k - 1$, where $F_1(X_1), \dots, F_k(X_k)$ are the cumulative probability functions of the marginal distribution of the predictors. In order to simulate the input variables we have to assume the type of the proposed copulas, and the strength of dependence between the pairs of random variables whose joint two-dimensional probability distributions are described by these copulas. In the AI community Pearson's coefficient of correlation r is often used as the measure of dependence. Unfortunately, its applicability is limited to the case of the classical multivariate normal distribution, or - in certain circumstances - to the case of the multivariate elliptic distributions (for more information see [8]). When dependent random variables cannot be described by such a model, and it is not an unusual case in practice, we propose to use Kendall's coefficient of association τ defined, in its population version in terms of copulas, as (see [9])

$$\tau(X, Y) = 4 \int \int_{[0,1]^2} C(u, v) dC(u, v) - 1. \quad (2)$$

Numerical comparisons of the values of Pearson's r , Kendall's τ , and - another popular nonparametric measure of dependence - Spearman's ρ are presented in [10], and show that the usage of Pearson's r in the analysis of data that cannot be described by the normal distribution may lead to wrong conclusions, especially in the case of negative dependence. Therefore, Kendall's τ is, in such cases, a much better measure of dependence.

In order to have a more realistic model for simulation purposes, it was proposed in [6] to use an in-between second level of latent (hidden) variables HX_1, \dots, HX_k . Each hidden variable HX_i is associated with the predictor variable X_i , and its fictitious realizations are measured on the same scale as the predicted continuous random variable T . The dependence between HX_i and X_i is described by a copula $C_{Hi}(F_{Hi}(HX_i), F_i(X_i))$. Moreover, in our model we assume that there exists a certain linear relationship between the expected value of HX_i and the expected value of X_i . This assumption is needed if we want to model the effects of the

shifts in the expected values of the predictors on the expected value of the predicted auxiliary variable T which is related to the hidden variables by a certain, possibly nonlinear, function

$$T = f(HX_1, \dots, HX_k). \tag{3}$$

In real circumstances, such as those described in [6], the probability distribution of T , and hence the probability distribution of Z_t , can be observed only in specially designed experiments. The results of such experiments can be viewed upon as data sets coming from supervised learning experiments. In our research we simulate similar experiments, and we use actual (i.e., generated by our software) and predicted (i.e., the results generated by classifiers) binary outputs for constructing and testing, several, say s , classifiers, K_1, \dots, K_s , each of the form

$$Z'_t = K(X_1, \dots, X_k). \tag{4}$$

The mathematical model described above was implemented in a software system written in FORTRAN. The reason for using this old programming language was twofold. First, because of a great amount of needed computations the usage of popular among statisticians interpreted languages like R is completely inefficient. Second, because of the long history of the usage of this programming language in statistics many numerically effective procedures are widely available.

B. Description of the experiment

In this paper, we describe the results for only four input variables. This restriction was due to limited time of computations. One has to note that even in this restricted model one run of Monte Carlo simulations may last several days of continuous work of a fast PC computer. The simulation process described in this paper consists of three parts. First, a stream of data points, i.e., the values of predictors, the values of hidden variables, the value of the unobserved auxiliary output variable, and the observed output binary variable are generated. Next, these simulated data serve as training data sets for building several classifiers. Finally, test data sets are generated, and used for the evaluation of considered classification (prediction) algorithms.

In our simulation experiment the probability distributions of predictors defined by a user on the first level of the model can be chosen from a set of five distributions: uniform, normal, exponential, Weibull, and log-normal. For the second level of the model a user can choose the probability distributions of the hidden variables from a set of distributions which are defined on the positive part of the real line: exponential, Weibull, and log-normal. The dependence between the pairs of predictors, and between predictors and associated hidden variables, can be described by the following copulas: independent, normal, Clayton, Gumbel (only positive dependencies), Frank, and FGM (only weak dependencies). The detailed description of these copulas can be found, e.g., in [9]. The strength of this dependence is defined by the value of Kendall's coefficient of association τ . The expected values of the distributions of the hidden variables in this simulation model depend in a linear way on the values of its related predictors. At the next stage of simulation, hidden random variables are transformed to the auxiliary output random variable T . The relation between the

hidden variables and T is strongly non-linear, and is described by operators of a "min-max" type. Finally, the auxiliary output random variable T is transformed to the binary output variable which is used for classification purposes. The proposed model allows to generate data with great variety of properties (non-linear dependence of a different strength, different probability distributions, etc.) that are significantly different from those usually assumed for linear regression models.

The scheme of the simulation of a data point, for an exemplary set of input parameters (probability distributions, copulas, and values of Kendall's τ), is presented in Figure 1. The values of four input attributes are generated, respectively, from the normal, exponential, logarithmic normal, and Weibull distributions. The generated values are statistically dependent, and the dependencies are described, respectively, by the following copulas: Clayton (with $\tau = 0.8$), Normal (with $\tau = -0.8$), and Frank (with $\tau = 0.8$). Then, for each input attribute the system generates an unobserved (hidden) value. These hidden values are generated, respectively, from the logarithmic normal, exponential, exponential, and Weibull distributions. The parameters of these distributions depend in a linear way upon the values of the respective input attributes (this dependence is not depicted in Figure 1). Moreover, they are also statistically dependent upon the values of the generated input attributes, and these dependencies are described, respectively, by the following copulas: Normal (with $\tau = -0.8$), Frank (with $\tau = 0.9$), Gumbel (with $\tau = -0.9$), and Normal (with $\tau = -0.8$), and Clayton (with $\tau = -0.8$). Finally, the real-valued output is calculated using the formula depicted in Figure 1, and this value is transformed, by using (1), to the binary output variable. The generated 5-tuple (4 input attributes, and a binary output value) describes one point in the training data set. The points of the test set are generated similarly, with the same or different (when robustness is evaluated) parameters of the model. The number of input variables (four) has been chosen in accordance with the opinion presented in [5] that in real situations the number of attributes which really influence quality characteristics of a classifier is usually small.

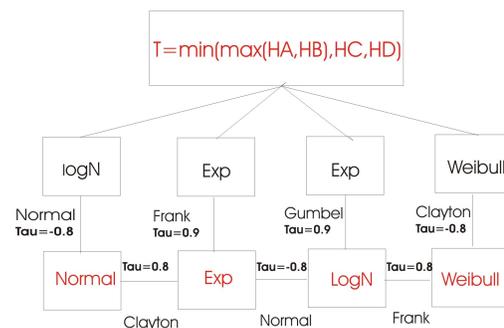


Figure 1. An exemplary scheme of the simulation of a data point

Several types of classifiers have been implemented in our simulation program. The classifiers are built using samples of size n_t of training data consisted of the vectors of the values of predictors (x_1, x_2, x_3, x_4) , and the actual value of the assigned class. In this paper, we consider only six of them which represent three different general approaches to the classification problem.

Binary linear regression. The first considered classifier is a simple (of the first order) binary linear regression (LINREG4). We label the considered classes by 0 and 1, respectively, and consider these labels as real numbers, treating them as observations of a real dependent variable in the linear regression model of the following form:

$$R_4 = w_0 + w_1 * X_1 + w_2 * X_2 + w_3 * X_3 + w_4 * X_4, \quad (5)$$

where R is the predicted class of an item described by explanatory variables X_1, X_2, X_3, X_4 , and w_1, w_2, w_3, w_4, w_0 are respective coefficients of the regression equation estimated from a training set of n_t elements. The value of R estimated from (5) is a real number, so an additional requirement is needed for the final classification (e.g., if $R < 0,5$ an item is classified as belonging to the class 0, and to the class 1 otherwise). The second considered classifier is also a linear one, but with additional variables describing interactions of the second order between the input variables (LINREG14). The regression function (of the second order) in this case is the following

$$R_{14} = w_0 + w_1 * X_1 + \dots + w_5 * X_1^2 + \dots + w_9 * X_1 * X_2 + \dots + w_{14} * X_3 * X_4. \quad (6)$$

The main advantage of these two classifiers is their simplicity. Moreover, the classical linear regression is implemented in all spreadsheets, such as, e.g., MS Excel. For this reason we have chosen these classifiers as the easiest to implement in practice without any specialized software.

Logistic regression. The next two classifiers are built using a generalized linear regression model, namely the logistic regression. The logistic regression is recommended by many authors (see, e.g., [2]) as the best regression tool for the analysis of discrete data. In this model the dependence of the output R_L upon the input variables is modeled by the logistic function

$$R_L = \frac{1}{1 + \exp(-f(X_1, \dots, X_4))}, \quad (7)$$

where the function $f(X_1, \dots, X_4)$ is described either by the right side of (5) of the LOGREG4 model, or by the right side of (6) of the LOGREG14 model. Unfortunately, the implementation of the logistic regression is not as simple as in the case of the linear regression. The estimation of its parameters requires the usage of numerical procedures that are implemented in specialized software (available, e.g., in the WEKA package).

Linear Discriminant Analysis (LDA). The last two classifiers implement the LDA introduced by Fisher, and described in many textbooks on multivariate statistical analysis and data mining (see, e.g., [2]). This method is historically the first classification method used in practice, and according to [5] its efficiency has been proved empirically by many authors. In the LDA statistical data are projected on a certain hyperplane estimated from the training data. New data points, projected on this hyperplane, which are closer to the mean value of the projected on this hyperplane training data representing the class 0 than to the mean value of training data representing the remaining class 1 are classified to the class 0. Otherwise, they are classified to the class 1. The equation of the hyperplane is given by the following formula:

$$L = y_1 * X_1 + y_2 * X_2 + y_3 * X_3 + y_4 * X_4 + y_0, \quad (8)$$

where L is the value of the transformed data point calculated using the values of the explanatory variables X_1, X_2, X_3, X_4 , and y_1, y_2, y_3, y_4, y_0 are respective coefficients of the LDA equation estimated from a training set. If Z_L denote the decision point, a new item is classified to the class 0 if $L \leq Z_L$, and to the class 1 otherwise. The LDA may not perform well in the case of unbalanced data. Therefore, in our simulation we implemented two methods of the calculation of Z_L . First, the classical one (LDA-SYM), when this point is just the average of the mean values of the transformed data points from the training set that belonged to the class 0 and the class 1, respectively. Second, an asymmetric one (LDA-ASYM), recommended for the analysis of unbalanced data sets, where Z_L is located asymmetrically between the two mean values mentioned above, depending upon the number of items belonging to each class in the test set. The calculation of the LDA equation (8) is not so simple. However, it can be done using basic versions of many statistical packages such as SPSS, STATISTICA, etc. Moreover, the LDA problem can be reformulated in terms of a simple linear regression, so the statistical tools available in spreadsheets may also be used for computations.

III. EVALUATION OF BINARY CLASSIFIERS

Proper evaluation of binary classifiers is not as simple as it looks like. If we do not consider any costs of misclassification the whole information about the quality of classifiers is contained in the so called confusion matrix, presented in Table I [4].

TABLE I. CONFUSION MATRIX

	Pred_Negative	Pred_Positive	
Act_Negative	True negative (TN)	False positive (FP)	N=TN+FP
Act_Positive	False negative (FN)	True positive (TP)	P=FN+TP

All measures of the quality of classifiers are built using the information contained in this matrix. A comprehensive overview of these measures can be found in many sources such as, e.g., Chapter 3 of the book by Japkowicz and Shakh [4]. The most frequently used measure is *Accuracy* ($= (TN + TP)/(N + P)$). It estimates the probability of correct classification. However, in certain circumstances (e.g., when classes are unbalanced) this measure does not let to discriminate the quality of different classifiers. This happens to be the case in experiments described in this paper. Other popular and important measures, such as *Precision* ($= TP/(TP + FP)$), *Sensitivity* or *Recall* ($= TP/(TP + FN)$), and *Specificity* ($= TN/(FP + TN)$), describe only certain features of binary classifiers. For example, high values of *Precision* in statistical terms are equivalent to low values of type I classification error when ‘‘Positives’’ are considered as the relevant class. Similarly, high values *Sensitivity* in statistical terms are equivalent to low values of type II classification error. When quality of the classification of ‘‘Negatives’’ is also worth of consideration, one has to take into account the value of *Specificity*. In the performed experiment we used all these measures for the evaluation purposes. However, in this paper, due to its limited volume, we present the analysis of an aggregate measure named the *F1 score* (or *F1 measure*), defined as the harmonic average of *Precision* and *Sensitivity*. Low values of this measure indicate that a classifier has a large value of at least one of type I or type II errors.

IV. RESULTS OF EXPERIMENTS

The simulation system described in Section II was used in many experiments with the aim to evaluate different binary classifiers. In this paper, we describe only one of them. In each instance of this particular experiment we simulated 50 runs, each consisted of one training set of 100 elements and 100 test sets of 1000 elements each. This small size of a training set was chosen in order to compare the results of simulations with those described in [6], [7], where it had a particular practical meaning. In each instance of the experiment, we used the same type of a copula for the description of all dependent random variables (in other experiments, not described in this paper, we used different copulas in one considered model). The strength of dependence was categorized into 6 categories: strong positive (Sp), medium positive (Mp), weak positive (Wp), weak negative (Wn), medium negative (Mn), and strong negative (Sn). For the Sp category the value of Kendall's τ was randomly chosen for each training set from the interval [0.7, 0.9]. The respective intervals for the remaining categories were the following: [0.4, 0.6] for Mp, [0., 0.2] for Wp, [-0.2, 0.] for Wn, [-0.6, -0.4] for Mn, and [-0.9, -0.7] for Sn. For each of the simulated 50 training sets the expected values of input variables (predictors) varied randomly in certain intervals. The simulated training sets were used for the construction of six classifiers described in Section II. For all test sets in one simulation run the description of the dependence between considered random variables (i.e., the copula, and the set of the values of Kendall's τ) was the same as in the respective training set. However, in choosing the expected values of the input variables (predictors) we considered two cases. In the first case, these expected values were the same as in the training set. Thus, the test sets were simulated using the same model as the respective training set. In other words, the considered classifiers were evaluated, in this case, on data generated by the same model as it had been used for their construction. In the second case, the expected values of the input variables used in the generation of test sets were *different* than the values used in the generation of the respective training sets. Those different values were chosen randomly around the values used for the generation of the training sets (by maximum $\pm 30\%$).

The presentation of the obtained results let us start with the analysis of the influence of the type of a copula describing the type of dependence on the *Accuracy* of considered classifiers. In Table II we present the obtained average values of *Accuracy* for 4 different copulas, and the strength of dependence belonging to the category Mp. We can see that the quality of the considered classifiers for a given copula is similar. Only the asymmetric LDA classifier is visibly worse. However, this quality is different for different types of copulas. This seems to be a very important finding, as the type of dependence is rarely (if ever) considered in the evaluation of classifiers. In the case described in Table II the observed (marginal) probability distributions are the same, and the estimates of the strength of dependence are also the same. Nevertheless, the accuracy of classification is visibly different, depending upon the type of dependence defined by the respective copula.

The situation becomes different when we use the *F1 score* for the evaluation of considered classifiers. The results of such evaluation (averaged for the same data!) are presented in Table III. First of all, we can see unacceptably low values of the *F1*

TABLE II. AVERAGE ACCURACY. THE SAME MODEL FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.769	0.835	0.741	0.752
LINREG14	0.769	0.833	0.735	0.751
LOGREG4	0.789	0.849	0.757	0.773
LOGREG14	0.765	0.833	0.739	0.747
LDA-SYM	0.741	0.765	0.729	0.729
LDA-ASYM	0.697	0.732	0.683	0.685

TABLE III. AVERAGE F1 SCORE. THE SAME MODEL FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.358	0.561	0.266	0.324
LINREG14	0.490	0.623	0.419	0.472
LOGREG4	0.537	0.662	0.464	0.509
LOGREG14	0.368	0.577	0.281	0.336
LDA-SYM	0.101	0.061	0.072	0.089
LDA-ASYM	0.549	0.598	0.484	0.562

score for the symmetric LDA classifier. Despite its quite good accuracy (see Table II) classification errors of this classifier are completely imbalanced. As the matter of fact, the precision of this classifier was good, but its sensitivity was really very low. The variability of the *F1* score observed in Table III is much greater than the variability of the *Accuracy*. It means that for different copulas the quality of considered classifiers measured by the *F1 score* may be significantly different. Moreover, if we look simultaneously on Tables II– III, we can see that the simple logistic regression classifier seems to be quite visibly the best when it classifies data generated by the same model as it had been used for the generation of the training set.

Let us now consider an interesting case when the model of data in test sets is *different* from that of training data. In reality, it means that a classifier is used on data described by a different probability distribution than the data used during its construction. In Tables IV– V we present average values of *Accuracy* and *F1 score* when the expected values of the input variables in the test sets have been randomly shifted around the values used for the generation of the training sets (by maximum $\pm 30\%$).

TABLE IV. AVERAGE ACCURACY. DIFFERENT MODELS FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.730	0.773	0.705	0.728
LINREG14	0.678	0.741	0.670	0.687
LOGREG4	0.759	0.809	0.725	0.749
LOGREG14	0.700	0.747	0.687	0.697
LDA-SYM	0.745	0.775	0.730	0.731
LDA-ASYM	0.643	0.672	0.631	0.646

TABLE V. AVERAGE F1 SCORE. DIFFERENT MODELS FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.317	0.440	0.259	0.300
LINREG14	0.443	0.519	0.356	0.420
LOGREG4	0.482	0.577	0.406	0.451
LOGREG14	0.379	0.492	0.282	0.338
LDA-SYM	0.138	0.126	0.101	0.124
LDA-ASYM	0.470	0.524	0.411	0.484

As we can expect, the values of quality indices in this case are lower in comparison to the case when training and test

data are described by the same probability distributions. The relative changes of their values are presented in Tables VI–VII for *Accuracy* and *F1 score*, respectively.

TABLE VI. RELATIVE CHANGE OF ACCURACY DUE TO DIFFERENT MODELS FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.950	0.925	0.950	0.968
LINREG14	0.883	0.889	0.911	0.914
LOGREG4	0.962	0.953	0.957	0.969
LOGREG14	0.915	0.896	0.930	0.933
LDA-SYM	1.004	1.012	1.001	1.003
LDA-ASYM	0.923	0.918	0.924	0.943

TABLE VII. RELATIVE CHANGE OF F1 SCORE DUE TO DIFFERENT MODELS FOR TRAINING AND TEST SETS

Classifier	Normal	Clayton	Gumbel	Frank
LINREG4	0.887	0.785	0.974	0.925
LINREG14	0.904	0.834	0.850	0.890
LOGREG4	0.897	0.872	0.875	0.886
LOGREG14	1.028	0.852	1.003	1.005
LDA-SYM	1.369	2.057	1.396	1.396
LDA-ASYM	0.855	0.876	0.848	0.862

The analysis of the robustness of the considered classifiers to an unexpected change of the underlying model of observed data is not simple and unequivocal. The simple logistic regression classifier (LOGREG4) still seems to be the best, but its loss of efficiency is not the best one.

Finally, let us consider the problem how the strength of dependence influences the robustness of classifiers to an unexpected change of the underlying model of observed data. We will illustrate this problem on the example of the LOGREG4 classifier which seems to be the best from among all classifiers considered in this paper. It seems to be quite obvious that there exists a general rule that “the stronger dependence (positive or negative) the better classification”. However, the relationship between the strength and type of dependence and the quality of classification may be not so simple. In Table VIII, we show how the values of the *F1 score* are changing for different copulas and different strengths of dependence.

TABLE VIII. AVERAGE F1 SCORE FOR LOGREG4 CLASSIFIER. THE SAME MODEL FOR TRAINING AND TEST SETS. DIFFERENT LEVELS OF THE STRENGTH OF DEPENDENCE

Dependence	Normal	Clayton	Gumbel	Frank	FGM
Sp	0.799	0.866	0.813	0.813	X
Mp	0.537	0.662	0.463	0.509	X
Wp	0.041	0.062	0.044	0.052	0.052
Wn	0.088	0.060	X	0.056	0.056
Mn	0.424	0.333	X	0.379	X
Sn	0.763	0.647	X	0.730	X

The results displayed in Table VIII reflect the complexity of the stated problem. First of all, the quality of classification strongly depends upon the type of dependence described by a respective copula. Only in the case of the normal (Gaussian) copula (the classical multivariate normal distribution is a particular case of a distribution described by this copula) the relationship between the strength of dependence and the quality of classification (measured by the *F1 score*) is symmetric. For the remaining copulas this relationship is visibly asymmetric (negative dependence leads to worse classification), and the

values of the *F1 score* may be quite different despite the same strength of dependence.

When the data in the test sets are generated by different models than in the training sets the values of the *F1 score* are changing. This is illustrated in Table IX for the case of the LOGREG4 classifier.

TABLE IX. AVERAGE F1 SCORE FOR LOGREG4 CLASSIFIER. DIFFERENT MODELS FOR TRAINING AND TEST SETS. DIFFERENT LEVELS OF THE STRENGTH OF DEPENDENCE

Dependence	Normal	Clayton	Gumbel	Frank	FGM
Sp	0.642	0.618	0.633	0.625	X
Mp	0.482	0.577	0.406	0.451	X
Wp	0.069	0.089	0.064	0.077	0.076
Wn	0.110	0.082	X	0.074	0.075
Mn	0.399	0.320	X	0.358	X
Sn	0.648	0.558	X	0.633	X

For strong and medium positive dependencies the strongest worsening of quality of classification has been observed when data are described by the Clayton copula. However, when dependencies are negative, the case of the Normal copula seems to be the worse. It is also surprising that for weak dependencies the values of the *F1 score* have even improved. It shows that in such cases this quality index is rather inappropriate as the results of classification to great extent seem to be random, as it is the case when dependencies are weak.

V. CONCLUSIONS

In the paper we have evaluated six binary regression type classifiers. For the comparison we used two measures of quality: the *Accuracy* (i.e., the probability of correct classification), and the *F1 score* which is the harmonic average of *Precision* (equal one minus the probability of type I error) and *Sensitivity* (equal one minus the probability of type II error). The evaluation was performed using a complex simulation software that allowed to model strongly nonlinear dependencies of different types (described by different copulas) and different strength (measured by Kendall’s τ). The distinctive feature of this research is taking into consideration a practical problem when objects classified by a certain classifier are described by a different probability distribution than the objects used for building (training) this classifier.

The performed experiments revealed that the quality of classification is strongly related to the type of dependence (type of the respective copula). This relationship may have different impact on the performance of different classifiers. For example, a simple linear regression classifier is quite robust to the change of the data model when the data are described by the Gumbel copula, but not robust when the data are described by the Clayton copula, even if the strength of dependence is in both cases the same. The performed experiments do not reveal unquestionable superiority of anyone of the considered classifiers. However, classifiers based on linear and logistic regressions are better than those based on Fisher’s linear discrimination. If we take into account both the quality of classification and the robustness to the change of the underlying model, *the classifier based on a simple (without interactions) logistic regression is the best one*. This could serve as the general recommendation for practitioners. However, when some additional information is available, other classifiers could be preferred. For example, if we know that

input attributes are dependent, and their dependence is described by the Frank copula, then the LDA classifier with an asymmetric decision criterion would be preferred. In practice, however, obtaining such specific information seems to be rather unlikely, so our general recommendation seems to be valid for the great majority of practical cases.

REFERENCES

- [1] R. Duda, P. Hall, and D. Stork, *Pattern Classification*, 2nd Edition. Wiley, 2000.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction*, 2nd Edition. New York: Springer, 2009.
- [3] I. Witten, E. Frank, and F. Hall, *Data Mining. Practical Machine Learning Tools and Techniques*, 3rd Edition. Morgan Kaufman, 2011.
- [4] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms. A Classification Perspective*. New York: Cambridge University Press, 2015.
- [5] D. Hand, "Classifier Technology and the Illusion of Progress", *Statistical Science*, vol. 21, pp. 1–14, 2006
- [6] O. Hryniewicz, "SPC of processes with predicted data: Application of the Data Mining Methodology", in *Frontiers in Statistical Quality Control 11*, S. Knoth and W. Schmid, Eds., Heidelberg: Springer, 2015, pp. 219–235.
- [7] —, "Process inspection by attributes using predicted data", in *Challenges in Computational Statistics*, S. Matwin and J. Mielniczuk, Eds., Cham: Springer, 2016, pp. 113–134.
- [8] P. Embrechts, F. Lindskog, and A. McNeil, "Modelling Dependence with Copulas and Applications to Risk Management", in *Handbook of Heavy Tailed Distributions in Finance*, S. Rachev, Ed., Amsterdam: Elsevier, 2003, ch. 8, pp. 329–384.
- [9] R. Nelsen, *An Introduction to Copulas*. New York: Springer, 2006.
- [10] O. Hryniewicz, and J. Karpiński, "Prediction of reliability - pitfalls of using Pearsons correlation", *Eksploracja i Niezawodność - Maintenance and Reliability*, vol. 16, pp. 472–483, 2014.

Bio-inspired Design of High-speed Transmission Line

High Signal Integrity Design for Printed Circuit Board Traces in GHz Domain

Moritoshi Yasunaga

Graduate School of Systems and Information Engineering
 University of Tsukuba
 Tsukuba, Japan
 e-mail: yasunaga@cs.tsukuba.ac.jp

Ikuo Yoshihara

Faculty of Engineering
 Miyazaki University
 Miyazaki, Japan
 e-mail: yoshiha@cs.miyazaki-u.ac.jp

Abstract—Regarding Signal integrity (SI) degradation problem in printed circuit boards, conventional design techniques based on the impedance-matching theory cannot work any longer in the GHz-domain. In this paper, we propose a novel bio-inspired SI improvement design methodology using genetic algorithms. We apply our proposed methodology to real memory bus systems and demonstrate its effectiveness using prototypes.

Keywords—Genetic Algorithms; Transmission Line; Signal Integrity; Printed Circuit Board.

I. INTRODUCTION

Signal Integrity (SI) degradation is one of the most serious problems in the printed circuit board (PCB) in the GHz-era [1] [2]. Figure 1 shows one of waveform examples observed in DDR-3 memory bus system. Conventionally, a number of trace designs, which are based on the impedance matching theory, have been used in the MHz-domain. The conventional designs, however, are becoming ineffective as the frequency increases and cannot work any longer in the GHz-domain.

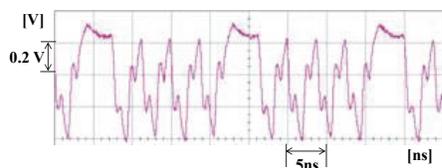


Figure 1. Distorted distal waveform in memory-bus system.

In Section II, we propose a novel trace (transmission) structure that can overcome the SI degradation problem. In Section III, we describe its design methodology based on the genetic algorithms (GAs) and details of GA operations. In Section IV, we demonstrate its effectiveness showing one of prototypes for memory-bus prototypes.

II. BIO-INSPIRED TRANSMISSION LINE STRUCTURE

In the proposed trace structure, which we call it segmental transmission line (STL), a transmission line is divided into multiple (N) segments of individual characteristic impedance Z_i ($i = 1, 2, \dots, N$) as shown in Figure 2. And Z_i are adjusted to achieve an ideal digital waveform at important points, such as input points to the Large Scale Integrated Circuits (LSIs) on the line by

superposing reflection waves, which are generated at the interfaces between adjacent segments Z_i and Z_j . Figure 3 shows a bird-eye view of the STL in the PCB. Characteristic impedance Z is a function of trace width W , so that Z can be thus controlled by adjusting W .

The adjustment of all Z_i , or W_i , however, results in a combinatorial explosion problem. We thus have proposed to apply the GAs to solve this problem. The STL consists of one-dimensional array of Z_i , which is similar in structure to the genome, so that it can be easily and well mapped onto the chromosome in GAs as shown in Figure 4.

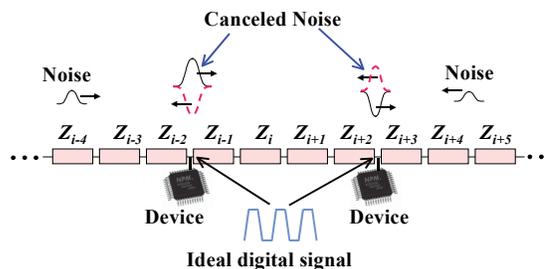


Figure 2. Principle of segmental transmission line (STL).

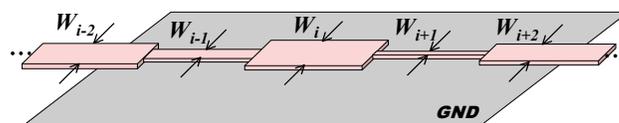


Figure 3. STL structure in PCB.

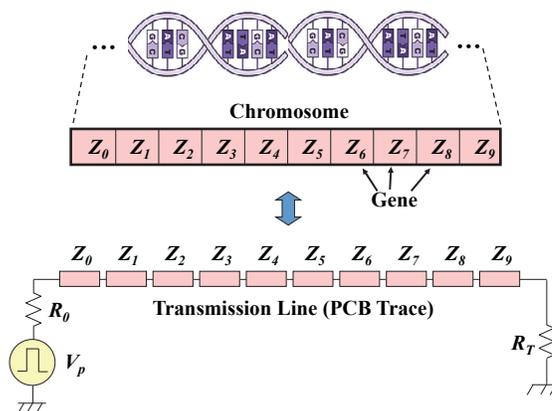


Figure 4. Mapping STL onto GA.

III. DESIGN METHODOLOGY BASED ON GA

A. Chromosome

In an earlier design, we used only the characteristic impedances Z_i as genes in a *simple chromosome* (see the upper section of Figure 5). As a modification of this early design, we proposed a *hybrid chromosome*, which was created by adding segments of length L_i as genes (see the lower section of Figure 5).

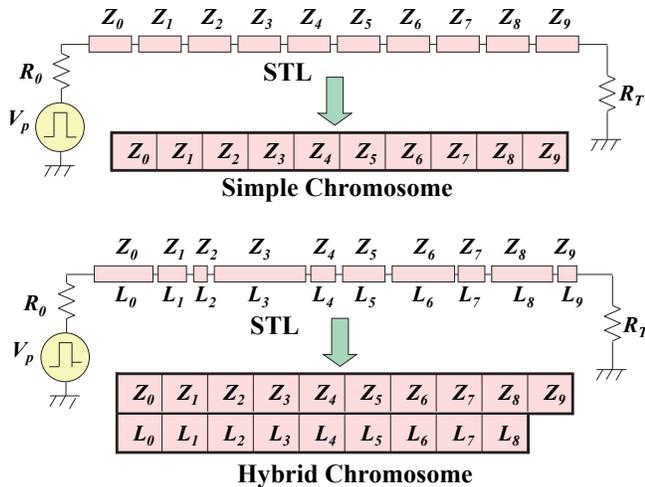


Figure 5. Simple chromosome (upper) and hybrid chromosome (lower).

The segment lengths L_i can adjust the timing of the superposition of the reflected waves, and thus we expect that the hybrid chromosomes result in a higher SI than that obtained from the simple chromosomes.

B. Crossover

It has been shown that, in the STL design, intervals of 1 Ω are sufficiently small that the Z_i can be treated as integer parameters. The Z_i are independent of each other in the segments, and thus no fatal gene can be generated in the simple crossover operations. In a simple chromosome or the Z_i part of a hybrid chromosome, genes can be easily exchanged by a simple crossover operation, as shown in Figure 6. In the figure, genes in chromosomes 1 and 2 are exchanged at point C_p .

On the other hand, the crossover operation for the segment length L_i is not as easy as it is for Z_i , because there is a strict condition that the sum of all segment lengths L_i is fixed as the trace length. Furthermore, the timing of the superposition of the waveforms is very sensitive to the propagation time. We thus have to treat the L_i s as real number genes with a fixed trace length.

Figure 7 shows a newly proposed crossover operation for the L_i genes; it is based on the BX- α crossover [3][4]. For example, a 20 cm trace is divided into 5 segments, and before the crossover operation, the segment lengths L_i are normalized and changed to the boundaries A1 to A4 in chromosome 1 and B1 to B4 in chromosome 2.

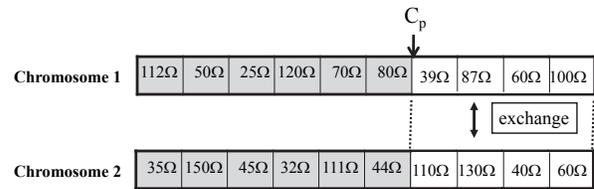


Figure 6. Simple crossover for characteristic impedance.

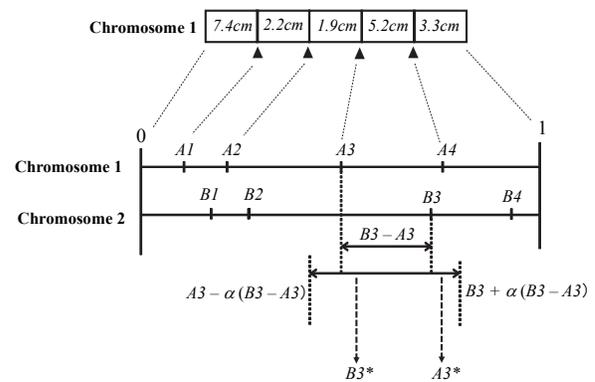


Figure 7. BX- α crossover for segment length.

Two boundaries, say, A3 and B3, which have no other boundaries between them, are then chosen at random. And the range B3 to A3 is then expanded by α percent ($10 \leq \alpha \leq 20$). Finally, two new boundaries, A3* and B3*, are chosen randomly in the expanded range, and they are changed from their previous lengths. This expanded BX- α crossover has worked well in practical applications and has found excellent solutions, which will be discussed below.

C. Fitness evaluation

For the periodic (clock) signals propagating in the PCB traces, each chromosome was scored based on half of the periodic waveform simulated using the SPICE circuit simulator, as shown in Figure 8 (see equations in the figure also). The reciprocal of the difference area $Diff$ between the ideal waveform $I(t)$ and the waveform $R(t)$ was used as the score (fitness), so that the score increased as the waveform improved or approached the ideal waveform.

For the random (data) signals, we used a long periodic signal of 1000000...1000000... as shown in Figure 9; this can be regarded as an impulse input. If the ideal impulse propagates in the trace, impulse response theory guarantees a high SI. Thus, in the STL design for the random signals, the reciprocal of the difference area $Diff$, including the reflection wave, works well as the score (fitness); see Figure 9.

IV. PROTOTYPE AND EXPERIMENTAL RESULTS

We have applied the STL to some high-speed digital data transfer systems. Figure 10 shows one of design examples, which is used in the Double Data Rate (DDR) memory bus. In the STL design, we used a set of characteristic impedances from 30 Ω to 120 Ω at intervals of 5 Ω intervals.

The measured waveforms in the scale-up prototypes are shown in Figures 11 and 12.

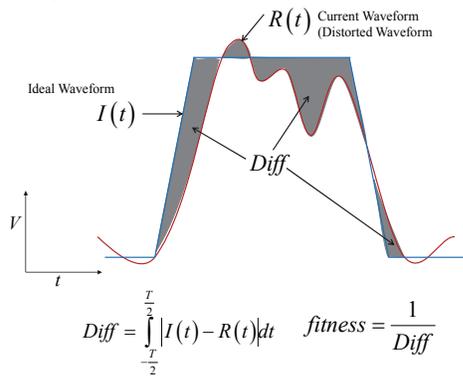


Figure 8. Fitness evaluation for periodic signals.

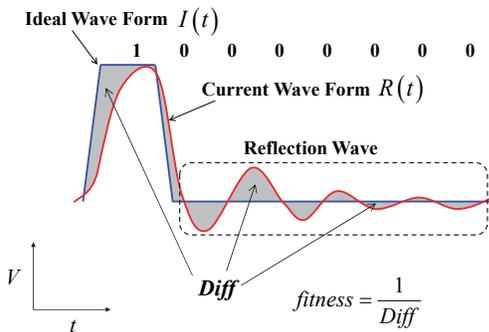


Figure 9. Fitness evaluation for random signals.

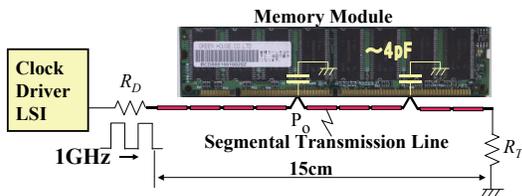


Figure 10. DDR memory bus system and STL.

In Figure 11, the periodical waveform in the conventional transmission line is seriously distorted with some reflection waves and it cannot be used as the clock signal in real systems. In the STL, the distorted waves are well improved and it is almost the same as the ideal clock signal.

In Figure 12, the eye diagram, which is used to evaluate SI in random digital signals, in the conventional transmission line is dramatically distorted, and its aperture is close to 0.2 V high and 1.1 ns wide, which is not of practical use.

In contrast to the conventional transmission line, the eye diagram in the STL clearly opens to a height of 1.1 V and a width of 1.3 ns, which is sufficiently large to be used in practice.

V. CONCLUSIONS

A novel PCB trace structure and its bio-inspired design methodology are proposed in order to overcome SI digiration problem in GHz-domain. Some remarkable SI

improvement results were demonstrated in the real memory bus prototype fabricated by the proposed methodology.

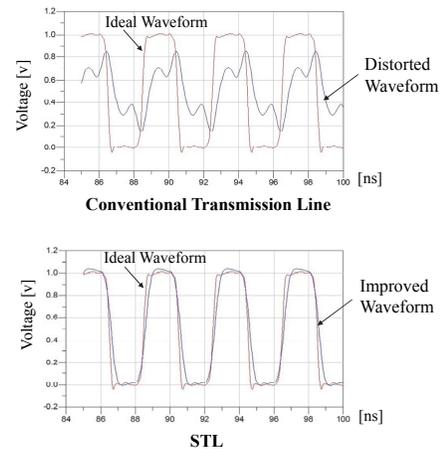


Figure 11. Clock signals observed in conventional transmission line and STL.

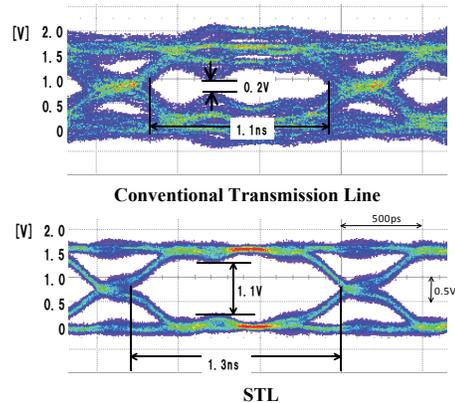


Figure 12. Eye-diagrams observed in conventional transmission line and STL.

ACKNOWLEDGMENT

This research was partially supported by JSPS/KAKENHI grant number 26289114 in Japan.

REFERENCES

- [1] M. P. Li, "Jitter, Noise, and Signal Integrity at High-Speed," 1st ed., Prentice Hall Signal Integrity Library, 2007.
- [2] E. Bogatin, "Signal and Power Integrity-simplified," 2nd ed., Prentice Hall Signal Integrity Library, 2010.
- [3] L. J. Eshelman, "Real-Coded Genetic Algorithms and Interval-Schemata," Foundations of Genetic Algorithms 2, 1993, pp. 187-202.
- [4] L. J. Eshelman, K. E. Mathias, and J. D. Schaffer, "Crossover Operator Biases: Exploiting the Population Distribution," Proc. ICGA97, 1997, pp. 354-361.

Evaluation and Monitoring for Disaster Management

Alexander Ryjov

Department of Mechanics and Mathematics
Lomonosov' Moscow State University
Moscow, Russia
e-mail: ryjov@intsys.msu.ru

Abstract—This article describes the main ideas of disaster management based on technology for status evaluation and progress monitoring of complex processes. Our work analyzes features of the disaster management process and the characteristics of the available information. The article also discusses the technological capabilities for evaluating and monitoring complex processes to support decisions at the stage when urgent assistance is needed.

Keywords-disaster management; technology for evaluation and monitoring of complex processes; perception-based descriptions; fuzzy hierarchical systems.

I. INTRODUCTION

Disaster management has become an important task for every country. For example, in 2012 (an average year in terms of disasters) we have the following statistics [1]:

- There were 905 natural catastrophes worldwide, 93% of which were weather-related disasters.
- Overall costs were US\$170 billion and insured losses \$70 billion.
- 45% were meteorological (storms), 36% were hydrological (floods), 12% were climatological (heat waves, cold waves, droughts, wildfires) and 7 % were geophysical events (earthquakes and volcanic eruptions).
- Between 1980 and 2011 geophysical events accounted for 14% of all natural catastrophes

Figures 1 – 3 [2] show us the trends:

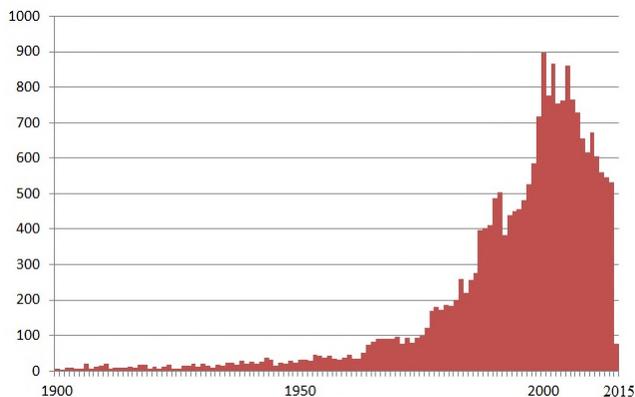


Figure 1. Number of disasters reported between the years 1900-2015.

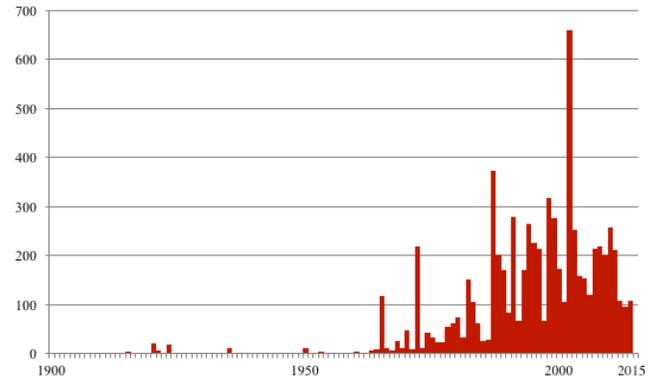


Figure 2. Number of people (in millions) reported affected by natural disasters between the years 1900-2015.

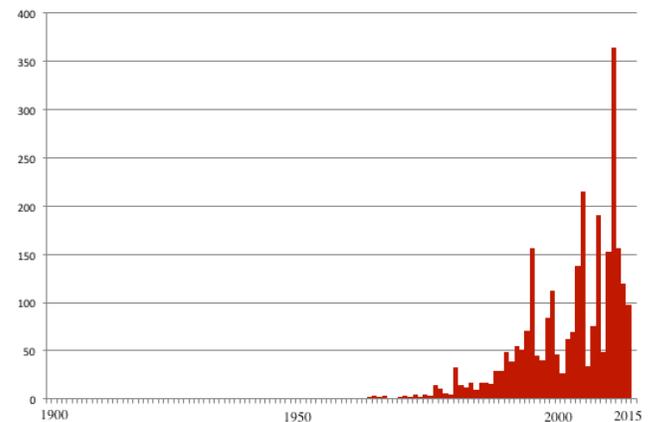


Figure 3. Estimated damages (in \$US billions) caused by reported natural disasters between the years 1900-2015.

This problem is a focus for international (The United Nations Disaster Assessment and Coordination - UNDAC [18]), regional (for example, Emergency Response Coordination Centre European Commission - ERCC [19]), and national organizations (the majority of countries have a Ministry of Emergency Situations, or similar).

Information and communications technologies can help disaster managers quickly access, contextualize, and apply near real-time information, improving the speed and effectiveness of critical actions such as warning the population at risk – examples of such information and communications technologies can be found on the web sites

mentioned above. In this article, we focus on the intellectualization of this type of systems.

The rest of this paper is organized as follows: Section II exposes the main features of disaster management systems and the main characteristics of the available information. Section III provides a description of the technology used for evaluation and monitoring of complex processes. This technology allows us to process information that is uncertain, fragmentary, and variable over time. In conclusion, we collect pro and contra arguments for the applicability of the technology for disaster management.

II. DISASTER MANAGEMENT CHALLENGES

In recent years, the field of disaster studies has emerged as a new academic field [3] [4] [5]. Specialization in disaster management includes all three stages (before, during and after) – risk analysis and contingency planning, management and coordination during the event, and a disaster recovery plan. The most important and dangerous type of disasters is Large Scale Sudden Disasters (LSSD) [3]. LSSD management stages are presented in Figure 4.

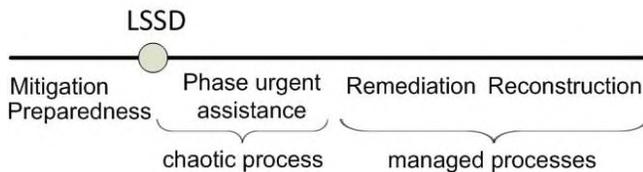


Figure 4. LSSD management stages.

The Remediation and Reconstruction stages are controllable enough, and modern information and communications technologies are capable to support these stages. The Urgent assistance stage is the main challenge from this point of view. The Urgent Assistance Phase has maximal uncertainty and maximal importance. We can single out the following top-5 characteristics for the Urgent Assistance Phase:

- Poor forecast – unexpectedness or short time for preparation
- Lack of information about the situation – huge uncertainty
- Lack of information about the available forces and resources – huge uncertainty
- The situation changes very fast – very short time for decision making
- High cost for wrong decisions

Based on this characterization, we can assert that the current status of decision support systems does not allow us to develop such a system for the Urgent Assistance Phase. The challenges are:

- Rationalize decision making for disaster preparedness
- Provide a base for vulnerability assessment and priority setting
- Provide monitoring and control for the Urgent Assistance Phase

- Time pressure
- Uncertainty
- Fast changes

III. TECHNOLOGY FOR EVALUATION AND MONITORING OF COMPLEX PROCESSES

Systems for Evaluation and Monitoring of Complex Processes (SEM) relate to a class of hierarchical fuzzy discrete dynamic systems. The theoretical base of such class of systems is made by the fuzzy sets theory, discrete mathematics, methods of the analysis of hierarchies which was developed by Zadeh [16] [17], Messarovich [7], Saaty [15] and others. SEM process uniformly diverse, multi-level, fragmentary, unreliable, and varying in time information about complex processes. Based on this type of information, SEM monitor the process' evolution and work out strategic plans of process development. These capabilities open a broad area of applications in business (marketing, management, strategic planning), socio-political problems (elections, control of bilateral and multilateral agreements, terrorism), etc. One of such applications is a system for monitoring and evaluating a state's nuclear activities (department of safeguards, International Atomic Energy Agency (IAEA)) [8]. This application is briefly described in this article.

A. Basic elements of SEM and their characteristics

The basic elements of SEM at the top level are the information space, in which information about the state of the process circulates, and the expert(s) who are working with this information and making conclusions about the state of the process and forecasts of its development (Figure 5).

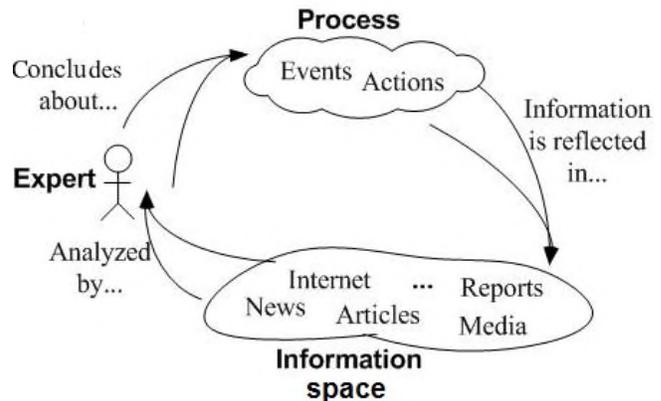


Figure 5. Basic elements of SEM.

The information space represents a set of various information elements, which can be characterized as follows:

- Diversity of the information carriers, i.e., presentation of the information in the articles, newspapers, computer kind, audio- and video-information, etc.;
- Fragmentary. The information most often related to any fragment of a problem, and the different fragments may be differently "covered" with the information;

- Multi-levels of the information. The information can concern the whole problem, some of its parts, or a particular element of the process;
- Various degree of reliability. The information can contain the particular data which has a various degree of reliability, indirect data, results of conclusions on the basis of the reliable information or indirect conclusions;
- Possible discrepancy. The information from various sources can coincide, differ slightly or be contradictory;
- Varying in time. The process develops over time, therefore, the information about the same element can be different between two different moments in time;
- Possible bias. The information reflects certain interests of the source of the information; therefore it can have tendentious character.

The experts are an active element of the monitoring system and, observing and studying elements of the information space, they draw conclusions about the state of the process and prospects of its development taking into account the properties listed above about the information space.

B. Basic principles of technology for evaluation and monitoring of complex processes

Systems for evaluation and monitoring of complex processes allow:

- to process uniformly diverse, multi-level, fragmentary, unreliable, information varying in time;
- to receive evaluations of status of the whole process and/or its particular aspects;
- to simulate various situations in the subject area;
- to reveal "critical ways" of the development of the process. This means to reveal those elements of the problem, the small changes that may qualitatively change the status of the process as a whole.

Taking into account the given features of the information and specific methods of its processing, it is possible to declare the main features of the technology as follows:

- The system provides the facility for taking into account data conveyed by different information vehicles (journals, video clips, newspapers, documents in electronic form, etc.). Such a facility is provided by means of storage in a database of a system of references to an evaluated piece of information, if it is not a document in electronic form. If the information is a document in electronic form, then both the evaluated information (or part thereof) and a reference thereto are stored in the system. Thus, the system makes it possible to take into account and use in an analysis all pieces of information, which have a relationship to the subject area irrespective of the information vehicle.
- The system makes it possible to process fragmentary information. For this purpose, a considerable part of

the model is represented in the form of a tree/graph (Figure 6).

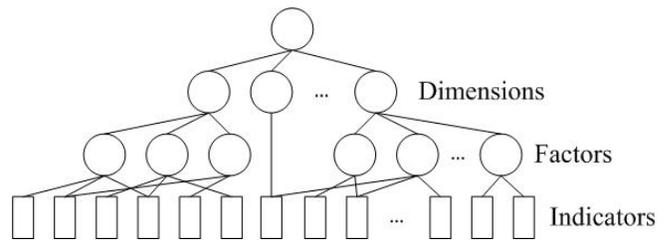


Figure 6. Structure of process model.

- Information with different degrees of reliability, some of it possibly tendentious, can be processed in the system. This is achieved by assessing the influence of a particular piece of information on the status of the elements of the model using fuzzy linguistic values.
- Time is one of the parameters of the system. This makes it possible to have a complete picture of the variation of the status of the model with time.

The summary of the information processing in SEM is presented in Figure 7.

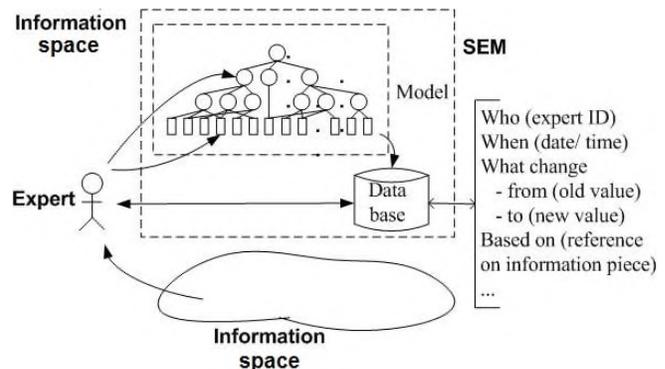


Figure 7. SEM workflow.

Thus, the systems constructed on the basis of this technology allow having the model of the process developing in time. It is supported by the references to all information materials, chosen by the analysts, with general and separate evaluations of the status of the process. Using the time as one of the parameters of the system allows to conduct the retrospective analysis and to build the forecasts of development of the process. There is the opportunity of allocation "of critical points", i.e., which are element(s) of the model for which a small change can cause significant changes in the status of the whole process. The knowledge of such elements has large practical significance and allows to reveal "critical points" of the process, to work out the measures on blocking out undesirable situations or work towards the achievement of desirable situations, i.e., somewhat operate the development of the process in time in the desirable direction.

C. Theoretical Basis

For effective practical application of the proposed technological solution, it is necessary to tackle a series of theoretical problems, the results of which are given below.

It is assumed that the expert describes the degree of inconsistency of the obtained information (for example, the readiness or potential for readiness of certain processes in a country [8]) in the form of linguistic values. The subjective degree of convenience of such a description depends on the selection and the composition of such linguistic values. Let us explain this on a model example.

Example [8]. Let it be required to evaluate the quantity of plutonium. Let us consider two extreme situations.

Situation 1. It is permitted to use only two values: “small” and “considerable quantity”.

Situation 2. It is permitted to use many values: “very small”, “not very considerable quantity”, ..., “not small and not considerable quantity”, ..., “considerable quantity”.

Situation 1 is inconvenient. In fact, for many situations both the permitted values may be unsuitable and, in describing them, we select between two “bad” values.

Situation 2 is also inconvenient. In fact, in describing a specific quantity of nuclear material, several of the permitted values may be suitable. We again experience a problem but now due to the fact that we are forced to select between two or more “good” values. Could a set of linguistic values be optimal in this case?

It is assumed that the system tracks the development of the problem, i.e., its variation with time. It is also assumed that it integrates the evaluations of different experts. This means that one object may be described by different experts. Therefore it is desirable to have assurances that the different experts describe one and the same object in the most “uniform” way.

On the basis of the above, we may formulate the first problem as follows:

Problem 1. Is it possible, taking into account certain features of the man’s perception of objects of the real world and their description, to formulate a rule for selection of the optimum set of values of characteristics on the basis of which these objects may be described? Two optimality criteria are possible:

Criterion 1. We regard as optimum those sets of values through whose use man experiences the minimum uncertainty in describing objects.

Criterion 2. If the object is described by a certain number of experts, then we regard as optimum those sets of values which provide the minimum degree of divergence of the descriptions.

It is shown that we can formulate a method of selecting the optimum set of values of qualitative indications (collection of granules [10]). Moreover, it is shown that such a method is stable, i.e., the natural small errors that may occur in constructing the membership functions do not have a significant influence on the selection of the optimum set of values. The sets which are optimal according to criteria 1 and 2 coincide. The results obtained are described in [12]. Following this method, we may describe objects with

minimum possible uncertainty, i.e., guarantee optimum operation of the SEM from this point of view.

Technology for evaluation and monitoring of complex processes assumes the storage of information material (or references to it) and their linguistic evaluations in the system database. In this connection the following problem arises.

Problem 2. Is it possible to define the indices of quality of information retrieval in fuzzy (linguistic) databases and to formulate a rule for the selection of such a set of linguistic values, the use of which would provide the maximum indices of quality of information retrieval?

In [12] was shown that it is possible to introduce indices of the quality of information retrieval in fuzzy (linguistic) databases and to formalize them. In [13] was shown that it is possible to formulate a method of selecting the optimum set of values of qualitative indications (collection of granules [13]) which provides the maximum quality indices of information retrieval. Moreover, in [12] [13] was shown that such a method is stable, i.e., the natural small errors in the construction of the membership functions do not have a significant effect on the selection of the optimum set of values. This proves that the offered methods can be used in *practical tasks* and can *guarantee optimum work of SEMs*.

Because the model of the process has a hierarchical structure, the choice and selection (tuning) of aggregation operators for the nodes of the model is one more important issue in the development of SEM. We may formulate this problem as follows:

Problem 3. Is it possible to propose the procedures of information aggregation in fuzzy hierarchical dynamic systems which allow us to minimize inconsistency in the model of process in SEM?

It is shown that it is possible to propose the following approaches based on different interpretations of aggregation operators: geometrical, logical, and learning-based. The last one includes leaning based on genetic algorithms and learning based on neural networks. These approaches are described in detail in [11].

D. Application’s features

Some applied information monitoring systems based on the technology described above has been developed. Based on this experience, we can formulate the following necessary stages of the development process:

- conceptual design;
- development of the demonstration prototype;
- development of a prototype of the system and its operational testing;
- development of the final system.

The most difficult point in the development process is the elaboration of the structure of the process model. In some well-developed areas (marketing, medicine), we used descriptions of the process from professional books and references (like [6]) as a draft of the model. We then coordinated this draft with the professional experts (conceptual design and development of the demonstration prototype stages), and “tuned” this improved draft during testing of the system (development of a prototype stage). Sometimes the process for monitoring is formalized enough

for application of information monitoring technology. An example of this situation is a state nuclear program evaluation procedure in IAEA [8]. The previously developed physical model of the nuclear fuel cycle was a good base for the model of the process in SEM. Based on this model, a prototype of the information monitoring system has been developed.

In order to develop a SEM for disaster management purposes, we need to develop the model of the process. The current status of the model is a result of the Mitigation Preparedness stage (Figure 4). We can understand the level of our readiness for disasters and our weak points (critical paths) for any time (in particular, for disaster momentum). Using critical paths, we can “calculate” the impact of different actions to the level of our readiness for disasters. After LSSD starts (urgent assistance stage) we can organize our work in the most effective and efficient manner because we have the actual status of our preparedness (from the previous stage). We can easily input changes into the situation, and see the result of those actions as a new situation.

The application of the technology for evaluation and monitoring of complex processes for one of the task in disaster management is presented in [14] (Case Study Risk-based evaluation Bridge under Flooding).

IV. CONCLUSION

Effective disaster management is an important task for all countries, regions, and the planet as a whole. The most dangerous disasters are large scale sudden disasters. For LSSD, the most important stage is the urgent assistance phase. Due to huge uncertainty and time pressure we cannot use standard information technologies for effectively supporting the analysts and decision makers during this stage. In this article, we have discussed an idea to use technology for evaluation and monitoring of complex processes for this stage. We have presented examples of successful applications of technology in similar areas (in terms of features of the available information).

SEMs are solving evaluation and monitoring task, allow user input of all available information in a “natural manner” and capable of:

- saving the history of process development,
- evaluating current status,
- modeling the future of process development.

SEMs are effective when:

- we do not have (cannot develop) a mathematical model of the process in the form of equations, automatas, etc.
- we have experts who are performing monitoring task.

We can develop SEM with minimal requirements for the task, when:

- it is possible to develop “semantic model” of the process in the form of set of concepts and their inter-dependencies
- we work with real information (we can learn or tune the system)

We can develop an optimal system in terms of:

- how easy it is for the user to input information (expert, analyst)
- co-ordination of estimations of users (experts, analysts)
- information support of processes of input information and modeling.

The capabilities described in this article represent useful information for disasters management centers.

REFERENCES

- [1] Natural Catastrophes in 2012 Dominated by U.S. Weather Extremes. Worldwatch Institute. Available from: <http://www.worldwatch.org/natural-catastrophes-2012-dominated-us-weather-extremes-0>, last accessed September 2015
- [2] EM-DAT: The OFDA/CRED International Disaster Database. Available from: www.emdat.be, last accessed September 2015
- [3] LSSD - Large Scale Sudden Disasters. The Homeland Security Academy at Zinman College, Wingate Institute. Available from: <http://security.wincol.ac.il/default.aspx?siteid=68&pageid=1813&lang=2>, last accessed September 2015
- [4] Online Certificate Course on Disaster Management. The International Federation of Red Cross and Crescent Societies (IFRC) and the Tata Institute for Social Sciences (TISS). Available from: <https://ifrc.org/fr/how-to-help/reseau-deformation/opportunit-es/online-certificate-course-on-disaster-management/>, last accessed August 2015
- [5] NEEDS - The First Conference in North European Emergency and Disaster Studies. Available from: <http://changingdisasters.ku.dk/calender/needs/>, last accessed September 2015
- [6] P. Kotler, Marketing Management (10th Edition). Prentice Hall, 1999, 784 p.
- [7] M.D. Messarovich, D. Macko, and Y. Takahara, Theory of hierarchical multilevel systems. Academic Press, N.Y.-London 1970 - 344 p.
- [8] A. Ryjov, A. Belenki, R. Hooper, V. Pouchkarev, A. Fattah and L.A. Zadeh, Development of an Intelligent System for Monitoring and Evaluation of Peaceful Nuclear Activities (DISNA), IAEA, STR-310, Vienna, 1998, 122 p.
- [9] A. Ryjov, “Estimation of fuzziness degree and its application in intelligent systems development,” Intelligent Systems. V.1, 1996, p. 205 – 216 (in Russian).
- [10] A. Ryjov, “Fuzzy Information Granulation as a Model of Human Perception: some Mathematical Aspects,” Proceeding of Eight International Fuzzy Systems Association World Congress 99, p. 82-86.
- [11] A. Ryjov, “On information aggregation in fuzzy hierarchical systems,” Intelligent Systems. V.6, 2001, p. 341 – 364 (in Russian).
- [12] A. Ryjov, The principles of fuzzy set theory and measurement of fuzziness. Moscow, Dialog-MSU Publishing, 1988, 116 p. (in Russian).
- [13] A. Ryjov, “Towards an optimal task-driven information granulation,” In: Information Granularity, Big Data, and Computational Intelligence. Witold Pedrycz and Shyi-Ming Chen (Eds.). Springer International Publishing Switzerland 2015, pp. 191-208. DOI: 10.1007/978-3-319-08254-7_9
- [14] J. Sunkpho, W. Wipulanusat, N. Kokkaew, and A. Ryjov, “Disaster Management based on Information Monitoring Technology,” Creative Construction Conference 2014 Proceeding, June 21-24 2014, Prague, pp. 515-521.

- [15] T.L. Saaty, The Analysis of the Hierarchy Process. Moscow, Radio and Swjaz, 1993 - 315 p. (in Russian).
- [16] L.A. Zadeh, "Fuzzy sets," Information and Control, 1965, v.8, pp. 338-353.
- [17] L.A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning. Part 1,2,3," Inform.Sci.8, 199-249; 8,301-357; 9,43-80 (1975).
- [18] The United Nations Disaster Assessment and Coordination (UNDAC). Available from: <http://www.unocha.org/what-we-do/coordination-tools/undac/overview>, last accessed September 2015 .
- [19] Emergency Response Coordination Centre (ERCC) European Commission. Available from: http://ec.europa.eu/echo/what/civil-protection/emergency-response-coordination-centre-ercc_en, last accessed September 2015

G-Form: A New Approach for Visual Interpretation of Deep Web Form as Galaxy of Concepts

Radhouane Boughammoura, Lobna Hlaoua and Mohamed Nazih Omri

Faculty of Sciences of Monastir, University of Monastir
Research Unit MARS, Monastir, Tunisia

E-mails: Radhouane.Boughammoura@gmail.com, Lobna1511@yahoo.fr, MohamedNazih.Omri@fsm.rnu.tn

Abstract—Deep Web is growing rapidly with multitude of devices and rendering capabilities. Despite the richness of Deep Web forms, their rendering methodology is very poor in terms of capacity of expression. Hence, the user has no indication about the richness of the query and the query capability when he interprets this interface. In this paper, we propose a new rendering approach of Deep Web forms which is easy to interpret by the user and reflects the exact meaning of the query. We have evaluated our algorithm on standard dataset and compared it to a well known state of the art algorithm. Our approach showed good performance with respect to standard measures.

Keywords- *Web Applications; Deep Web; Information Retrieval; Query Interface; Query Interpretation; Galaxy of Concepts; Pertinence of a Concept; Human Computer Interface; Visualization.*

I. INTRODUCTION

Deep Web is the part of the Web which is not reachable via hyperlinks [2][4][7][12][14][15]. It is hidden behind Web forms which give access to Deep Web databases. Information on databases is a really big treasure. More than 90% of the information from the Web comes from Deep Web. In addition, this information is very rich in terms of quality of service offered to internet users [12][13][14]. We aim by our job to reveal the Deep Web to novice internet users via new, simple, and easy-to-use Web forms.

A Web form is an information retrieval interface which give access to Deep Web data. It is a graphical representation of the query using a set of fields. Users form their query by visual interpretation of the meaning of the query interface. The design method of the Web form [1][3][5][6][10][11] is very important since it is the only source of inspiration for novice users in order to understand the meaning of the query. A bad interpretation leads to an incorrect query and hence restricts access to Deep Web services. In this paper, we focus on the design aspect of Web forms in order to offer to novice users easy- to-use forms.

In the Web form presented in Fig. 1.(a), we notice presence of white fields crossing the Web form horizontally. These fields are very important; they indicate the presence of semantic entities (or semantic concepts). A novice user may not pay attention to these fields.

We present below the relevance of the white fields.

Let us consider an example in which the user is not interested in non-stop flights but is searching for a flight with one stop to reach the destination. Suppose the user is searching for all flights having as destination "Tunis" offered by an airline company with one stop city. The user may give by mistake the destination city and the number of passengers and leave the departure city empty. This request will be wrong unless the user knows all stop cities for destination "Tunis". This is not evident and will be a burden for a novice user as he must formulate as many queries as there are stop cities.

In this paper, we present a new design methodology which simplifies the design of the Deep Web form. While our methodology preserves the query capability of the Web form, it removes the complexity of the query with an easy-to-use form containing all necessary and pertinent fields. The resulting form becomes very simple and, more importantly, semantically very rich.

The rest of the paper is organized as follows. Section II presents a brief review of related works. Section III explains the motivation of our new approach G-form. In Section IV, we detail the principle of the G-Form and our experiments will be presented in Section V. Section VI concludes the paper.

II. RELATED WORK

According to the literature, Z-form [9] is considered as the most used form in the Deep Web. Z-Form is a flat query where all fields are listed at the same level of granularity. The name Z-Form comes from the fact that the user reads Z-Form like reading lines in a paragraph: he begins by the first line, then the second, etc. This reading strategy resembles to the letter Z (see Figure 1.b).

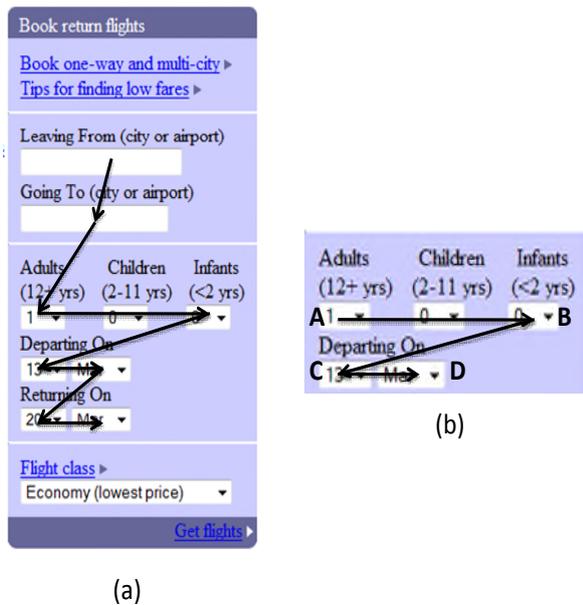


Figure 1. Z geometric pattern in a Z-Form

Z-Forms are the most used information retrieval query interface on the Web. Despite their reputation, Z-Forms have drawbacks. The first drawback is that all fields are rendered in the same interface. When the number of fields is large, the query becomes very complex and novice users find it difficult to formulate a correct query. We have seen that a line segment covers an entire line and forms hence one semantic entity of fields. However in many cases, more than one semantic entity may appear in one line. The Z-geometric pattern does not detect all semantic entities in the line, but considers the entire line as one semantic entity.

Ko-Chiu [20] proposes a new approach for effective surfing in the visualized interface of a digital library. This interface is designed for novice users (children). They found that information retrieval seeking of novice users is influenced by their curiosity and hindrance. Ko-Chiu studies the interactions and the usability of various search interfaces, and the enjoyment or uncertainty experienced by children when using virtual game-like interface. When novice users search for information, they have specific directions but they do not have a specific search target. These novice users have no special training or beliefs regarding search strategies. A visual interface based on the navigation experience is used to help users build mind maps. This interface stimulates curiosity of novice users in order to enhance the information retrieval experience.

III. AFFINITIES BETWEEN DEEP WEB AND GALAXY

G-Form is a Deep Web form which is inspired by the concept of a “galaxy. First, we will present the affinities between a galaxy and a Deep Web form, then we explain the way galaxy-form is build.

TABLE 1. ANALOGY BETWEEN DEEP WEB FORM AND GALAXY

Deep Web Form	Cosmic Universe
Field	star
Group of fields	galaxy
Super-group of fields	Super-galaxy
degree of pertinence of field	Distance separating stars
Mean average pertinence of group of fields	Center of mass of galaxy

Hypothesis :

Novice users regard deep web form like they regard cosmic universe.

Stars are fields and a galaxy is one semantic group of fields. When novice users consider the Deep Web form, they move from one semantic entity to another entity just as an astronaut moves from one planet to another or from one galaxy to another in the cosmic universe. If distance is a measure of this cosmic travel, pertinence is the measure of relevance between fields. The center of mass of one galaxy is equivalent to the mean average pertinence of one semantic entity. Sub-sections A and B detail the similarity between a Deep Web form and the galaxy.

A. Web forms

Deep Web forms organize fields respecting a hierarchical schema (see Figure 2). This schema gives the query its meaning. In our previous work [18] we have presented an algorithm, called VIQI (Visual Interpretation of Deep Web Query Interfaces), which is able to extract the hierarchical schema from Z-Form. Figure 2 gives the resulting output of our algorithm when applied to the query interface (left).

The hierarchical schema (see Figure 2) detects the presence of 4 groups: Departure={Leaving From, Going To}, Number of Passengers={Adults, Children, Infants}, Departure Date={Day of Departure, Month of Departure}, and Returning Date={Returning Day, Returning Month}, and 10 fields: Leaving From, Going To, Adults, Children, Infants, Day of Departure, Month of Departure, Returning Day, Returning Month, and Flight Class) and one super group :root.

As we have mentioned before, the hierarchical aspect of the Deep Web form indicates the presence of semantic relations between entities which are “is-a” and “part-of”. For this reason, we measure the pertinence of one galaxy of fields as the center of mass of the galaxy.

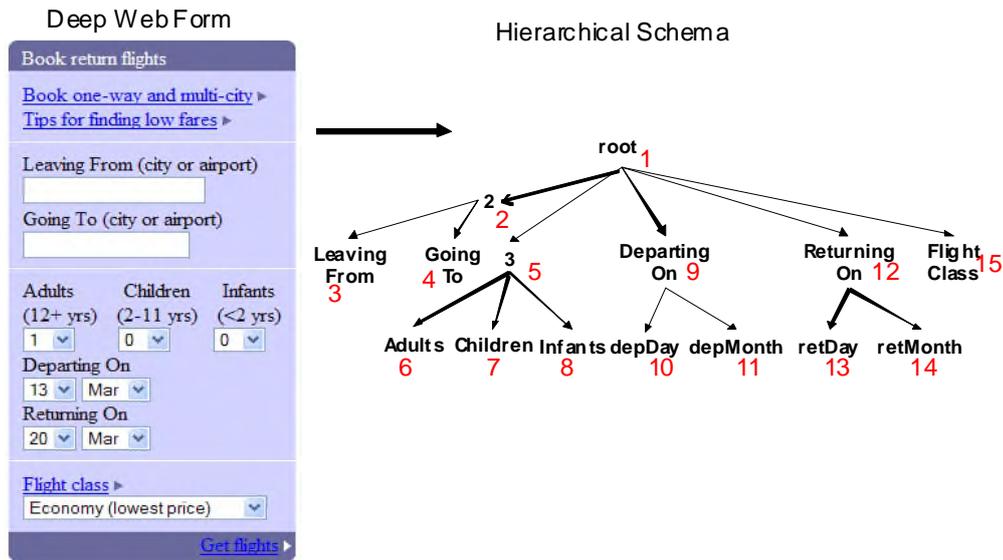


Figure 2. Deep Web form and its hierarchical scheme

Another important aspect in Deep Web forms is degree of pertinence of fields (see red numbers below schema elements of Figure 2). In G-Form, fields are organized according to their pertinence. The most relevant fields are rendered before fields with less pertinence: the most relevant fields are placed on the top left of the Web form, while fields with less pertinence are placed right most on the bottom of the Web form. Hence the field “Adults” is more pertinent than the field “Infants” (pertinence decreases from left to right) and also more pertinent than the field “Children”. And field “Going From” is more pertinent than “Flight Class” (pertinence decreases from top to down) as by default users choose economic class while they must mention where they are going in order to formulate a correct query.

The degree of pertinence in G-Form corresponds exactly to the depth first search (DFS) traversal of schema elements. DFS identifies the order of visiting of the schema elements. In Figure 3, we have indicated under each schema element its degree of pertinence. We remark that $I < J$ if element I is rendered before element J in the Web form (on the left). And $DegreeOfPertinence(I) > DegreeOfPertinence(J)$.

This way, we can measure the relative relevance degree of a group of fields forming one semantic entity. Suppose D is the mean average pertinence degree of a group of fields and r is the distance between the relevance degree of a field and the degree of pertinence of fields group:

if $D/r > \epsilon$, then field is relevant in the group

if $D/r < \epsilon$ then field is not relevant

B. Galaxy

Planets in universe are structured according to hierarchical schema like G-Forms. Planets belong to galaxies and there is

also super-galaxies grouping a set of galaxies. When we regard the sky by night we observe thousands of brilliant stars (see Figure 3). In reality, each brilliant point is not only a star, but may be another galaxy. As the distance between the observer and galaxy is very large light coming from the galaxy appears like a single point.

We present an example. Let us consider Andaman galaxy, situated at distance r from Earth and having dimension D (see Figure 3) as one point wherever in space as Earth is far away from Andaman galaxy. Andaman galaxy is observed as a single point situated in the center of mass of the galaxy, and has as mass the total mass of the entire galaxy.

In mathematics, quotient D/r is:

$$D/r = \frac{\text{Size of square containing Andaman}}{\text{Distance of Center of mass relative to Earth}} \tag{1}$$

If quotient D/r is very small, we can replace the sum of all stars of galaxy Andaman by only one term situated in the center of mass.

if $D/r > \epsilon$ then star is observed from the galaxy

if $D/r < \epsilon$ then star is observed as a single point

Figure 4 shows planet Earth (on the left) and Andaman galaxy (on the right). The square on the bottom shows a zoom on Andaman galaxy. First, it is clear that, according to an observer inside Andaman galaxy, our galaxy Milky Way may be approximated by a mass point situated at the center of mass. In the galaxy Andaman (or Milky Way) itself, this geometric picture repeats, as indicated in Figure 4. While the quotient $D1/r1$ is very small, stars situated at the smallest box can be replaced by their center of mass.

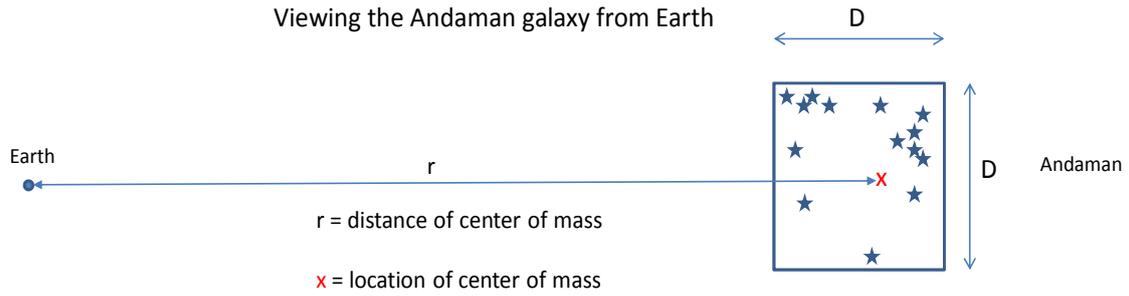


Figure 3. Regarding Andaman galaxy from Earth

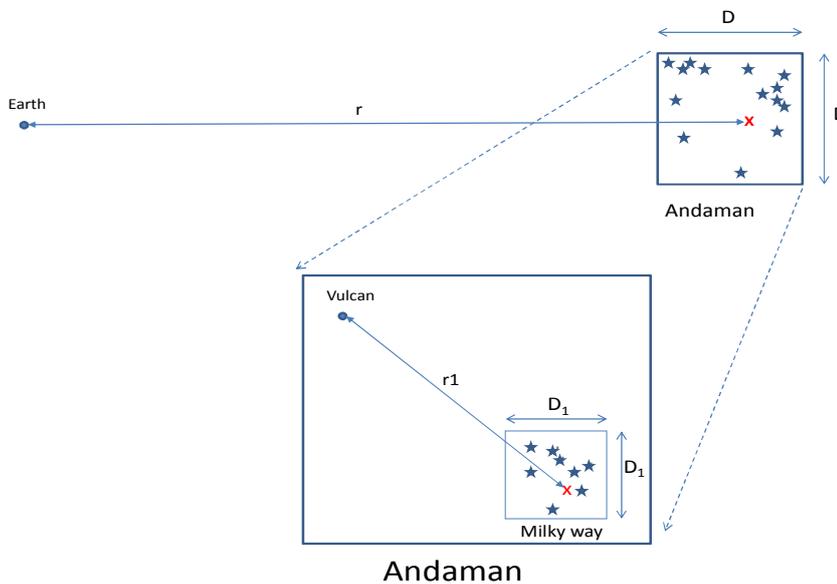


Figure 4. Andaman galaxy analogy

IV. PROPOSED APPROACH: G-FORM

We propose a new approach for visual interpretation of Deep Web forms. The interpretation of the form is based on the hierarchical schema (see Section 3.A) inspired from galaxy. We choose to use a hierarchical representation of the query instead of a flat representation as this representation is richer according to the semantics of the query. The principle of our approach is explained in the following algorithm.

The first Web form is rendered as Z-Form of stars where all fields are stars. When the user clicks on a star, we consider that the user moves to the galaxy containing the field. Our algorithm determines the immediate pertinent fields in the galaxy. A field is considered as pertinent if it is not far from the

center of mass (clicked field) of the galaxy: its degree of pertinence is under ϵ distance, fixed by the user, from the clicked field.

In Table 2, we show which pertinent fields are rendered when the user clicks on a star in Figure 6.(a). For example, let us take ϵ fixed to 1:

- If the user clicks the field with degree of pertinence equal to 3, then only fields “Leaving From” (with pertinence degree 3) and “Going To” (with pertinence degree 4) are rendered, as they are situated under distance inferior or

```

1) Procedure Render_G-Form( {f1, f2, ..., fn}, pertinence
   set, ε)
2) begin
3)   for I from 1 to n
4)     fi ← Not pertinent
5)   end For
6)   if clickOn( fj) then /* fj is the field clicked */
7)     fj ← pertinent
8)   else
9)     for I from 1 to n
10)      D ← distanceOfPertinence( fi, fj)
11)      if (D < ε) then
12)        fi ← pertinent
13)      else
14)        fi ← Not pertinent
15)      end If
16)    end For
17)  end IF
18) end.
    
```

Algorithm 1. Rendering algorithm of G-Form

TABLE 2. OBSERVED FIELDS RELATIVE TO THE CLICKED FIELD

Pertinent fields user clicks	3	4	6	7	8	10	11	13	14	15
3	*	*								
4	*	*								
6			*	*						
7			*	*	*					
8				*	*					
10						*	*			
11						*	*			
13								*	*	
14								*	*	*
15									*	*

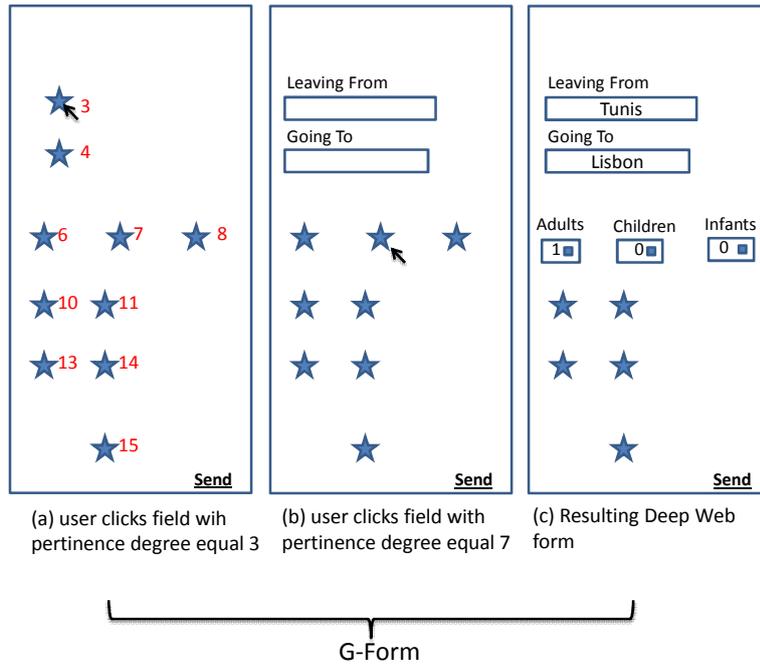


Figure 5. Rendering strategy of G-Form

equal to 1 from the field “Leaving From” (clicked field) having degree of pertinence equal to 3 (see Figure 5.b).

- If the user clicks a field with pertinence 7, then only fields “Adults”, “Children”, and “Infants” are rendered as they are situated under distance 1 from field “Children” (clicked field) having degree of pertinence equal to 7 (see Figure 5.c).

In Table 2, the rows correspond to the clicked star and columns correspond to the rendered fields. According to Table 2, we remark that the rendered fields depend on the clicked field because, when the user clicks a star, we consider that the observer moves to the galaxy of this star. Hence, only stars at quotient $D/r > \epsilon$ (see Section 4.B) are observed from this galaxy and all the others are observed as brilliant stars.

V. EXPERIMENTAL RESULTS

Our approach renders the query according to its semantic representation (schema of the query). We have tested the performance of G-Form on a standard dataset ICQ [19]. ICQ is a collection of query interfaces collected from the Deep Web services. For each query interface, its manually extracted query schema is available on dataset. Interfaces are collected into five classes of subjects: Airfare, Automobile, Books, Real estate, and Jobs.

Our evaluation methodology is as follows. We build G-Form for every schema of Web form available on the dataset. Then, we build Table 2 which simulates the user clicks on different stars in the Web form. Then, we count the number of correct entities (group, super-group of fields). An entity is considered as correct if it is semantically coherent. For

example entity {Adults, Children, Infants} is a correct entity as it describes the number of passengers. While entity {Time, Adults, Children} is not correct because Date of flight and Number of passengers overlap.

We count for each G-Form number of extracted entities, number of extracted entities which are correct, and then we measure precision, recall, and F1 of the algorithm.

$$Recall = \frac{\text{number of extracted entites}}{\text{total number of entites in the Web form}} \quad (2)$$

$$Precision = \frac{\text{number of extracted pertinent entities}}{\text{number of extracted entities}} \quad (3)$$

$$F1 = \frac{2Recall * Precision}{Recall + Precision} \quad (4)$$

The experimental results are summarized in Table 3 with ϵ equals 1:

TABLE 3. EXPERIMENTAL RESULTS

Domain	Airfare	Auto	Books
#extracted	214	102	108
#extracted & pertinent	146	78	70
#total_ entities	200	105	110
Precision	0,68	0,76	0,64
Recall	0,73	0,74	0,63

Figures 6, 7, and 8 show that our approach performs with better results on the domain of interest "Auto". The "Auto" domain contains flat queries, i.e fields are organized at the same level. Choosing a good ϵ coefficient makes the visual representation of the query very easy.

Our approach allows to achieve 73% of recall for "Airfare" domain. Queries in this domain are hierarchical with many levels. Choosing a small ϵ coefficient renders the concepts in the "Airfare" domain as small galaxy composed of 2 or 3 fields.

TABLE 4. COMPARISON BETWEEN OUR APPROACH (G-FORM) AND Z-FORM

	Domain	Precision	Recall	F1
Our approach	Airfare	0,68	0,73	0,70
	Auto	0,76	0,74	0,75
	Book	0,64	0,63	0,64
Z-Form	Airfare	0,66	0,70	0,67
	Auto	0,72	0,71	0,71
	Book	0,62	0,60	0,60

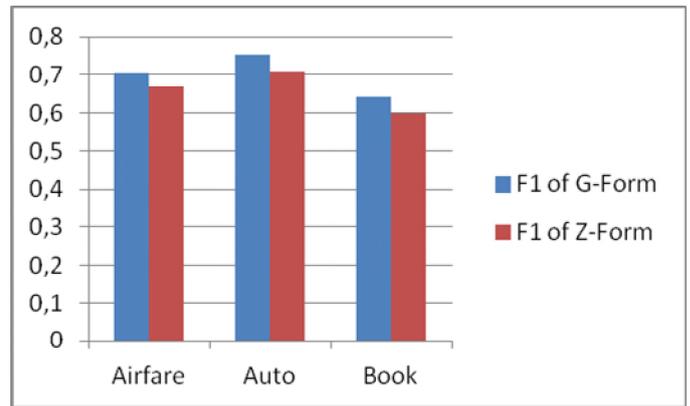


Figure 8. Comparison of F1 of our approach and Z-Form

Figures 6, 7 and 8 summarize the results shown in Table 4. We notice that, with regard to all measures, the two curves corresponding to the two approaches follow the same pace. This phenomenon can be explained by the fact that the process of rendering of fields is based on the query schema which is common to the two approaches. However, Figures 6, 7 and 8 show that the performance of our approach is always superior to the performance of Z-Form. Our approach attends its maximum precision for "Airfare" domain as queries in this domain are hierarchical and well adapted to rendering the strategy of our algorithm.

TABLE 5. PRECISION FOR DIFFERENT QUERY COMPLEXITIES

	Precision		
	$\epsilon=1$	$\epsilon=2$	$\epsilon=3$
Airfare	0,738	0,649	0,635
Auto	0,732	0,693	0,792
Book	0,637	0,601	0,712

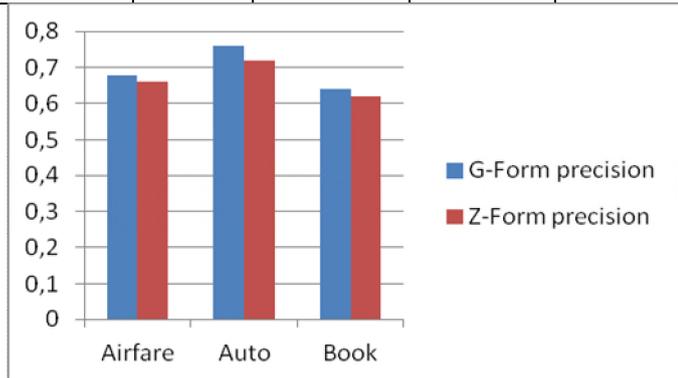


Figure 6. Comparison of precision of our approach and Z-Form

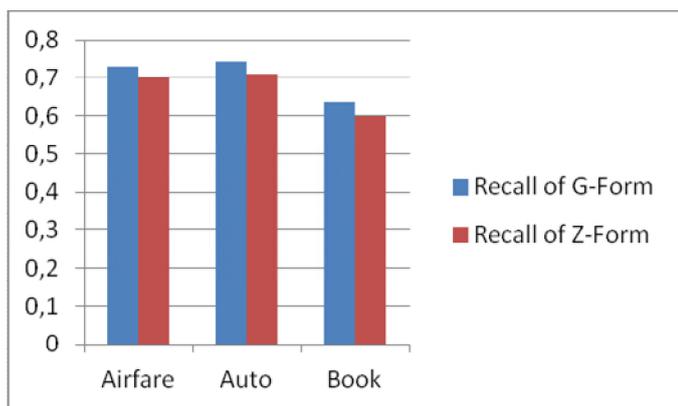


Figure 7. Comparison of recall of our approach and Z-Form

The coefficient ϵ is an indicator of the complexity of the query. For small values of ϵ , only a small group of fields are rendered; the other groups are rendered as stars. This is the case of the "Airfare" domain, which is considered as the most complicated domain. For a large coefficient ϵ , a large group of fields are rendered and interpretation is close to Z-Form. This is the case of the "Book" domain, which is formed with flat queries. The complexity measure of the query can be shown for epsilon 3: "Airfare" is the most complex, "Book" is relatively more complex than airfare, and "Auto" is the simplest query.

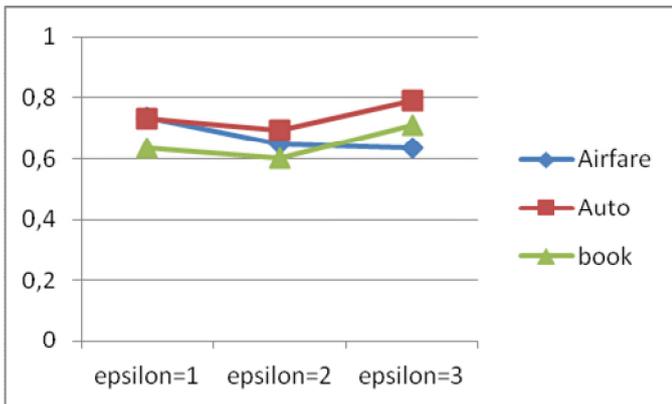


Figure 9. Precision for different query complexities

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a method to solve the complexity of queries in Deep Web forms. We proposed a new design approach G-Form inspired from the concept of a galaxy. It offer to novice users easy-to-use Deep Web forms and reveals clearly the exact meaning of the query even if its schema is complex. There is a strong analogy between the concept of a galaxy and Deep Web form with respect to structure and granularity of entities in each concept.

G-Form is better than Z-Form, which is a well known state of the art algorithm. Z-Form lacks design expressivity because there is no background concerning the semantic value of the query, while our approach is based on a hierarchical schema, which reveals clearly the semantic of the query. Our approach clearly has better performance than Z-Form with respect to precision, recall, and F1 measures of performance.

REFERENCES

- [1] G. Agarwal, G. Kabra, K. C-C Chang, "Towards rich query interpretation: walking back and forth for mining query templates", In proceedings of the international conference on world wide web, 2010.
- [2] F. Jiang, L. Jia, W. Meng, X. Meng, "MrCoM: A Cost Model for Range Query Translation in Deep Web Data Integration", In Proceedings of SKG '08, 2008.
- [3] Z. Zhang, B. He, and K. C-C Chang, "Light-weight domain-based form assistant: querying web databases on the fly", In Proceedings of VLDB '05, 2005.
- [4] Z. Zhang, B. He, and K. C.-C. Chang, "On-the-fly Constraint Mapping across Web Query Interfaces". In Proceedings of VLDB-IIWeb'04, 2004.
- [5] J. Jansen and Dick C.A. Bulterman, "Enabling adaptive time-based web applications with SMIL state", In Proceedings of DocEng '08, 2008.
- [6] M. Jayapandian, H. V. Jagadish. 2008, "Expressive query specification through form customization", In Proceedings of EDBT '08, 2008.
- [7] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, A.Y. Halevy, "Google's Deep Web crawl", In Proceedings of VLDB, 2008.
- [8] E-C. Dragut, T. Kabisch, C. Yu, U. Leser, "A hierarchical approach to model web query interfaces for web source integration", In Proceeding of VLDB 2009, 2009.
- [9] W. Wensheng, A-H Doan, C. Yu, W. Meng, "Modeling and Extracting Deep-Web Query Interfaces", In Proceedings of AIIS 2009, 2009.
- [10] Z. Zhang, B. He, and K. Chang, "Understanding Web query interfaces: Best-effort parsing with hidden syntax", In Proceedings of SIGMOD'04, 2004.

- [11] Heidinger, C., K. Bohm, Buchmann E., and Spoo M. "Efficient and secure exact-match queries in outsourced databases." World Wide Web: 1-39, 2013.
- [12] L.T.H.Vo, J. Cao, and W. Rahayu, "Structured content-based query answers for improving information quality", In World Wide Web: 1-24, 2014.
- [13] J. Losada, J. Raposo, A. Pan, P. Montoto. "Efficient execution of web navigation sequences." World Wide Web: 1-27, 2013.
- [14] R. Boughammoura, MN. Omri, and H. Youssef, "Fuzzy Approach for Pertinent Information Extraction from Web Resources", Journal of Computing and e-Systems, Vol. 1 No. 1, 2008.
- [15] R. Boughammoura, MN. Omri, "Statistical Approach for Information Extraction from Web Pages", In proceedings of International Symposium on Distance Education (EAD'2009), 2009.
- [16] R. Boughammoura, MN. Omri, "SeMQI: A New Model for Semantic Interpretation of Query Interfaces", In Proceedings of NGNS'11, 2011.
- [17] R. Boughammoura, MN. Omri, Hlaoua, L. "VIQI: A New Approach for Visual Interpretation of Deep Web Query Interfaces", ICITeS 2012.
- [18] R. Boughammoura, MN. Omri, Hlaoua, L. "Information Retrieval from Deep Web based on Visuel Query Interpretation", International Journal of Information Retrieval Research, 2(4), 45-59, 2013.
- [19] The UIUC Web Integration Repository, Computer Science Department, University of Illinois at Urbana-Champaign, <http://metaquerier.cs.uiuc.edu/repository>, 2003.
- [20] Ko-Chiu Wu, Affective surfing in the visualized interface of a digital library for children, Information Processing & Management, Volume 51, Issue 4, July 2015, Pages 373-390

Experimental Analysis of Black Virus Decontamination by DisJ

Jie Cai

School of Computer Science
Carleton University
Ottawa, ON, Canada K1S 5B6
Email: jie.cai@carleton.ca

Abstract—In this paper, we experimentally investigate the problem of black virus decontamination. The black virus decontamination is a recently investigated network security problem occurring in networked systems supporting mobile agents. The existing research work on the topic has been focusing on theoretical investigations and analyses. Among the existing simulators for reactive distributed algorithms in network applications, we use Distributed Algorithm Simulation Java, which combines many advantages and overcomes many shortcomings of existing simulators. We consider the basic solution protocol for decontaminating an arbitrary network. We investigate its behaviour, properties and performance through an extensive number of computer simulation runs. The simulation results not only confirm the existing theoretical results, but also disclose many interesting behaviour/properties of the solution protocol. In particular, they show that the examined protocol outperforms random search. The influence of graph connectivity density and size on complexities (move, time, and agent size) is clearly depicted.

Keywords—Black Virus; Mobile Agent; Graph Exploration and Decontamination; Simulation.

I. INTRODUCTION

Recently, many investigations have been performed on various distributed security issues caused by introducing mobile agents into computer networks [1]. For example, a malicious agent can cause computer nodes to malfunction or crash, while a contaminated computer node can in turn destroy mobile agents for various malicious purposes. The former situation is categorized as *harmful agent*, and the latter as *harmful host*.

In the *harmful agent* problem, a dangerous mobile agent moves through the network infecting the visited sites; the task is to decontaminate the network using a team of system agents avoiding recontamination. The problem is referred as *intruder capture* (IC), *graph decontamination*, or *connected graph search*. The mobile intruder is harmful to network sites, but not to the system agents. This problem has been investigated in different settings and topologies by Barrière et al. [2], [3], Blin, Fraignaud, Nisse, and Vial [4], Dereniowski [5], Floccchini et al. [6], [7], [8], Fomin, Thilikos, and Todineau [9], Imani, Sarbazi-Azad, Zomaya, and Moinzadeh [10], Luccio et al. [11], [12], [13], Nisse [14], Shareghi, Sarbazi-Azad, and Imani [15], Yanga, Dyerb, and Alsapach [16], among others.

For harmful host, the theoretical focus has been on the *black hole search* (BHS) problem, in which a network node is infected by a process which destroys any arriving agent

without any detectable trace. The problem has been extensively investigated in different settings and topologies by Chalopin, Das, Labourel, and Markou [17], [18], Cooper, Klasing, and Radzik [19], Czyzowicz et al. [20], [21], Dobrev et al. [24], [22], [25], [26], [23], Glaus [27], Klasing, Markou, Radzik, and Sarracco [28], and Shi [29]. A black hole is *static*, that is, it does not propagate in the network and so it is not harmful to other sites.

A. Black Virus Decontamination

The Black Virus (BV) was introduced in [30], [31] to combine factors missed by BHS and IC to create a new model, in which a harmful process is mobile (like an intruder) and harmful to the system agents (like a black hole). Therefore, the Black Virus Decontamination (BVD) problem is to model a novel security issues caused by mobile agents. The authors studied in detail the BVD problem for three large classes of common network topologies: multi-dimensional grids, tori and hypercubes. Recently, a deterministic exploration protocol for BVD in arbitrary network was developed [32]. It has been proven that *monotonicity* (that is, once a node is explored or decontaminated, it is never recontaminated again) is a necessary condition for a solution protocol to be damage optimal. Theoretical complexity analyses on BV spreads, number of agents, moves, and simulation time are performed for all the solutions. All protocols are optimal both in terms of spread (the number of casualties) and size (the number of agents).

B. Simulation Work on Mobile Agents

1) *Mobile Agents in Network Applications and Their Simulation*: Mobile agents are used in some practical network applications, for example, distributed data mining [33], network management [34], routing [36], consensus problems, network mapping, multiagent coordination for connection, and etc. Amin and Mikler attempted to use mobile agents to design and implement agent based Distance Vector Routing (ADVR) to reduce the overhead and overcome the robustness issues associated with conventional routing protocol [36]. However, except mentioning the fact that Object-Oriented paradigm is adopted, no details on the simulations were provided. Rubinstein and Duarte investigated a mobile agent based network management solution to address scalability and efficiency issues [35]. Network Simulator (NS) is used in the simulation

work. Olfati-Saber and Murray studied consensus problems for networks of dynamic agents with fixed and switching topologies in [37], [38]. Experimental results are provided to demonstrate the effectiveness of the theoretical results. Again the details of the simulation were not presented. Minar, Kramer, and Maes investigated the cooperation of mobile agents for mapping networks to overcome the shortcomings of a centralized solution in [39]. The mobile-agents approach is chosen to obtain routing maps in a distributed and decentralized strategy. Ji and Egerstedt address the connectedness issue in multiagent coordination, i.e., the problem of ensuring that a group of mobile agents stays connected while achieving some performance objective in [44]. In particular, they study the rendezvous and the formation control problems over dynamic interaction graphs.

2) *Simulation Platforms for Distributed Algorithms in Network Applications*: In this subsection, we briefly describe existing simulators for reactive distributed algorithms in network applications with the purpose of: comparing simulation and platforms in network systems; and having reference terms for Distributed Algorithm Simulation in Java (DisJ), a simulation tool we introduce later in this paper.

Distributed Algorithms in Java (DAJ) [45], Toolkit for Distributed Algorithms in Java (T-DAJ) [40], Distributed Algorithms Platform (DAP) [41], Simulation of Network Algorithm (SinAlgo) [42], Distributed Algorithm Simulation Terrain (DisASter) [43] are all platforms for designing, implementing, testing, simulating, and visualizing distributed algorithms. DAP and SinAlgo are mainly suited for wireless network. The above simulators provide various good features although they usually do not have all of them: Object-Oriented Design and implemented by popular languages, i.e., C++ and Java; user friendly GUI; synchronous and/or asynchronous settings; fix or mobile networks; and various level debug capabilities. However, they have some common limitations or disadvantages: only supporting message passing model; only supporting bi-directional link; requiring some level of configuration or coding to create network topology; tightly coupled algorithm implementation and topology creation; lack of statistics calculation and/or display; no support on adversary events; limited interactive information display during simulation.

DisJ, used in this investigation, overcomes almost all the above shortcomings. In addition to its rich functionalities, one of the main advantages is it decouples users' protocol developing activity from defining network topology and executing the protocol. This means intended protocol and network topology are developed, defined, and built separately from each other and from simulation engine. However, one of the main inconveniences is lack of automation to run large number of simulations to generate statistics results.

C. Main Contributions

The problem of exploring and decontaminating a Black Virus in arbitrary graph by multiple mobile agents has been

studied by simulation using DisJ. The model, objective, constraints, and a deterministic solution are reviewed and presented. Large number of simulations on different sizes of graphs with many connectivity densities are carried out.

The algorithm beats random exploration for all graphs at each connectivity level. The simulation disclosed many interesting behaviours of the solution protocol. Simulation demonstrates the worst case complexity analysis in [32], e.g., agents may move in one direction to explore one node, then move to an opposite direction to explore next node at the other end of a graph. In addition, agents may pass some nodes multiple times. In addition to proving the analytical results, simulation provides deep understanding on influence of graph connectivity density and size on complexities (move, time, and agent size). The move, time, and agent size increase with connectivity, but move and time start decreasing after reaching maximum at 40%-60% connectivity levels. The move, and time and agent size seem to increase quadratically and linearly respectively with the graph size.

With regard to statistics of simulation results, it is observed that the standard deviations for moves, agents, and times are all very small compared with average estimations. Particularly the larger graphs and the higher the connectivity levels, the better statistical results are.

The rest of the paper is organized as follows. Section 2 introduces the framework and model, and basic strategy and algorithm. Section 3 introduces the DisJ platform, describes sample graphs, and presents simulation results and analyses. Section 4 presents our conclusion.

II. FRAMEWORK AND ALGORITHM

In this section, we first introduce the framework and model, then present general strategy and algorithm, finally discuss synchronous and asynchronous settings.

A. Framework and Model

The agents operate in a network whose topology is modeled as a simple undirected connected graph $G = (V, E)$. We denote by $E(v) \subseteq E$ the set of edges incident on $v \in V$, by $d(v) = |E(v)|$ its degree, and by Δ the maximum degree of G . Every node has a distinct *id*, visible to the agents visiting it. The links incident to a node are labeled with distinct port numbers.

Agents are modeled as entities with computing power. They can move from a node to one of its neighbour. Communication among agents occurs when they meet at the same node. Each agent has a unique id. In G there is a node infected by a BV whose location is unknown, and any agent arriving at the BV is destroyed. When that occurs, the BV clones itself and spreads from the *current* node to all the neighbouring nodes. Arriving at a node, a clone BV infects the node and stays inactive (until further triggering by agents) if there is no agent on the node; otherwise, the clone BV is destroyed. Thus, the only way to eliminate a BV from the system is to surround it

target contains a BV or not, EA or a BV eventually returns to "current" node, so LEA knows what happens. Synchronization is concerned only in one task, i.e., sending SAs to $N_{ex}(v)$ during shadow exploration. LEA instructs EA to explore the target only after all $N_{ex}(v)$ are protected by SAs. In synchronous setting, LEA calculates all $N_{ex}(v)$ and the shortest paths to them. Let $dist_{max}$ be the maximum of all shortest paths. LEA uses $dist_{max}$ to make sure that all SAs arrive at their destination nodes for protection before it instructs EA to explore the target. In asynchronous setting, LEA has to visit all shadow destinations one by one to be sure that all of them are properly shadowed by SAs before it instructs EA to explore the target. Both synchronous and asynchronous protocols have been implemented, and simulations are executed for both. Asynchronous complexities (agent move and simulation time) are larger than synchronous complexities, but within the same order, so in the following, the simulation discussions and results are focus on the synchronous protocol.

III. SIMULATION OF BVD

In this section, we first introduce the DisJ platform, then describe sample graphs, and finally present simulation results and analyses.

A. Simulation Platform, DisJ

The simulation software used in this work is called DisJ, implemented in Eclipse environment as a plug-in [46]. The simulation engine per se is an event based simulation engine, which is driven by events put on an event heap. One of the main characteristics is that DisJ decouples the users' protocol developing activity from defining user network topology. DisJ can be used to assist teaching or develop distributed protocols by researchers. It provides basic utilities (nodes, links, events, timers, and etc.) and other extensive features, e.g., random delays, different faults with probability, communication (unicast, multi-cast, and broadcast), and etc.

The original simulation engine supports only the message-passing model. To support mobile agents, more functions have been added: injecting agents on nodes; allowing an agent to move from one node to a neighbouring node; supporting different communication mechanisms among agents on the same node via whiteboard, token, or message; and finally supporting agents to create new agents.

DisJ provides rich features for developing protocols, designing network topologies, and debugging. It provides nice GUI interface to allow the users to define their network topologies by automatically generating, drawing, or inputting network topology from network matrix files. In addition to basic debugging features, it also provides advanced features, watching variables and states, restarting, logging execution and replaying, adjusting speed, and etc. It calculates basic statistics pertaining to distributed algorithms, e.g., the number of messages, agent moves, simulation time, and etc. The APIs for programming a protocol is small and simple. DisJ also has been designed with extensibility in mind. Some of

the new features which can be imaged now are: network capability where 2 or more engines can link up in a network, dynamic addition of nodes, and more. DisJ has very good documentation, which includes a detailed user manual and a cookbook. They show step-by-step instructions for installing DisJ plug-in into Eclipse, defining a topology, writing protocol, and executing a protocol in defined topology. Figure 2 shows a sample DisJ Interface.

B. Simulation Sample Graphs

In order to obtain reliable data, we prepared 2255 graphs with different sizes and network connectivity densities. The sizes of the sample graphs are 20 (*graph20*), 40 (*graph40*), 60 (*graph60*), and 80 (*graph80*), and 100 (*graph100*) respectively. We define the network connectivity density/level as a ratio of the number of links of a graph to the one of a complete graph with the same size, i.e., $\frac{2m}{n(n-1)}$. For each size of graphs, we consider 10 connectivity levels, i.e., 10%, 20% to 100%. We implemented a computer program to randomly generate 50 graphs for each connectivity level of each size. Each graph is represented as *graph#1_#2_#3* (#1: graph size; #2: connectivity; #3: graph instance). Two graphs are generated manually, i.e., *graph20_18* and *graph40_11*, with special arrangement of degrees for certain nodes for easy visualization of the behaviour of different solution protocols. Following Figure 3 shows the sample *graph20_18*.

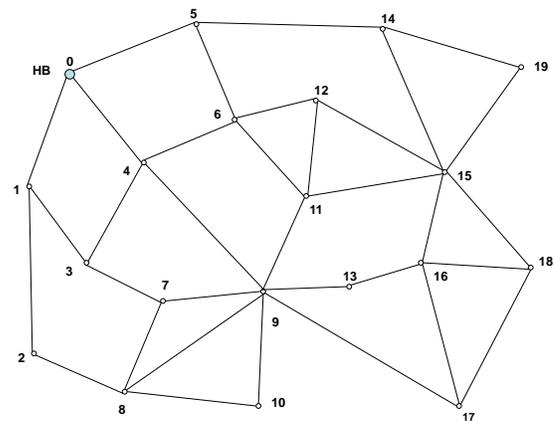


Figure 3. A sample *graph20_18*.

C. Simulation Results

1) *Comparison between MRDE (Minimum Residual Degree Exploration) and Random Exploration (RANE)* : Given two different exploration protocols, P_1 and P_2 , how do we determine which one is better? Recall that the objective of the BVD is to remove all BVs with minimum contamination to the graph. Because we do not know the location of the BV a priori, the d_r of a node when the node is being explored represents the extent of possible contamination.

When a node i is explored, its residual degree, d_{r_i} , is recorded. After exploring all nodes of the graph, they are

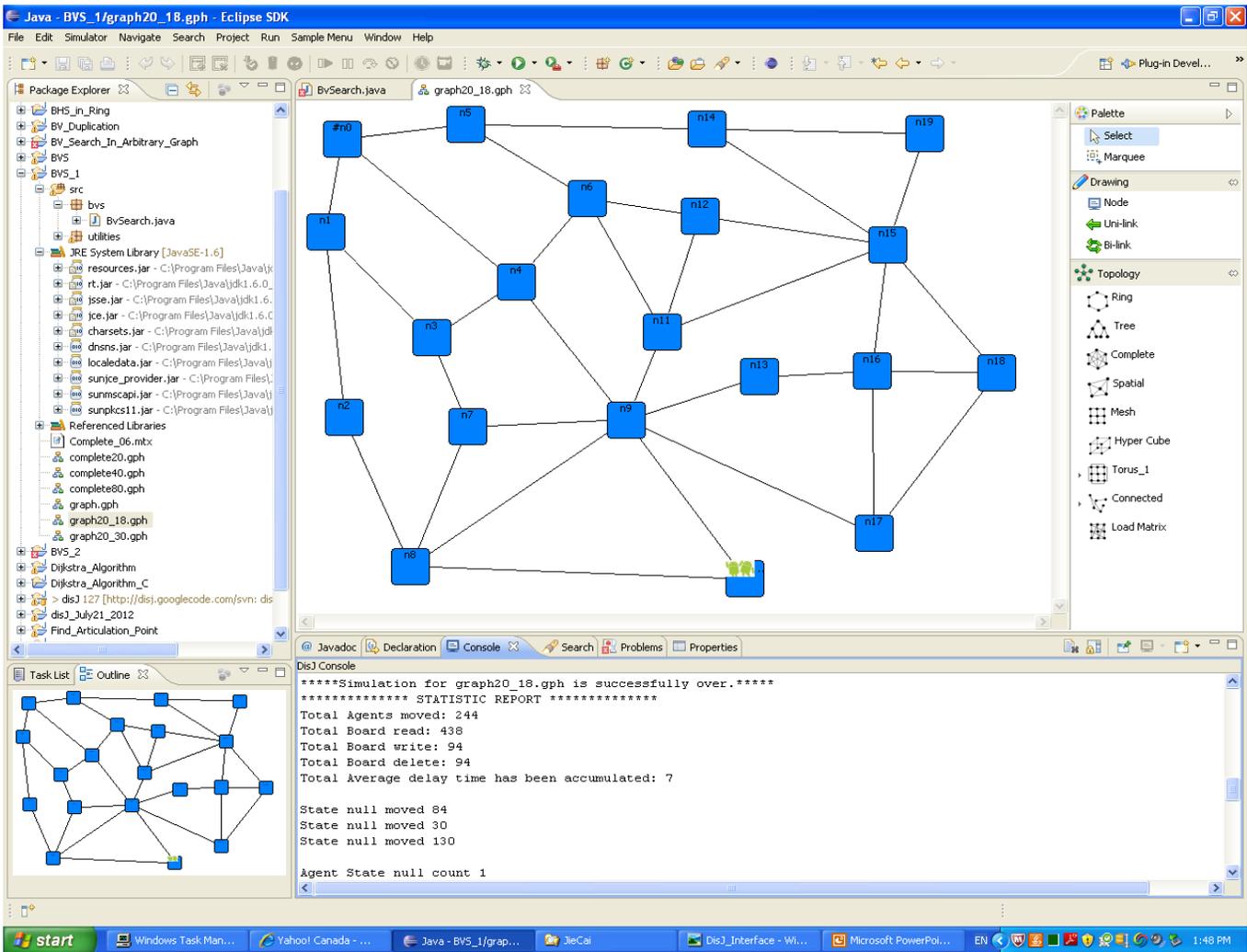


Figure 2. DisJ Interface.

sorted into an array according to their residual degrees in descending order. If the same d_{r_i} in the array appears multiple times, we use a coefficient, i.e., c_i , to record this. The resultant array is called *Array of Residual Degree (ARD)*. Comparing two exploration strategies becomes equivalent to comparing two *ARDs*. The arrays are compared lexicographically. If P_1 's largest $d_{r_1}(P_1)$ is smaller than P_2 's largest $d_{r_1}(P_2)$, P_1 is better than P_2 . If $d_{r_1}(P_1) = d_{r_1}(P_2)$, we compare their coefficients. Whichever protocol has a smaller c_1 is better. If two protocol's c_1 's are the same, we continue to compare the second largest d_{r_2} 's of P_1 and P_2 , and this comparison continues until a better protocol is determined.

In RANE, the next target node is randomly chosen among the unexplored direct neighbours of G_{ex} . We compared MRDE and RANE for the graphs with sizes of 20, 40, and 80 nodes. For each size category, we created 10 connectivity levels, so there are 30 comparisons in total. Simulation results demonstrate MRDE is never worse than RANE for all test scenarios. As shown in Figure 4, there are two curves, one for d_r 's of each SS. The figures clearly show MRDE is better

than RANE. In RANE, d_r changes dramatically, while in MRDE, d_r changes smoothly. It also demonstrated that, when the connectivity increases, the difference between MRDE and RANE decreases. When connectivity approaches 100%, i.e., the complete graph, MRDE and RANE are the same.

2) *Exploration Behavior and Properties:* Running the simulation on sample graph20_18 shown in Figure 3, we observed the following behaviours of the algorithm:

- G_{ex} is continuous. BV and unexplored nodes never cut G_{ex} into isolated pieces. This behaviour matches the monotonous property of the algorithm.
- Changing HB has effect on local SS, but has no impacts on SS in far areas. The behaviour is obvious when watching the searching process starting from nodes 0 to 8. This is a good property because it provides a little flexibility for users to choose where to start to explore the graph.
- SSs are influenced by the graph structures. It is observed that the initial SS starting from nodes 13, 16, 17, or 18 are

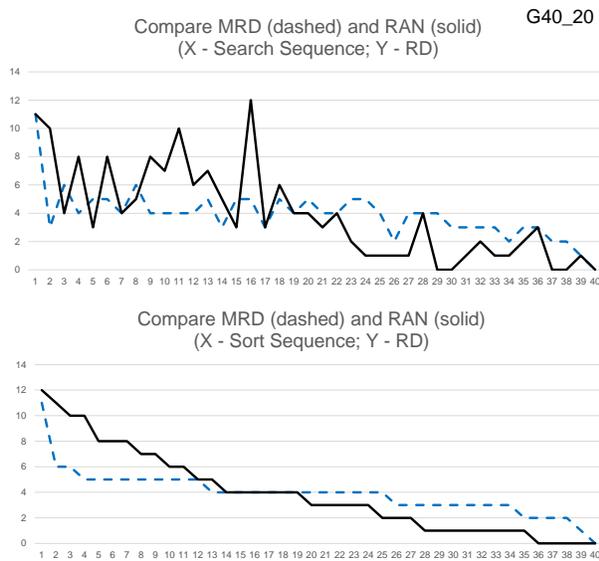


Figure 4. MRD_vs_RAN_G40_20.

dramatically different from those starting from node 0 to 8. This is because initially the exploration is confined by nodes 9 and 15, which have high degrees and are the exits for agents to explore other areas. After breaking through a barrier at node 9 or 15, all SSs follow the same/similar paths in the rest of the graph.

- Starting from the nodes with high degrees could reduce the $Size(G)$ and exploring cost.
- The exploration sequence demonstrates the worst case complexity analyses in [32]. Agents may move to one direction to explore one node, then move opposite to explore another node at far end of a graph. In addition, agents may pass some nodes multiple times. When agents start from node 0, the exploration first moves in one direction along nodes 1 and 2, follows an opposite direction alone nodes 2, 1, 3, 4, and etc., and repeats this changing direction behaviour for several times.

3) *Statistics of Simulation Results:* As mentioned before, at each connectivity level of a given graph size, 50 graphs were generated, simulation is run for each graph, and complexities (agents, moves, and time) are recorded. Then we calculate averages and standard deviations of the complexities for each connectivity level of a given graph size. The ratio of standard deviation over average for complexities are plotted. Figure 5 shows the result for move complexity.

It is observed that, in all cases simulated, the standard deviations for moves, agents, and times are all very small compared with average estimations. With regard to the graph sizes, graph20 has the largest ratio of standard deviation over average for complexities. Large graphs produce small ratio, i.e., better statistical results. This is obviously true because small size graphs do not generate as good statistical results as compared with large size graphs. With regard to the connectivity levels, 10% connectivity generates the largest

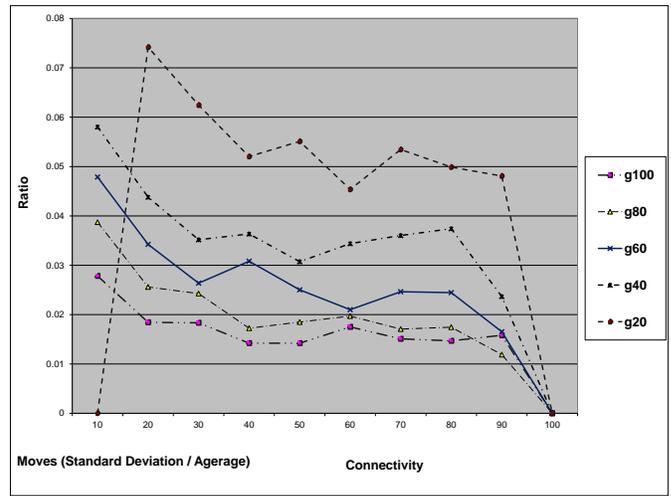


Figure 5. Moves SD Ratio vs Connectivity.

ratio of standard deviation over average for complexities. This is true because at connectivity level 10%, the number of links of the graphs is very small (close to the connectivity of ring or tree of the same size). The higher the connectivity levels, the smaller the ratio of standard deviation over average. At 100% connectivity, the ratio reaches zero. In this case, all the nodes are the same, so no matter where agents start and which path to take to explore the graph, the simulation results are the same.

4) *Influence of connectivity density on complexity:* Refer to the simulation results in Figures 6, 7, and 8, for move, agents, and time in graph20's, graph40's, graph60's, graph80's, and graph100's.

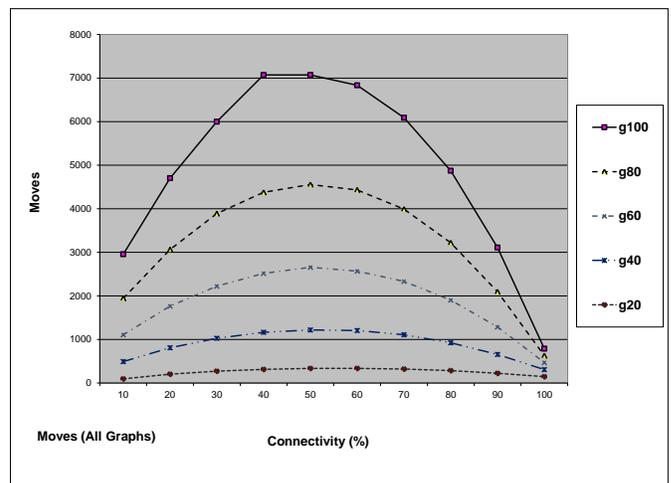


Figure 6. Moves vs Connectivity.

For given size of a graph, with the increase of connectivity level, we have the following results and observations:

- Move cost gradually increases to maximum at 40%-60% connectivity levels, then it gradually decreases. It is interesting

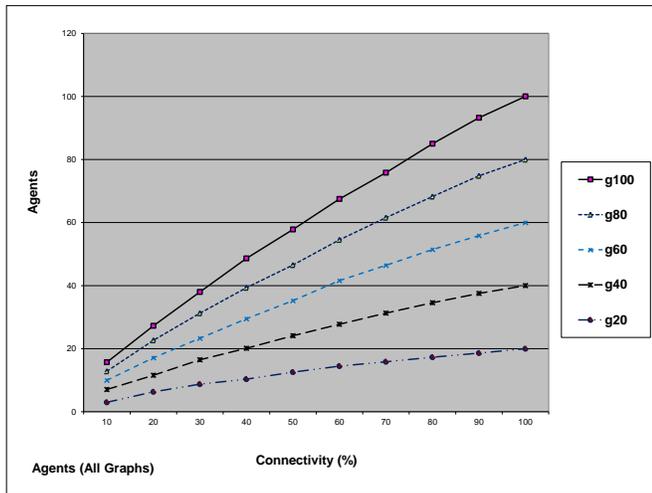


Figure 7. Agents vs Connectivity.

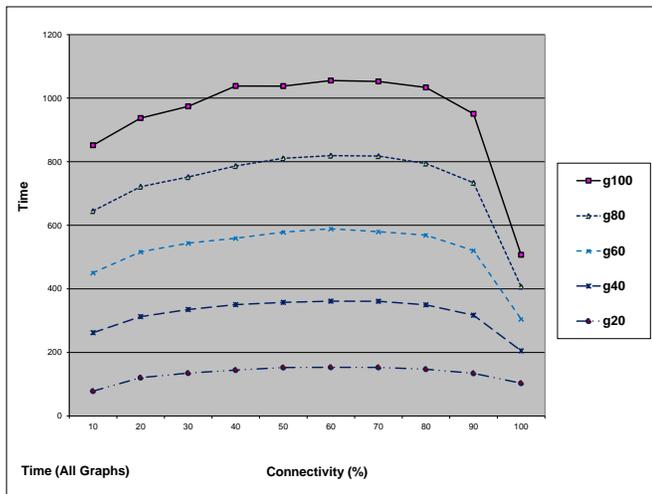


Figure 8. Time vs Connectivity.

to note that move costs are close for different graph sizes at 100% connectivity level. This demonstrates that dense graph optimization does improve move cost.

b. Time cost gradually increase to maximum at middle connectivity levels, then gradually decreases. It is noted that time costs are close for different graph sizes at 100% connectivity level. This demonstrates that dense graph optimization does improve time cost.

c. Team size continuously increases with the increase of connectivity. It reaches the maximum, i.e., graph size, at 100% connectivity level. It is obvious that high connectivity means high degrees, thus more shadow agents are needed to shadow exploration.

d. Regardless of the graph size, the behaviour of MRDE and RANE differentiates in low connectivity levels, but gradually becomes close at high connectivity. At 100% connectivity, the behaviours are the same. Complete graphs are completely symmetrical, i.e., choosing any one node as

next target to explore gives exactly the same simulation results.

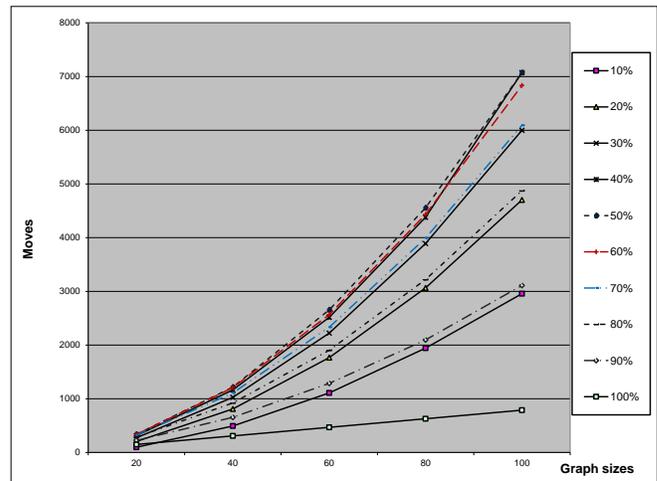


Figure 9. Moves vs Graph Size.

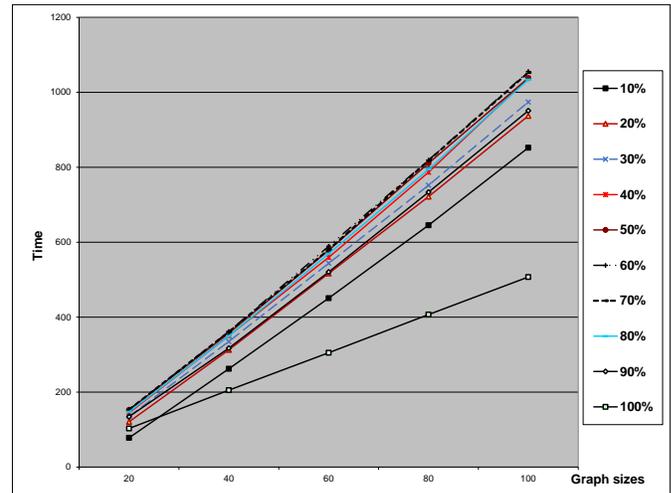


Figure 10. Time vs Graph Size.

5) Influence of graph size on complexity: For given connectivity levels, with the increase of graph sizes, the move, agent, and time are shown in Figures 9, 10, and 11:

a. From Figure 9, it is observed that the move costs seem to increase quadratically with the graph sizes. The move costs for 10% and 90%, 20% and 80%, 30% and 70%, 40%, 50% and 60% are very close respectively. Among all the connectivity levels, the complete graph has the lowest move costs, while 40%, 50% and 60% connectivity levels have the highest move costs. These behaviours demonstrate well that move costs increase with connectivity levels. It saturates at around 50% connectivity, then decreases to minimum when connectivity is 100%.

b. From Figure 10, it is observed that the execution time seems to increase linearly with the graph sizes. The execution times for 20% - 80% are relatively close to each other. Among

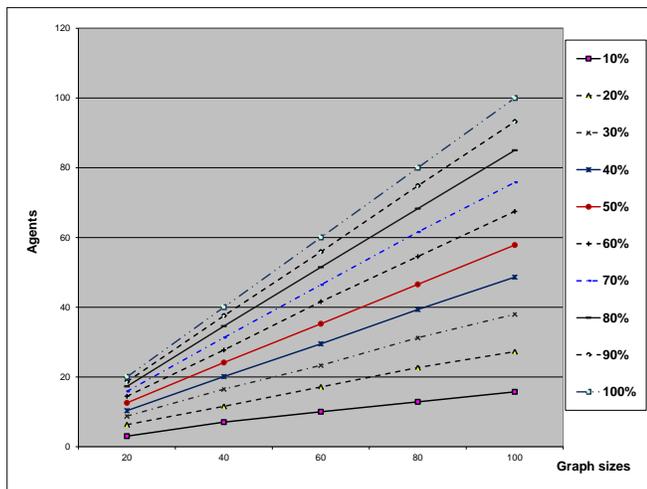


Figure 11. Agents vs Graph Size.

all the connectivity levels, the complete graph has the lowest time costs, while other connectivity levels have relatively high time costs. Generally speaking, the time costs are lower than move costs for same graph size and connectivity level because many activities happen in parallel.

c. Team size all increase when graph size increases. From Figure 11, it is observed that the Team size increases (linearly) with the graph size. As mentioned before, the connectivity also impacts the needed agents linearly.

IV. CONCLUSION

We have investigated the BVD problem in arbitrary graphs by mobile agents, a newly introduced network security problem. All the existing work has been focusing on theoretical investigation and analyses. In this paper, we instead focused on the experimental investigation of BVD. Among the existing simulators for reactive distributed algorithms in network applications, we used DisJ, which combines many advantages and overcomes many shortcomings of existing simulators. In addition to basic features, it offers mobile-agent support, flexible network element settings, simple APIs, user-friendly GUI, strong debugging capabilities. We considered the basic solution protocol for decontaminating an arbitrary network, MRDE; we investigated its behaviour, properties and performance through an extensive number of computer simulation runs; in the investigation we used as a term of comparison the natural "randomized" strategy, RANE.

The simulation results have disclosed many interesting behaviour and properties of the solution protocol. In particular,

a. With regard to the influence of connectivity density on complexity, we demonstrate that:

- Move cost gradually increases to maximum at 40%-60% connectivity levels, then gradually decreases.
- Time cost gradually increase to maximum at middle connectivity levels, then gradually decreases.

- Team size continuously increases with the increase of connectivity. It reaches the maximum, i.e., graph size, at 100% connectivity level.
- MRDE and RANE differentiate in low connectivity levels, but gradually get close at high connectivity. At 100% connectivity, the behaviours are exactly the same.

b. With regard to the influence of graph size on complexity, we demonstrate that:

- The move costs seem to increase quadratically with the graph sizes.
- The execution time seems to increase linearly with the graph sizes.
- Team size all increases when graph size increases.

These results constitute the first experimental investigation of solution protocols for the Black Virus decontamination problem. Indeed, simulation is a valuable tool to gain an insight into the nature of the problem. Future work could be: 1) some improvements on DisJ to automatically process a batch of graphs and collect results; 2) some simulation work on systems with multiple black virus.

ACKNOWLEDGMENT

The author would like to express his sincere appreciation to Professor Paola Flocchini and Nicola Santoro for their enthusiastic guidance and advice throughout this research.

REFERENCES

- [1] P. Flocchini and N. Santoro, "Distributed Security Algorithms by Mobile Agents," *Distributed Computing and Networking, Lecture Notes in Computer Science*, 2011, pp. 1-14.
- [2] L. Barrière, P. Flocchini, F. V. Fomin, P. Fraignaud, N. Nisse, N. Santoro, and D. M. Thilikos, "Connected graph searching," *Information and Computation*, 2012, pp. 1-16.
- [3] L. Barrière, P. Flocchini, P. Fraignaud, and N. Santoro, "Capture of an intruder by mobile agents," *14th Symposium on Parallel Algorithms and Architectures (SPAA)*, 2002, pp. 200-209.
- [4] L. Blin, P. Fraignaud, N. Nisse, and S. Vial, "Distributed chasing of network intruders," *Theoretical Computer Science*, 2006, pp. 70-84.
- [5] D. Dereniowski, "Connected searching of weighted trees," *Theoretical Computer Science*, 2011, pp. 5700-5713.
- [6] P. Flocchini, M. J. Huang, and F. L. Luccio, "Decontamination of hypercubes by mobile agents," *Networks*, 2008, pp. 167-178.
- [7] P. Flocchini, M. J. Huang, and F. L. Luccio, "Decontaminating chordal rings and tori using mobile agents," *International Journal on Foundations of Computer Science*, 2006, pp. 547-563.
- [8] P. Flocchini, F. L. Luccio, and L. X. Song, "Size optimal strategies for capturing an intruder in mesh networks," *International Conference on Communications in Computing (CIC)*, 2005, pp. 200-206.
- [9] F. V. Fomin, D. M. Thilikos, and I. Todineau, "Connected graph searching in outerplanar graphs," *7th International Conference on Graph Theory (ICGT)*, 2005, pp. 213-216.
- [10] N. Imani, H. Sarbazi-Azad, A. Y. Zomaya, and P. Moizadeh, "Detecting threats in star graphs," *IEEE Transactions on Parallel and Distributed Systems*, 2009, pp. 474-483.
- [11] F. Luccio and L. Pagli, "A general approach to toroidal mesh decontamination with local immunity," *23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2009, pp. 1-8.
- [12] F. Luccio, L. Pagli, and N. Santoro, "Network decontamination in presence of local immunity," *International Journal of Foundation of Computer Science*, 2007, pp. 457-474.
- [13] F. L. Luccio, "Continuous search problem in Sierpinski graphs," *Theory of Computing Systems*, 2009, pp. 186-204.

- [14] N. Nisse, "Connected graph searching in chordal graphs," *Discrete Applied Mathematics*, 2009, pp. 2603-2610
- [15] P. Shareghi, H. Sarbazi-Azad, and N. Imani, "Capturing an intruder in the pyramid," *International Computer Science Symposium in Russia (CSR)*, 2006, pp. 580-590.
- [16] B. Yanga, D. Dyerb, and B. Alspach, "Sweeping graphs with large clique number," *Discrete Mathematics*, 2009, pp. 5770-5780.
- [17] J. Chalopin, S. Das, A. Labourel, and E. Markou, "Tight Bounds for Scattered Black Hole Search in a Ring," *Lecture Notes in Computer Science*, 2011, pp. 186-197.
- [18] J. Chalopin, S. Das, A. Labourel, and E. Markou, "Black hole search with finite automata scattered in a synchronous torus," *25th International Symposium on Distributed Computing (DISC)*, 2011, pp. 432-446.
- [19] C. Cooper, R. Klasing, and T. Radzik, "Searching for black-hole faults in a network using multiple agents," *10th International Conference on Principles of Distributed Systems (OPODIS)*, 2006, pp. 320-332.
- [20] J. Czyzowicz, S. Dobrev, R. Královic, S. Miklík, and D. Pardubská, "Black Hole Search in Directed Graphs," *16th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2009, pp. 182-194.
- [21] J. Czyzowicz, D. R. Kowalski, E. Markou, and A. Pelc, "Searching for a black hole in synchronous tree networks," *Combinatorics, Probability & Computing*, 2007, pp. 595-619.
- [22] S. Dobrev, P. Flocchini, R. Královic, P. Ruzicka, G. Prencipe, and N. Santoro, "Black hole search in common interconnection networks," *Networks*, 2006, pp. 61-71.
- [23] S. Dobrev, P. Flocchini, R. Královic, and N. Santoro, "Exploring an unknown dangerous graph using tokens," *Theoretical Computer Science*, 2013, pp. 2845.
- [24] S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro, "Searching for a black hole in arbitrary networks: Optimal mobile agents protocols," *Distributed Computing*, 2006, pp. 1-18.
- [25] S. Dobrev, P. Flocchini, and N. Santoro, "Mobile search for a black hole in an anonymous ring," *Algorithmica*, 2007, pp. 67-90.
- [26] S. Dobrev, N. Santoro, and W. Shi, "Using scattered mobile agents to locate a black hole in a unoriented ring with tokens," *Int. Journal of Foundations of Computer Science*, 2008, pp. 1355-1372.
- [27] P. Glaus, "Locating a black hole without the knowledge of incoming link," *5th Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSOR)*, 2009, pp. 128-138.
- [28] R. Klasing, E. Markou, T. Radzik, and F. Sarracco, "Approximation bounds for black hole search problems," *Networks*, 2008, pp. 216-226.
- [29] W. Shi, "Black hole search with tokens in interconnected networks," *11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, 2009, pp. 670-682.
- [30] J. Cai, P. Flocchini, and N. Santoro, "Network Decontamination from a Black Virus," *27th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Boston (MA), 2013, pp. 696-705.
- [31] J. Cai, P. Flocchini, and N. Santoro, "Decontaminating a Network From a Black Virus," *International Journal of Networking and Computing* 2014, pp. 151-173.
- [32] J. Cai, P. Flocchini, and N. Santoro, "Black Virus Decontamination in Arbitrary Networks," *The 3rd World Conference on Info Systems and Technologies (WorldCIST'15)*, Azores, Portugal, 2015, pp. 991-1000.
- [33] M. Yubao, and D. Renyuan, "Mobile Agent technology and Its Application in Distributed Data Mining," *First International Workshop on Database Technology and Applications*, 2009, pp. 151-155.
- [34] M. Kona, and Z. Xu, "A framework for network management using mobile agents," *27th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Florida, 2002, pp. 15-19.
- [35] M. G. Rubinstein, and O. C. M. B. Duarte, "Evaluating the performance of mobile agents in network management," *Global Telecommunications Conference*, GLOBECOM, 1999, pp. 386-390.
- [36] K. A. Amin, and A. R. Mikler, "Agent-based distance vector routing: a resource efficient and scalable approach to routing in large communication networks," *Journal of Systems and Software*, 2004, pp. 215227.
- [37] R. Olfati-Saber, and R. M. Murray, "Consensus Protocols for Networks of Dynamic Agents," *Proc. Amer. Control Conf.*, 2003, pp. 951956.
- [38] R. Olfati-Saber, and R. M. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, 2004, pp. 1520-1533.
- [39] N. Minar, K. Kramer, and P. Maes, "Cooperating Mobile Agents for Mapping Networks," *Proceedings of the First Hungarian National Conference on Agent Based Computing*, 1999, pp. 287-304.
- [40] W. Schreiner, "A java toolkit for teaching distributed algorithms," *Proceedings of the 7th annual conference on Innovation and technology in computer science education*, 2002, pp. 111-115.
- [41] I. Chatzigiannakis, A. Kinalis, A. Poulakidas, G. Prasinos, and C. Zaroliagis, "DAP: A Generic Platform for the Simulation of Distributed Algorithms," *Proceedings of the 37th Annual Symposium on Simulation, ANSS*, 2004, pp. 166-177.
- [42] Distributed Computing Group, "SinAlgo - Simulator for Network Algorithms," <http://www.disco.ethz.ch/projects/sinalgo/>, August 2015
- [43] R. Oechsle, and T. Gottwald, "DisASter (distributed algorithms simulation terrain): a platform for the implementation of distributed algorithms," *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '05, 2005, pp. 44-48.
- [44] M. Ji, and M. Egerstedt, "Distributed Coordination Control of Multiagent Systems While Preserving Connectedness," *IEEE Transactions on Robotics*, 2007, pp. 693-703.
- [45] M. Ben-Ari, "Interactive Execution of Distributed Algorithms," *ACM Journal of Education Resources in Computing*, 2001, article 2.
- [46] N. Santoro, "Development of a Simulation Engine for Distributed Algorithms System Manual," *Internal Report*, Carleton University, 2001.

Multi-Agent Technology in Real-time Intelligent Resource Management Systems

Igor Mayorov

Samara State Technical University
Smart Solutions, Ltd
Samara, Russia

e-mail: imayorov@smartsolutions-123.ru

Petr Skobelev

Samara State Aerospace University
Samara, Russia
e-mail: petr.skobelev@gmail.com

Abstract—The article describes the main principles of intelligent real-time resource management systems based on the use of multi-agent technology. Features of the new generation of systems are demonstrated that implement the full cycle of autonomous resource management, from reaction to real-world to monitoring deviations between the plan and the fact on the basis of the developed multi-agent platform. The article also presents several applications of scheduling systems in various areas, including cargo flow management for the International Space Station, workshop management in machine-building enterprises, railway traffic and cargo transportation management. Adaptability of multi-agent systems to external disruptive events is demonstrated. Finally, the similarities between multi-agent systems and non-equilibrium thermodynamics of Ilya Prigogine are described.

Keywords—multi-agent technology; resource management; dynamic scheduling; real-time; demand-resource network; adaptability

I. INTRODUCTION

Methods for enterprise resource planning, which would give the possibility to quickly, flexibly and effectively make smart decisions reflecting a balance between interests of many participants in the conditions of growing complexity of the modern world, play a crucial role in a variety of different applications.

In this regard, it is not surprising that development of computational algorithms for allocation, planning and optimization of resources is moving towards development of adaptive resource management systems suitable for rapidly changing conditions of the modern environment, although the software market is still dominated by batch systems in which orders and resources are known in advance and do not change in the course of computation.

In contrast to the classical large, centralized, indivisible and sequential software programs, multi-agent systems (MAS) are set up in the form of distributed groups of small autonomous software objects running asynchronously but concurrently in order to produce the result.

Multi-agent technologies are increasingly gaining the position of one of the most innovative tools for real-time planning for a wide range of tasks. Scientific and practical bases of multi-agent approach to solving complex problems and building distributed systems began to take shape in the last decades of the 20th century at the junction of artificial intelligence, object-oriented and concurrent programming, Internet technologies and telecommunications.

Multi-agent technologies are at the heart of recent developments [1]-[4] making it possible to respond to the challenges which modern systems are facing nowadays. They give the possibility to present the process of solving any complex problem (in this case – resource management) as a process of self-organization and searching for a balance between opposing interests of basic demand agents and resource agents, implemented through mechanisms of negotiation with concessions on the basis of market mechanisms of service delivery.

The article is organized as follows. Section II describes the multi-agent approach which is applied in Smart Solutions products, and summarizes modern state of traditional and multi-agent approaches to resource planning. Section III deals with application of the developed multi-agent methods to scheduling of cargo flow and flight program of International Space Station, planning of a group of spacecrafts for Earth remote sensing, real-time railway traffic management of Russian Railways, and cargo transportation management. Implementation efficiency of multi-agent systems in these domains is proved. Section IV provides a method of adaptability evaluation of real-time multi-agent systems. Section V outlines a thermodynamic approach to description of multi-agent systems. In Conclusion results of successfully implemented methods by Smart Solutions are given as well as further steps in multi-agent system development.

II. MULTI-AGENT APPROACH TO SOLVING THE PROBLEMS OF ADAPTIVE PLANNING BASED ON DEMAND-RESOURCE NETWORKS

Solving traditional resource planning tasks is usually formulated as a batch process, where all orders and resources are known in advance and do not change in the process of work [5][6]. At present, traditional ERP-systems (Enterprise Resource Planning) has included more and more resource schedulers often called ASP methods (Advanced Scheduling and Planning), which are developed by such companies as SAP, Oracle, Manugistic, i2, ILOG, J-Log, etc.

However, such systems implement, as a rule, batch linear and dynamic programming, constraint programming, and other methods based on combinatorial search of variants [6], which appear to be of little use in practice. To reduce complexity of combinatorial search, new heuristic and metaheuristic methods are used [7], which give the opportunity to obtain appropriate results in a reasonably short period of time due to reducing combinatorial search variants.

Besides, one can use “greedy” local search methods, simulated annealing, constraint programming, taboo search, genetic and ant colony algorithms, etc.

The stated methods also use batch processing. They are hardly extended by additional target criteria and do not allow considering various factors which are often used in real life, which can be set not only by formulas and in equations, but also by tables and diagrams.

Moreover, search of variants in real data takes up too much time, though the results are usually quite improbable and hardly comparable to decisions made by people in real life.

From the very beginning multi-agent technology has been used for solving traditional optimization tasks with the use of distributed decision making approaches, for example, Distributed Constraint Optimization task (DCOP) [8]. Besides, several bio-inspired methods have been developed for solving resource scheduling problems, for example swarm optimization, hybrid methods based on artificial immune system and Particle Swarm Optimization (PSO) [9] [10].

One of the new approaches is built on bio-inspired distributed problem solving of resource scheduling problems based on multi-agent technology with economic reasoning [11][12].

Despite the fact that PSO is a powerful stochastic evolutionary algorithm, its disadvantage is that it can lead to a local optimum. In order to increase algorithm productivity, various methods are suggested, for example, improvement of initial swarm characteristics [13][14]. In the multi-agent optimization method with adaptive parameters [15], it is suggested to modify the range of speed changes to avoid speed increase, which will allow for reducing search time of the optimal decision.

Besides, PSO algorithm modification – Two-swarm Cooperative Particle Swarm Optimization (TCPPO) [16] that uses two swarms “driving” and “driven” will give the opportunity of increasing adaptability level of swarm intelligence.

The application of evolutionary algorithms and swarm optimization algorithms, in particular, in multi-agent systems allows to solve problems of high complexity that cannot be solved by other ways, due to the combinatory rising computations complexity [17].

The multi-agent approach is used primarily for solving multi-criteria planning tasks, including quality of products or services, time for their implementation, price (prime cost), risks, etc. In the proposed approach, the system itself chooses goals for improving the vector of its parameters, based on the achieved results and the current situation with orders and resources. As its primary objective for improving its condition the system chooses the criterion having the worst indicator values. Implementation of multi-agent approach in development of an intelligent system for dynamic planning is based on the concept of demand-resource networks and the method of conjugated interactions for real-time enterprise resource management on the virtual market [1]-[3].

According to this concept, each request, order or other demand as well as each resource (production resource, machinery, equipment, vehicles, personnel) are assigned to software agents that negotiate with other agents and plan order fulfillment “just-in-time” (JIT) or “as soon as possible” (ASAP). This ensures support of collective coordination and decision-making in real time at various stages of planning and production plan execution in the various subdivisions working together to solve common problems.

Planning is done in several interrelated stages: if changes are made at one of the stages, it is necessary to make adjustments at all subsequent stages. Some examples of such external influences are changes in the cost of resources and orders, changes in the schedule of materials deliveries, equipment failure and so on. But on the other hand, the resource itself receives proposals from various claims (demands) and decides which of the orders are more suitable for it.

These decisions can not be made once and for all and are not made that way. They can be reviewed and modified as the situation is changing and new events are happening in real time. At the same time, new connections established between agents cause changes in operating conditions for other agents, and thus define the process of system self-organization leading to restructuring of the schedule in response to emerging events. The result is considered to be achieved and the system completes its work only when none of the agents have opportunities to improve their state, the time for finding a solution has run out, or the user has interrupted the process in order to enter interactive mode.

Multi-agent technology makes it possible to create software agents that are trying to optimize their target parameters. For example, an order in a factory or in a cargo company wants to be executed just in time and at minimal cost, or a resource (machine, truck, etc.) wants to be used as efficiently as possible and have no downtime or overwork. Agents at first do it “selfishly” (independently), without asking anyone – that is why they manage to do it very quickly if resources are available. However, if the decisions of other agents create conflicts, they are able to negotiate, make concessions and seek a solution (consensus) in favor of the mutual interest that unites them (for example, a worker or a driver, a workshop or a car fleet).

This approach can be considered an example of Distributed Problem Solving, in which a complex task is decomposed into subtasks that can be solved independently, but then the solutions obtained are combined and detected conflicts are solved.

III. INDUSTRIAL APPLICATIONS OF MULTI-AGENT SYSTEMS IN SMART SOLUTIONS DURING 2010-2015

A. Multi-Agent System for Scheduling of Flight Program, Cargo Flow and Resources of International Space Station

This project was commissioned by S.P. Korolev Rocket and Space Corporation (RSC) “Energia” with the goal of solving the challenges of flight program and cargo flow planning for the International Space Station (ISS) [4]. The

multi-agent system provides interactive support for the following tasks [18]:

Flight program design, that results in distribution of space ships' docking to ISS across modules and time considering various constraints:

- Cargo flow strategic and operational scheduling that results in distribution of deliveries of units, blocks and systems across transportation flights and manned spacecraft;
- Fuel deliveries and both strategic and tactical scheduling based on a forecast of ISS position changes, Sun activity, operations plan and flight program;
- Water, food and other supply delivery scheduling based on information about the expeditions and flight program;
- Scheduling of items returned to the surface and disposal of waste items;
- Flight crew time scheduling.

The main feature of the system is the adaptive scheduler of cargo flow following the real time changes in demand. New cargo orders may displace those already allocated, but with lower priority or delivery deadline.

The system constructs the flight and cargo plan after taking into account the availability of space on the next spacecraft to be launched, available space on board of the station, disposal of existing cargo, which provides more available and other factors, while allowing the user to adjust the results manually.

The results of the implementation are as follows:

- The system made it possible for the first time to keep track of redundant or missing equipment aboard the station.
- The time required for scheduling cargo flow for a duration of one year has decreased from 176 to 8 hours, and the time for its approval - from 264 to 88 hours.
- Keeping the current cargo plan up to date saves up to 200 hours per year.
- Preparation of reserve flight plans for emergencies takes 320 hours less per year.
- The time for allocation of one spacecraft's cargo aboard the station went from 264 down to 128 hours, for a total of 544 hours per year for "Progress" spacecraft and 320 hours per year for "Soyuz" manned capsule.
- Automatic verification of lists of cargo for disposal eliminates duplicates and saves about 312 hours per year.

Fuel balance planning, water and other supply balance planning and crew's working schedule allocation all take 10%-15% less time, for a total of 270 hours per year.

B. Smart Satellites: System for Management of a Group of Spacecrafts for Earth Remote Sensing

A multi-agent system for management of target usage of spacecraft groups for Earth remote sensing has been developed. Smart Satellites makes it possible to adaptively

redistribute sensing tasks among the spacecrafts within groups [19][20].

The space system is represented as a heterogeneous multi-agent system in which agents are represented by spacecrafts for Earth remote sensing, satellites-retranslators which serve to ensure operational communication between agents and, finally, ground stations. Management is implemented through coordinated interaction of spacecrafts: the satellites dynamically form a team, distribute among themselves the task and solve it in parts, depending on their location and of on-board equipment capabilities. To formalize the description of the desired object an ontology is used which is presented in the form of a semantic network.

The experimental results showed that the use of multi-agent management can reduce the total time of sensing in complex dynamic environments, and ensure viability of the system in case a number of spacecrafts leave the group.

C. Smart Railways: Railway Traffic Management System

Smart Railways, a distributed intelligent system for real-time railway traffic management, commissioned by the Russian Railways, is aimed at building and adapting multilinked and multilevel schedules for operation of Russian Railways subdivisions in case of unforeseen events, including the schedules for high-speed, passenger and cargo trains, locomotives, stations, crews of locomotive drivers [21]. Intelligent system for management of passenger railway traffic is developed to control the traffic of high-speed "Sapsan" trains. The system builds the initial master-plan and then makes adjustments under the influence of occurring events, such as, for example, maintenance work on track sections. The logics of reaction to events is implemented as follows: each event starts a chain for rescheduling of resources in the system. Moreover, proactive optimization of plans is carried out in order to search for better options while there is still time for the system operation.

The developed system makes it possible to achieve the following important indicators:

- almost no delays of high-speed "Sapsan" trains;
- building the whole schedule takes up to 45 minutes;
- reaction to events - up to 30 seconds.
- All the main safety requirements are met (intervals between trains, no crossing of bulk cargo trains and high-speed trains, etc.) - 99%;
- technical requirements (acceleration and breaking time, choice of platforms, standing times) - 97%.
- The average delay became less than 8% (up to 30 trains in conflict);
- the average time for return to the master-plan became 1.5 times less.
- Productivity of dispatchers' work increased by more than 2 times.

D. Smart Factory: Multi-agent System for Workshop Management in Machine-building Enterprises

The multi-agent system "Smart Factory" is designed to increase productivity and efficiency of factories by means of adaptive resource allocation, planning, optimization and

monitoring of machine assembly workshops in real time [22]. The system gives managers the opportunity to input information on new events and start rescheduling, as well as to connect or divide operations and adjust the plan, initializing a chain of automatic changes to the plan.

These solutions increase efficiency of the factory in the following way:

- machine load is increased by 20%,
- output per 1 worker is increased by 30%,
- observation of contracts and terms of product output as well as transparency of production processes are increased by 90%,
- operational cost control is increased by up to 100% in real-time mode,
- actual production cycle is reduced by up to 30%,
- stocks of finished goods and stock reserve are reduced by up to 15%,
- complexity of scheduling and forecasting results of the factory operation is also reduced.

E. Smart Trucks: Multi-agent System for Cargo Transportation Management

Smart Trucks, a multi-agent system for cargo transportation management, was designed for a customer with its head office in Moscow and more than a dozen branches across the country. The customer organizes cargo transportation using its own fleet of over 100 trucks and more than a hundred outsource carriers involved [23]. The system implements the full cycle of resource management in real time and provides the capability to automatically control the business process of application receipt, loading and unloading of cargo through communication with the driver via a cell phone. The driver must input the start and end signals for corresponding operations (loading, transportation, unloading).

As a result of implementation of the system during the first year of its operation

- the number of completed orders has increased by about 4.5%,
- utilization ratio of the trucks belonging to the customer's own fleet has also been increased,
- the number of delays to the customer has been reduced by 3.5%,
- complexity of calculations and the number of errors has been reduced,
- idle run of each truck has been reduced by 3-5%,
- downtime of trucks has been reduced by 5-7%,
- fines and penalties have been reduced by 2-4%,
- and there was also growth of other important indicators of resource utilization.

IV. ADAPTABILITY OF MULTI-AGENT SYSTEMS

Agents interacting with each other in the demand and resource system, form a solution of a complex general dynamic schedule task by dividing it into simple parallel tasks of placing orders at resources. When the current state is

getting worse, the agents focus on mitigating the impact of negative external effects. Agents of demand and resource network improve their own states locally and, thus, provide an increase in indicators of the whole system, for example, overall satisfaction.

When a new order, not distributed by the system, appears, the system satisfaction decreases at first, as the coming agent does not find the best position immediately. Only some time later, the general satisfaction starts increasing due to rescheduling and improvement of agent states. The system enters the state of non-equilibrium, and then agents aim at achieving a new local equilibrium state. In order to assess dynamics of multi-agent system, calculating the average satisfaction of demand and resource agents is suggested depending on time.

The adaptability degree of multi-agent system γ is introduced, which reflects the rate of local equilibrium recovery

$$\gamma = (y_2 - y_1) / T, \quad (1)$$

where γ is a minimal value of satisfaction after impact, y_2 is the average satisfaction of system agents after impact, T is the time needed for equilibrium recovery of the average satisfaction y_2 , see Figure 1:

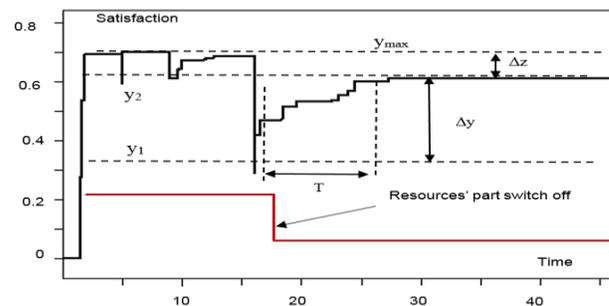


Figure 1. Adaptability of the system depending on partial resource switch off.

After maximum decrease of the average satisfaction to y_1 level, the system turns to a new quasi-equilibrium state y_2 after the time period T , and Δz is nonrecoverable lost satisfaction.

Partial recovery effect can be observed not only when resources are switched off, but also when new task flows occur discontinuously. The higher the adaptability rate, the higher the ability of agents to self-organization in elimination of negative effects.

V. THERMODYNAMIC DEMAND – RESOURCES NETWORKS SCHEDULING MODEL

Even scheduling for a small organization is a much more complex and dynamic task than it may first seem. Processes of self-organization are very similar to those described in the works of Ilya Prigogine and they play a very important role in scheduling within the developed systems [24][25].

Schedules in complex systems are to be considered as "unstable equilibriums" which means that they have different

strength in different directions. Each new order brings in the money that plays the role of energy in the system. The system demonstrates such non-linear events as order and chaos, autocatalytic reactions, fluctuations, etc. When the number of conflicts waiting for resolution increases and so does the number of messages in the system, it is assumed that the temperature of the schedule in this section is increasing as well. In order to support the schedule structure, agents pay some tax that dissipate part of input energy.

If the orders stop coming into the system, the schedule will later gradually break down starting with the weakest connections between the agents. Then the system will transition into the state of chaos having zero energy and eventually it will cool completely.

VI. CONCLUSION

The paper contains a review of multi-agents systems developed by SEC "Smart Solutions" from 2010 to 2014. It demonstrates that even today multi-agent systems make it possible to solve complex tasks and design industrial systems for resource management of a fundamentally new class, which are based on the main principles of self-organization and evolution. The developed methods and instruments for designing multi-agent systems can be applied for solution of a wide range of complex tasks – from cargo delivery to the International Space Station to railway transportation. This proves the high efficiency of the developed approach. Implementation results confirm evaluations made earlier which are the evidence of increase in efficiency of using resources by 20-40% by means of real-time decision-making. Implemented projects illustrate important advantages of the approach, such as service level increase, opportunity to solve very complex tasks of scheduling and resource optimization in real time, high operational efficiency and flexibility of solutions, reliability and viability of developed systems as well as minimization of risks for enterprises.

ACKNOWLEDGMENT

The study was supported by the Ministry of education and science of Russian Federation.

REFERENCES

- [1] P. Skobelev and V. Vittikh, "Models of Self-organization for Designing Demand-Resource Networks," *Automation and Remote Control*, Journal of Russian Academy of Science, no 1, 2003, pp. 177-185.
- [2] V. Vittikh and P. Skobelev, "The compensation method of agents interactions for real time resource allocation," *Avtometriya*, Journal of Siberian Branch of Russian Academy of Science, no 2, 2009, pp. 78-87.
- [3] P. Skobelev, "Multi-Agent Systems for Real Time Adaptive Resource Management," in *Industrial Agents: Emerging Applications of Software Agents in Industry*, P. Leitão and S. Karnouskos, Eds. Amsterdam: Elsevier, 2015, pp. 207-230.
- [4] P. Skobelev, "Multi-Agent Systems for Real Time Resource Allocation, Scheduling, Optimization and Controlling: Industrial Applications," *Proc. 5th Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems (HoloMas 2011)*, Springer, vol. 6867, 2011, pp. 1-14.
- [5] Y. T. Leung, "Handbook of Scheduling: Algorithms, Models and Performance Analysis", CRC Computer and Information Science Series, Chapman & Hall, London, 2004.
- [6] S. Chaleshtari and Sh. Shadrokh, "A Branch and Bound Algorithm for Resource Constrained Project Scheduling Problem subject to Cumulative Resources." *World Academy of Science, Engineering and Technology*, vol. 6, 2012, pp. 23-28.
- [7] S. Vos, "Meta-heuristics: The State of the Art". In *Local Search for Planning and Scheduling*, eds A Nareyek, Springer-Verlag, Berlin, 2001, pp.1-23.
- [8] C. R. Rolf and K. Kuchcinski, "Distributed constraint programming with agents". In *Proceedings of the second international conference on Adaptive and intelligent systems*, Springer-Verlag, Berlin, 2011, pp. 320-331.
- [9] L. Gongfa, "A hybrid particle swarm algorithm to JSP problem." *IEIT Journal of Adaptive & Dynamic Computing*, 2011, pp. 12-22.
- [10] Q. Xueni and Y. Lau, "An AIS-based Hybrid Algorithm with PSO for Job Shop Scheduling Problem". In *Proceedings of the tenth IFAC Workshop on Intelligent Manufacturing Systems, Lisbon, (IFAC' 2010)*, vol. 10, 2010, pp. 350-355.
- [11] M. Pinedo, "Scheduling: Theory, Algorithms, and System", Springer, Berlin, 2008.
- [12] P. Leitao and P. Vrba, "Recent Developments and Future Trends of Industrial Agents." In *Proceedings of the 5th International Conference on Holonic and Multi-Agent Systems, (HoloMAS 2011)*, Springer, Berlin, pp. 15-28.
- [13] T. Dong, "A Review of Convergence Analysis of Particle Swarm Optimization." *Int. Journal of Grid and Distributed Computing*, vol.6, no.6, 2013, pp.117-128.
- [14] M. Imrana et al., "An Overview of Particle Swarm Optimization Variants." *Procedia Engineering*, vol. 53, 2013, pp. 491-496.
- [15] A. Oliinyk, "The Multiagent Optimization Method with Adaptive Parameters." *Artificial Intelligence journal*, no.1, 2011, pp. 83-90.
- [16] S. Sun and J. Li, "A two-swarm cooperative particle swarms optimization." *Swarm and Evolutionary Computation*, vol. 15, 2014, pp. 1-18.
- [17] M. Tasgetiren et al., "Particle swarm optimization and differential evolution algorithms for job shop scheduling problem." *International Journal of Operational Research*, vol. 3, no. 2, 2008, pp. 120-135.
- [18] A. Ivashenko, I. Khamits, P. Skobelev P and M. Sychova, "Multi-Agent System for Scheduling of Flight Program, Cargo Flow and Resources of International Space Station." In *Proceedings of the 5th Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems (HoloMas 2011)*, Springer, vol. 6867, 2011, pp. 165– 174.
- [19] P. Skobelev, E. Simonova, A. Ivanov, I. Mayorov, V. Travin and A. Zhilyaev, "Real time scheduling of data transmission sessions in the microsatellites swarm and ground stations network based on multi-agent technology." In *Proceedings of the 6th Int. Conf. on Evolutionary Computation Theory and Applications (ECTA 2014)*, SciTePress, 2014. – pp. 153-159.
- [20] I. Belokonov, P. Skobelev, E. Simonova, V. Travin and A. Zhilyaev, "Multiagent planning of the network traffic between nanosatellites and ground stations," *Procedia Engineering: Scientific and Technological Experiments on Automatic Space Vehicles and Small Satellites*, vol. 104, 2015, pp. 118-130.
- [21] A. Belousov, P. Skobelev and M. Stepanov, "Network-centric approach to adaptive real-time train scheduling." In *ProcProceedings of the Sixth Int. Conf. on Swarm Intelligence (ICSI 2015)*, Springer, LNCS 9141, Part II, 2015, pp. 290-299.
- [22] V. Shpilevoy, et al., "Multi-agent system "Smart Factory" for real-time workshop management in aircraft jet engines production." In *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS'13)*, IFAC, 2013, pp. 204-209.
- [23] P. Skobelev, A. Lada and I. Mayorov, "Finding an initial plan of transport resources FTL allocation in a special VRP problem using linear programming methods." In *Proceedings of the 19th World*

Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2015), in press.

[24] I. Prigogine and I. Stengers, Order out of Chaos: Man's new dialogue with nature. Flamingo, 1984.

[25] G. Rzevski and P. Skobelev, Managing complexity. WIT Press, London-Boston, 2014.

Modeling the Dynamics of Insulin-Glucose Subsystem Using a Multi-agent Approach Based on Knowledge Communication

Sebastian Meszyński, Oleksandr Sokolov
 Faculty of Physics, Astronomy and Informatics
 Nicolaus Copernicus University
 Toruń, Poland
 {sebcio, sokolov}@fizyka.umk.pl

Abstract — **Mathematical analytical modeling and computer simulation of the physiological system is a complex problem with a great number of variables and equations. The objective of the present research is to describe the insulin-glucose subsystem using multi-agent modeling based on intelligence agents. Such an approach makes the modeling process easy and clear; moreover, new agents can easily be added to the investigations.**

Keywords - *multiagent paradigm; compartments; normoglycemia; insulin-glucose system; physiology.*

I. INTRODUCTION

Expert systems have emerged as developed methods of artificial intelligence in the eighties of the last century. The most-widely used definition of an expert system is the one which defines an expert system as a computer program that uses the procedures for requesting a solution to non-algorithmic problems, or, in other words, problems requiring the participation of experts in the field (experts) in order to benefit from their knowledge or expertise. At the moment, expert systems are these computer programs which perform complex tasks with high requirements on intellectual capacity, and manipulate large amounts of input data. The basic features distinguishing expert systems from other computer programs are as follows:

- the use of knowledge instead of the data. It is stored separately and independently from the rest of the executive system.
- the knowledge base is created and stored in a symbolic form. Any kind of inference is conducted on the basis of the symbolic computations that can be compared to the manipulation of human natural language. A symbolic representation of knowledge is understood as all kinds of rules, semantic networks.
- the reasoning often involves drawing conclusions from the so-called meta-knowledge and are used for learning mechanisms - possessed knowledge becomes enlarged (generalization) with a new phenomenon or is updated in the form of a replacement of the wrong rules with the valid rules; such systems are also able to justify given answers corresponding to explanations of a deductive process.

Extended possibilities of expert systems in the fields of the knowledge-acquirement speed, the analysis of the potential solutions to a problem and the decision concerning a definitive answer, led to the development of multi-agent systems. The main difference between systems and multi-agent systems is based on the indirect contact of expert systems with the environment. Input does not come from sensors, but from an additional agent which is usually human. Multi-agent systems are complex systems, from communication and cooperation among agents themselves to pursuit of a common objective. This design allows troubleshooting of a diffuse character or computationally complex one. When teaching multi-agent systems, the concept of agent is presented as an autonomous object with the initiative of action based on the observation of the environment in which it is located. It also has the ability to use the resources of this environment, and the motivation is to solve the problem posed in front of it. This definition forces the agent to know sensor inputs through which it can receive signals from the environment, and effectors which will be able to influence the surrounding environment. The most important task of the agent is to decide which of the possible courses of action is the best at the time of knowledge about the problem to achieve this goal.

This branch of science concerns the resolution of the nature diagnostic problems and therapeutic nature using a large base of knowledge and a broad spectrum of causes and effect relationships between different states of health of the patient and the interaction between different treatments [1][2][3] which should simultaneously be performed as part of the patients' overall treatment plan. This rather specific branch of science is based on expert knowledge - here the doctor is a good candidate to use artificial intelligence systems. These systems are an addition to proper disease diagnosis, correct diagnosis, and proper treatment process in order to overcome the disease or reduce complications of the disease. Examples of medical problems should also mention the following: databases of patients, search for medical knowledge, support for medical decisions, evaluation of the efficacy, etc. At the same time, the use of multi-agent systems is implemented in order to eliminate the drawbacks of the previous solutions in medical systems that showed a limited autonomy, limited interaction with the environment and a smaller store of knowledge to solve a problem. The new directions of research focus, for example, on the treatment of advanced stage by many physicians at the same

time. See [4][5][6][7][8]. The examples of the use of these systems in order to improve the quality of treatment can be found in many works. They can also be used as part of the healthcare system, ranging from sending ambulances to optimize the route, choosing a hospital that is best equipped with the apparatus necessary for the patient transported, to supporting drug treatment and analysis of its implications.

More information on the application of multi-agent systems in medicine and related sciences can be found in publications [9][10][11]. In the publication [12], multi-agent systems have been subjected to critical analysis as a result of which it is possible to define the areas of application describing artificial intelligence systems and limitations that must be taken into consideration in order to benefit from these opportunities.

II. THE USE OF MULTI-AGENT SYSTEM IN PHARMACOKINETICS

A. Compartments theory

A separate group of medical problems are the methods of description, analysis and prediction of the dynamics of the selected substances in a human body. The most common methods used for the description and analysis of these phenomena are based on differential equations, whose construction and analysis are based on major or minor assumptions and approximations. One of the said methods is described by using a compartments model, as illustrated in Fig.1. On the basis of such a model, the appropriate equations describing the dynamics of the system created are built. The modeling assumes the existence of so-called compartments in the body where a uniform concentration of the substance to be analyzed is located.

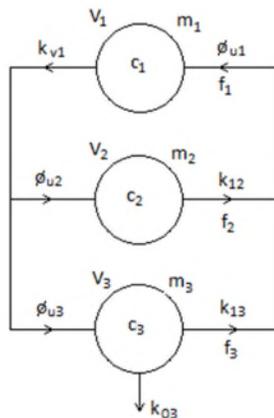


Figure 1. An example of a compartment model (the model has three compartments here).

In some models, the number of compartments may be higher due to an attempt to approach mathematical description of transport of the substance in the body to real physiological value, and fostering real pharmacokinetics of the substance. In the description, the multi-compartments model is a particularly useful application of matrix (1) which, in the final finished form, allows the calculation of each point of the interest stream.

However, it should be emphasized that a task of this type is difficult, or, at least, problematic.

$$\frac{dm(t)}{dt} = \mathbf{A}m(t) + \mathbf{F}d(t) \tag{1}$$

satisfying the initial conditions:

$$\mathbf{m}(t = 0) = \mathbf{m}(0) \tag{2}$$

Finally, it can be shown that the solution is determined by the formula:

$$\mathbf{m}(t) = e^{\mathbf{A}t}\mathbf{m}(0) + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{F}d(\tau) d\tau \tag{3}$$

Given the complexity of physiological systems, one often seeks to get a quantitative result rather than qualitative.

B. The compartment model in terms of multi-agent paradigm

In this view, the agent is to be understood as any clearance reflecting the specified area of the body which changes the concentration of the test substance. The agent's behavior is defined by the function describing the distribution of substances seen as a change in its output stream φ_{01} . If the agent represents a compartment through which a certain amount of the tested substance flows, then this agent (its behavior) is defined by an appropriate transmittance reflecting changes in the output to changes in the agent at the entrance agent – Fig. 2.

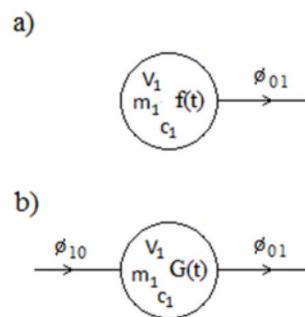


Figure 2. Agent concept diagram representing the compartment:
a) the agent as the source of the substance,
b) the agent intermediary in the flow of the substance.

The description of one compartment agent model means determining the flux distribution function φ_{01} and the time step t defining the rate of change of the flux. In addition, one must declare the initial conditions - in this case the level of drug concentration $c(0)$ at $t = 0$. The initial level determines the value at which the stream will begin to change over time. Because the function is defined as the behavior of the agent, it will show the direction and the rate of change. The flow of the elimination of substances is given by:

$$\varphi_{01} = f(t) = k_{01}m_1(t) \tag{4}$$

and, by using the formula for the change of mass over time:

$$\frac{dm_1(t)}{dt} = -k_{01}m_1(t) \tag{5}$$

We obtain the function to determine changes in the weight of the agent:

$$m_1(t) = m_1(0)e^{-k_{01}t} \tag{6}$$

where $m_1(0)$ represents the mass of a drug at time $t = 0$.

Therefore, the effluent of the agent can be represented as:

$$\varphi_{01}(t) = k_{01}m_1(t) = k_{01}m_1(0)e^{-k_{01}t} \tag{7}$$

One can also specify the variation of weight of the $U_{01}(t)$ excreted with the agent in the process of elimination:

$$U_{01}(t) = \int_0^t \varphi_{01} dt = m(0)(1 - e^{-k_{01}t}) \tag{8}$$

For the purposes of multi-agent simulation environment, the time t is a discrete time in which the agent generates a "signal" for the appropriate behavior.

$$t_{k+1} = t_k + \Delta t \tag{9}$$

where Δt is the time step increment function describing the agent behavior.

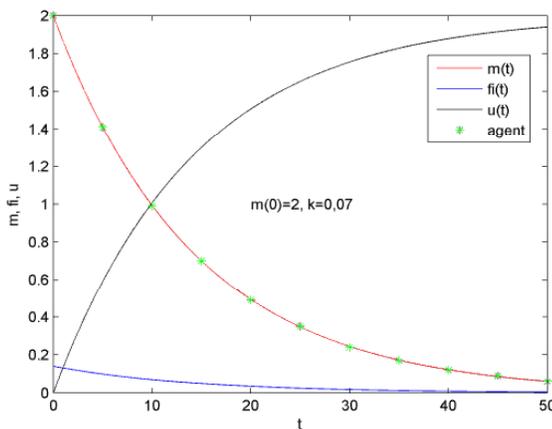


Figure 3. Simulation graphs for the equations (6) (7) (8) and the value generated by the agent.

III. THE OVERALL CONCEPT OF MULTI-AGENT MODEL

A. Multi-agent concept

The section below will focus on the concept of a multi-agent system where the work is aimed at the restoration of glucose homeostasis. The amount of glucose supplied from the gastrointestinal tract into the blood depends on the amount, composition and frequency of meals. On the other

hand, the energy demand of tissues and organs is variable. The concentration of glucose in the blood of a healthy individual is maintained within relatively narrow limits of about 4.5 - 9.0 mmol/L (81 - 162 mg/dL). Mechanisms of preventing glucose concentration decline in the blood as well as its excessive growth are extremely important for the proper functioning of the body. These mechanisms act on the substrate, hormonal, and nervous system.

The principle of the mechanism of the substrate is directly controlled by the change of glucose in tissues depending on the flow to the cells, or the availability of the other substrates.

As a part of the hormonal control, the most important hormone lowering blood glucose – insulin – should be considered. The effect of insulin in the liver mainly involves stimulation of glycogen synthesis, and inhibition of gluconeogenesis. The insulin found in muscle and fat will affect the glucose transporter proteins across cell membranes stimulating glucose uptake by these tissues as well as stimulating glucose oxidation and glycogen synthesis [13]. The indirect effects of insulin uptake, oxidation and size of the glycogen involves its rate inhibiting effects of lipolysis and oxidation of fats [14][15]. The nervous control of blood glucose is mediated by two branches of the autonomic nervous system. The parasympathetic nervous system stimulates glycogen synthesis in the liver and secretion of insulin, contributing in this way to lower blood glucose.

The activation of this system is associated with the consumption of a meal (olfactory stimuli, taste), and the presence of food in the gastrointestinal tract.

Based on the above-described three ways of adjusting the level of glucose in the blood, we propose a multi-agent model that reflects, as much as possible, the mechanism of action and structures responsible for normoglycemia. Considering the above, we propose a layered multi-agent model which closes an appropriate regulatory mechanism in each layer.

The proposed model consists of three layers – Fig. 4:

- layer 1 - base layer where the agents representing the cell are located. This layer reflects the basic building block of individual cell structure of the body's organs. This layer processes occur on/in a cell scale . Layer 1 can be called the cells layer.
- layer 2 - layer of organs, which enables communication between them through biochemical signals. It is the layer of the actual process of normoglycemia. Layer 2 can be called the physiological layer.
- layer 3 - layer representing the selected areas of the brain directly related to the stabilization process of glucose. This layer simulates the processes involving information flow control of the glucose and insulin dynamics in the blood, and provides the opportunity to simulate the psychological stimuli that affect blood sugar levels. Layer 3 can be called the psychological layer.

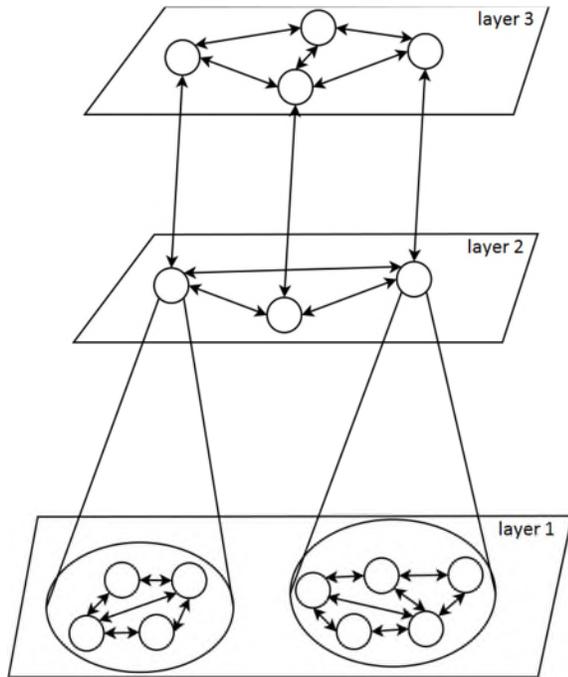


Figure 4. The idea of layered multi-agent model.

In the described approach, the emphasis is on the modeling of organs which play an active role in glucose homeostasis.

It should be stipulated that having this layer representing the single-cell level create the opportunities for simulation also in the cellular scale and molecular - if incorporated into the cell interaction and exchange between biochemical information.

Adopted layers number 2 and 3 are constructed in accordance with the agents representing organs involved in normoglycemia, and the environment fuzzy logic programming language to build a simple knowledge base. The multi-agent environment is built on the basis of the JADE environment and programming language Java. The agents act as the appropriate organ (pancreas, liver, adipose tissue, gastrointestinal tract as a source of food, and insulin-independent mechanism for glucose utilization). Each agent is assigned its own task, in the form of behavior described by using the tool, or knowledge base. The first description applies to a situation in which the agent is the source medium, i.e. food in the form of glucose. Then the agent is treated as the one producing its own interior medium which then goes into the environment common to all agents. Specific interactions between agents are shown in Fig. 5. This approach allows to make more complex and advanced analysis than the models based on differential equations. The use of this type of multi-agent model has many advantages over analytical methods:

- Rules can be easily modified
- The objective function and the definitions of limitations may be more complex
- The attributes of individual organs / agents can be easily defined

- More opportunities to analyze simulation results.

Such agents are obtained by subtracting from the differential equation members who bring changes representing the response of the organ increases or decreases in glucose and / or insulin to the equation. Fig. 6 presents a concept for the equation based on Hardy-Stolwijk to show a method of identifying agents that are presented in Fig. 5.

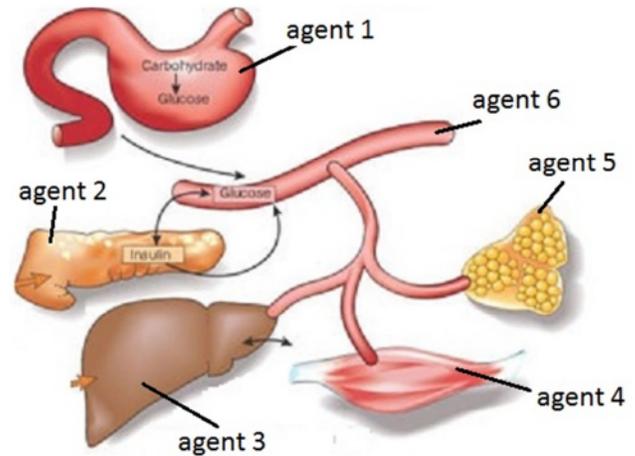


Figure 5. A schema connection between agents-organ.

$$\frac{dg}{dt} = \overset{\text{agent3}}{\omega} - \overset{\text{agent4}}{vgi} - \overset{\text{agent5}}{\lambda g} + \overset{\text{agent1}}{G}$$

$$\frac{di}{dt} = \overset{\text{agent2}}{-ai + \beta(g - \psi)}$$

Figure 6. Equation of Stolwijk-Hardy model used to separate the functional parts of multi-agent system.

B. Agent dynamic models based on fuzzy logic rules

Each agent of glucose-insulin model answers to its behavior and products the output according to its inner state. For instance, the agent representation of Stolwijk&Hardy model gives the members of the right side of the equations an output of the agent. These members have parameters (alfa, beta, etc.) whose values depend on the case modeled (healthy, or ill man). Moreover, during the simulation, we have the same values of these parameters that produce the linear response and not always are realistic (the STELLA model is used for this correction).

We propose generating the appropriate output using fuzzy rule bases that can produce any complex nonlinear

function. Moreover, such an approach can be easily accepted by a medic due to their simplicity and clearness.

The simplest way to provide fuzzy rule base is illustrated in the example of member ϑgi of Stolwijk&Hardy model. Agent 4 is response to this output.

If parameter $\vartheta = Const$, we can use the member ϑgi as the simplest output. In other cases, we have to use "If" condition, or rule (for instance, in the cases of healthy man, or type-II diabetes one). In a more complex behavior, we have to construct the set of rules (rule base).

The fuzzy rule base considers all variables as fuzzy sets with appropriate membership functions. Moreover, we can perfectly avoid now the parameter ϑ , which is not easy to understand for a medic.

Let us consider the fuzzy sets for variables insulin i and glucose g . After normalization, the membership functions for these variables can be shown as in Fig.7.

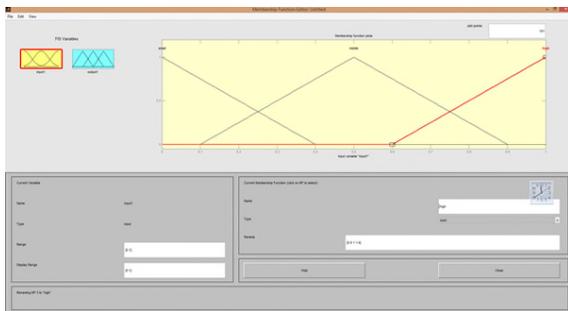


Figure 7. Membership functions of variables insulin and glucose (normalized).

The rules can be formed in Mamdani or Taskagi-Sugeno manner. In Mamdani model, we use fuzzy sets both in premise and conclusion of the rules. Fig. 8 shows the variables connection.

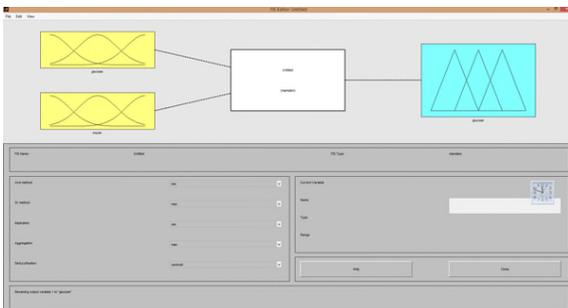


Figure 8. Structure of agent's rules.

Therefore, in our case, we could create 9 rules for fuzzy sets of variables insulin i and glucose g with 3 membership functions. The rules have the following form:

If Insulin is Small **and** Glucose is High **then** Glucose is Middle.

The set of rules for agent 4 is shown in Fig. 9.

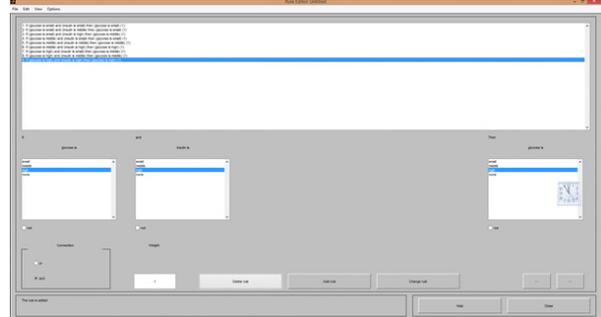


Figure 9. A set of rules for agent 4.

The Mamdani style inference is shown in Figure 10.

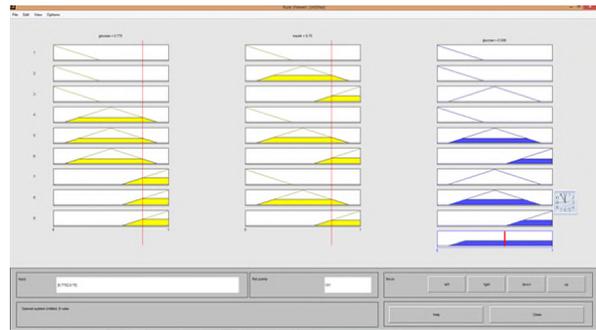


Figure 10. Inference engine of Mamdani model.

The function of agent output is shown in Figure 11.

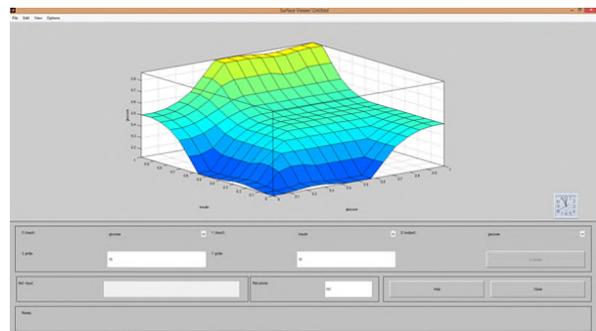


Figure 11. Surface of rule base mapping.

The advantages of the approach above include easiness and clearness for agent output description (we connect only semantic variables like insulin i and glucose g without additional coefficients like ϑ) on one hand, and complex nonlinear behavior of the agent, on the other.

IV. RESULTS

Thanks to our proven analytical methods for solving the equations in Fig. 6, we achieve an objective method for determining the benefit of the resulting simulation based on a the multi-agent system. In order to verify the correct operation of multi-agent system, it was asked to write the software multi-agent system in JAVA with using JADE multi-agent environment. As a result of the development of

the agents that have been implemented in philosophy according to Fig. 5, simulation results such as shown in Fig. 12 were obtained.

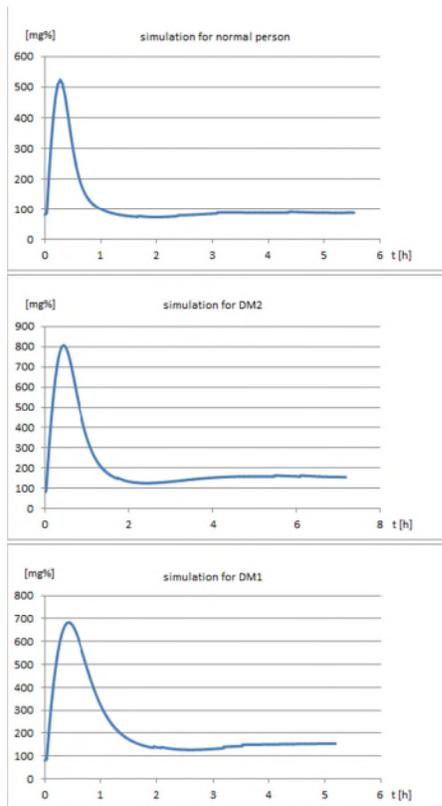


Figure 12. The results created based on work of multi-agent system.

V. CONCLUSION AND FUTURE WORK

The agent approach allowed us to modify the dynamic model of insulin-glucose system in the direction of modeling of inner behavior of each agent and communication with each other. Such modeling gives the possibility to address more complex behavior (not only with coefficients in the equations like in Fig. 6 but using fuzzy relations between parameters).

With the presented approach, we obtained satisfactory results that coincide with the results obtained from the simulation of the analytical equations. The use of fuzzy logic allows to dispense with rigidly assigned coefficients (parameters) models, allowing a more flexible approach. One can take into account many factors that can shape the response to individual agents to changes in their environment, through which this study should understand the changes in insulin sensitivity, body weight changes, efficiency of secretion of insulin, glucose half-life, e.g. Our next efforts will be directed towards the communication

among agents for exchange of information about current states and to take a common decision.

REFERENCES

- [1] B. L. Iantovics, "A Novel Mobile Agent Architecture. Proceedings of the 4th International Conference on Theory and Application of Mathematics and Informatics," Albac county, Acta Universitatis Apulensis, Vol. 11, 2006, pp. 295–306.
- [2] B. L. Iantovics, "Cooperative Medical Diagnosis Systems. Proceedings of the International Conference Interdisciplinarity in Engineering," Tg. Mures, 2005, pp. 669–674.
- [3] R. Unland, "A Holonic Multi-Agent System for Robust, Flexible, and Reliable Medical Diagnosis," In: R. Meersman, Z. Tari (Eds.): OTMWorkshops 2003, Springer-Verlag, LNCS, Vol. 2889, 2003, pp. 1017–1030.
- [4] J. Ferber, "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence," Addison Wesley, 1999.
- [5] A. Vesnenko, I. Popov, A. A. Pronenko, "Topo-Typology of the Structure of Full-Scaled Clinical Diagnoses in Modern Medical Information Systems and Technologies," Plenum Publishing Corporation Cybernetics and Systems Analysis, Vol. 38, 2002, No. 6.
- [6] B. L. Iantovics, "A Novel Diagnosis System Specialized in Difficult Medical Diagnosis Problems Solving," Emergent Properties in Natural and Artificial Systems, Understanding Complex Systems, Springer-Verlag, Heidelberg, 2006, pp. 187–197.
- [7] St. Kim, "Ubiquitous Healthcare: The OnkoNet Mobile Agents Architecture," In: M. Aksit, M. Mezini, R. Unland (Eds.): Proceedings of the 3rd International Conference Netobjectdays. Objects, Components, Architectures, Services, and Applications for a Networked World (NODE 2002), Springer-Verlag, Germany, LNCS, Vol. 2591, 2003.
- [8] J. Huang, N. R. Jennings, J. Fox, "An Agent-Based Approach to Health Care Management," International Journal of Applied Artificial Intelligence, Vol. 9, 1995, No. 4, pp. 401–420.
- [9] P. Redou, S. Kerdelo, C. Le Gal, G. Querrec, V. Rodin, J. F. Abgrall, J. Redou, "Reaction-agents : first mathematical validation of a multi-agent system for dynamical biochemical kinetics," Lecture notes in computer science, springer, 2005, 3808, pp.156-166.
- [10] V. Rodin, G. Querrec, P. Ballet, F. Bataille, G. Desmeulles, J. F. Abgrall, J. Tisseau, "Multi-Agents System to model cell signalling by using Fuzzy Cognitive Maps," Application to computer simulation of Multiple Myeloma., 2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering.
- [11] M. A. de Cerqueira Gatti, C. J. Pereira de Lucena, "An Agent-Based Approach for Building Biological Systems: Improving the Software Engineering for Complex and Adaptative Multi-Agent Systems," Monografias em iência da Computação, number 14/07, ISSN 0103-9741.
- [12] F. Amigonil, V. Schiaffonati, "Multiagent-Based Simulation in Biology: A Critical Analysis,"
- [13] D. Kelley, A. Mitrakou, H. Marsh, F. Schwenk, J. Benn, G. Sonnenberg, M. Arcangeli, T. Aoki, J. Sorensen, M. Berger, "Skeletal muscle glycolysis, oxidation, and storage of an oral glucose load," J. Clin. Invest. 81, 1563–1571.1172/JCI113489.
- [14] P. J. Randle, P. B. Garland, C. N. Hales, E. A. Newsholme, "The glucose fatty-acid cycle. Its role in insulin sensitivity and the metabolic disturbances of diabetes mellitus," Lancet. 1963;1:785–789.
- [15] M. Roden, T. B. Price, G. Perseghin, K. F. Petersen, D. L. Rothman, G. W. Cline, G. I. Shulman, "Mechanism of free fatty acid-induced insulin resistance in humans," J Clin Invest 97: 2859-2865, 1999

Multicast Routing for High-Quality Multimedia Environments: Deployment and New Problems

Pavel Troubil, Hana Rudová

Faculty of Informatics
Masaryk University
Brno, Czech Republic

Email: pavel@ics.muni.cz, hanka@fi.muni.cz

Petr Holub

Institute of Computer Science
Masaryk University
Brno, Czech Republic

Email: hopet@ics.muni.cz

Abstract—Motivated by remote interactive collaboration over high-quality multimedia, we solve the multigroup multicast routing problem under critical runtime limits and we focus on realistic aspects crucial for deployment of advanced collaborative environments. We developed solvers based on mixed integer programming and ant colony optimization, which are integrated in currently developed environments to support education of deaf students at Czech universities. They are also deployed for distributed media production for live broadcasting of sporting and cultural events. The paper presents our ongoing research and development aims in deployment of the environments and new arising problems in multicast routing.

Keywords—multicast routing; planning; multimedia; interactive collaboration

I. INTRODUCTION AND MOTIVATION

The recent years have witnessed rapid performance growth of commodity hardware and its multimedia processing capabilities. The progress stimulated development of interactive applications, which can use high-quality media for communication and remote collaboration. Cheaper deployment of such applications brought remote work and cooperation into new areas, like distributed musical performances [1], or distributed media production in broadcasting. New opportunities arose in other areas, e.g., telemedicine [2] and remote scientific visualization. In the pursuit to create self-organized collaborative tools which will support these novel forms of remote interactive communication, we develop methods for automated planning of concurrent network transmissions. They are essential for intelligent behaviour of the collaborative tools, as they eliminate the need of manual setup. The most recent solvers enable effective response to dynamics of computer networks while scaling to medium-sized collaborating groups, supporting 20–30 participants in a videoconference.

We target applications with extensive demands on quality of service provided by computer networks, particularly capacity and end-to-end delay. The high-quality interactive multimedia applications usually transmit video streams with high image resolutions, either compressed by codecs with low compression rates or uncompressed at all. Such videos in UHD (Ultra High Definition) or 4K resolution need a network capacity between 2–24 Gb/s. In order to maintain image without visible quality degradation for applications like telemedicine, the transmitted bandwidths always remain relatively high in comparison to available capacities, although networks may enforce higher compression rates. For example, Digital Cinema

standard [3] specifies JPEG2000 compressed formats with up to 250 Mb/s bandwidth for 4K video resolution. Low compression rates do not only increase image quality and required bandwidth, but are also encoded faster than codecs with high compression rates. In addition, compression latencies contribute to the total transmission latency from source to destination, which should be minimized for interactive applications. The International Telecommunication Union in their recommendation G.114 [4] declares that end-to-end latencies above 150 ms introduce negative impact on users' perception of interactivity in remote communication.

On top of the UltraGrid [5], a software for low-latency transmissions of high-quality video, the CoUniverse middleware [6] is used to orchestrate the collaborative environments, i.e., configure multiple point-to-point transmissions to build complex interactive applications. This paper aims to present to the community our ongoing work on transmission planning problems in the CoUniverse and methods which we proposed and implemented to solve them. We will also describe the current development of novel collaborative environments with specialized applications, and new planning problems which they introduce.

The CoUniverse is driven by a user-empowered approach, i.e., any user can run the collaborative environment on her own and has full control over all applications within. Running the environment shall be available at any time, as simple as a Skype call, and allow top-tier quality of multimedia. The transmission planning is then similar to peer-to-peer videoconferencing tools, e.g., Celerity [7]. The common denominator is a need to operate without any administrative privileges to computers nor network. As a result, knowledge of physical topology of an underlying network is incomplete or missing at all. Network administrators do not publish the topologies, since they need to modify them anytime, usually in reaction to load or failures. Only some experienced users might have partial knowledge of the underlying topology, usually their local network and its neighborhood. Planning of transmissions relies often on knowledge of overlay topology, i.e., mutual visibility among the collaborative tools similar to reachability by the ICMP (Internet Control Message Protocol) ping. And since native IP multicast is often unavailable across independently administered networks [8], the user-empowered environments also require user-empowered substitutes for multipoint communication called Application level multicasts (ALM) [9].

The problem of planning network transmissions is known

as multicast routing in literature on combinatorial optimization [10]. The multicast routing is either single-group, i. e., with single data source which delivers data to a set of destinations, or multigroup with multiple data sources and an independent set of destinations for each of them. Mixed integer programming methods were proposed for both single and multiple groups [11]. But for scalability reasons, current research is focused on heuristics, particularly approaches from computational intelligence. For the single group, genetic algorithms are repeatedly revisited, e. g., [12], and ant colony optimization (ACO) techniques are also proposed often [13] [14]. The multigroup problem is harder to solve and is also less studied. Again, the genetic algorithms are the most successful [15].

The rest of the paper is structured as follows. In Section II, we formulate the problems and mention methods applied to solve them. Section III sketches two practical environments, where we implement and deploy solvers proposed in Section II. In Section IV, we analyze the modified routing problems which arise from the deployment.

II. PROBLEMS AND METHODS

We model the computer network by a directed weighted multigraph $G = (V_o \cup V_u, E_o \cup E_u)$. The graph covers two layers of network representation. Overlay nodes V_o are computers running the collaborative tools and overlay links E_o represent mutual reachability of the tools. Knowledge of this layer is always available, including latencies and (potentially inaccurate) capacities of the overlay links. Yet, it is not always sufficient for reliable planning decisions. Overlay links may share underlying physical network infrastructure, but the sharing is not captured in the representation of the overlay links, and may result in overloading of the physical infrastructure when several overlay links are incorrectly considered as independent. The underlay nodes V_u and underlay links E_u represent the underlying network where available — even when it is only part of the topology. In such case, knowledge of association between overlay and underlay links is required. Capacities of some underlay links (but not necessarily all) can be known.

For each data source $s \in S$ ($S \subset V_o$), there is a set of destinations $D(s)$, which request transmission from the source s . The overlay nodes, which are neither sources nor destinations, are called ALM agents. The agents can receive the data, forward them further, replicate them, and also process contents of the data. They provide infrastructure for multipoint communication by mimicking of native IP multicast in the overlay network. Solution of the multigroup multicast routing problem is a set of multicast trees, one tree per source. Each multicast tree is a subgraph of the overlay network topology graph (V_o, E_o) . The tree is directed, rooted in the source s , and set of its leaves equals to the set of destinations $D(s)$.

For each link in all trees, we also have to decide the multimedia format for transmission along the link. Multimedia can be encoded in various formats. Video formats differ, among others, in image resolution, framerate, compression algorithm, and its parameters. For easier representation and simpler solution, we abstract from these attributes and work with a discrete set of formats, which are common in practice. The attributes are expressed in bandwidth required for transmission of the formats, subjective quality perceived by receiving users,

latency introduced by on-the-fly encoding or decoding, and amount of hardware resources for the encoding or decoding. The ALM agents are able to transcode multimedia formats on-the-fly, i. e., decode them and encode in a different format. Exploiting this capability, the planning can assign different formats to each incident link of an ALM agent in a multicast tree. Each destination in a multicast tree can receive data in a different format, which shall provide as good quality as possible and be adapted to destination's network capacity.

Solutions have to respect capacities of network links, limits of hardware resources (performance of processor and graphics card or available memory) at overlay nodes, and sometimes also (un)availability of features required for processing of some formats at the nodes. We aim to optimize two components of objective function: minimize total transmission latency, which includes latency of data transmission as well as delay induced by multimedia processing at overlay nodes, and maximize subjective quality of multimedia received at destinations.

We solved the above mentioned problem by methods of mixed integer programming (MIP). At first, we analyzed the problem without transcoding and knowledge of underlay network topology [16], and evaluated suitability of several cycle-avoidance methods adopted from the travelling salesman problem. Later, we extended our MIP models with the underlay knowledge and support of format selection [17]. The planning of multimedia transcoding is the main distinction of our methods from related work on multicast routing.

Next problem extension introduces uncertain network capacities. The user-empowered approach and inherent inaccuracy or absence of network knowledge can be partially overcome with automated tools for network topology inference [18] and bandwidth estimation [19]. In addition to uncertainty in network parameters caused by continuously changing state of the network, these automated tools provide results with only limited reliability. In some of our tests, the tools misestimated actual bandwidth by more than 50% on 1 Gb/s links. Instead of assuming that capacities of overlay network links are known exactly, we model them by cumulative distributive functions as is usually the case in network problems with uncertainty [20]. We solved the extended problem by a metaheuristic algorithm based on ant colony optimization, which fits well with the uncertainty and allows sufficient scalability [21].

The iterations of the algorithm adhere to the classical structure of iterations in the ACO. First, the solutions are generated for all data sources in parallel. We adopt the tree-building approach similar to single group algorithm from paper [13], where ants construct the trees by moving in network topology graph. In a next phase, the individual multicast trees are gathered and their load on the network summed. The results are used to set parameters for tree generation in subsequent iterations. Finally, the pheromones are deposited on links used by the trees in an amount proportional to the quality of the solution.

The design of the algorithm also allowed its simple application on the multicast routing problem with dynamic reconfiguration [22]. The dynamic reconfiguration is needed when a solution is deployed, a communication is underway, and the input for planning changes as a result of change in network state or source and destination sets. In such situations, the problem needs to be solved again with ongoing aim to

optimize latency and quality, but data transmission paths shall be modified as little as possible. Every change of path from source to a destination interrupts the communication for several seconds and disturbs users. The algorithm in paper [22] is designed to avoid the interruptions, but does not guarantee path preservation.

Since our algorithms target area of interactive applications, they also have to provide interactive response when the user is waiting for establishment of the communication or a dynamic reconfiguration. The ACO algorithm can solve problems with hundreds of overlay nodes within a second [21] [22].

III. CURRENT PROGRESS IN DEPLOYMENT

We are currently working on future deployment of the planning algorithms within the CoUniverse, focusing on two application areas: distributed media production for live broadcasting and education of deaf students at Czech universities.

A. Distributed Media Production for Live Broadcasting

Live broadcasting from sporting and cultural events requires a lot of personnel and specialized technical equipment, which is transported to venues by many production trucks (vehicles equipped with production control rooms). Technology usually used for live broadcasting requires live editing directly at the venue. The director switches in real-time between available cameras, always selecting one source for transmission to the broadcasting center via a satellite connection. High prices of broadcasting transmission routes enforced the presence of large teams and equipment at the venue, since their expensive transport between the venues was still cheaper. These days, computer networks with sufficient capacity cover most of the venues and will allow to substitute original broadcasting routes via satellites with video transmissions over computer networks. Still, many venues will be connected through low-capacity network links in comparison to network core, but on-the-fly multimedia transcoding can overcome the limitations, and will allow to reduce expenses on transport of personnel and the equipment.

Media production in live broadcasts could follow the scheme depicted in Figure 1. Two concurrent sporting events are held at two venues. The director works from the broadcasting control center. Content from all cameras at both venues is delivered in preview quality to the broadcasting control (red arrows). The preview quality requires much lower bandwidth, but is sufficient for video cutting decisions. Based on the decisions (green arrows), video content from one source is delivered to the distribution center in full quality. The editor can switch in real-time among all the sources, and the distribution center feeds the selected content to satellite distribution or terrestrial broadcasting networks (blue dashed arrows). We demonstrated a prototype of such live broadcasting system at The 14th Annual LambdaGrid Workshop [23], where we transmitted an ice hockey match captured by 5 cameras from Czech Republic to New Zealand. Building upon experience from the demonstration, another one will be prepared for the 15th Annual LambdaGrid Workshop in autumn 2015. The demonstration shall present multiple improvements of the system, particularly features important for the broadcasting practice. We expand on them in Section IV.

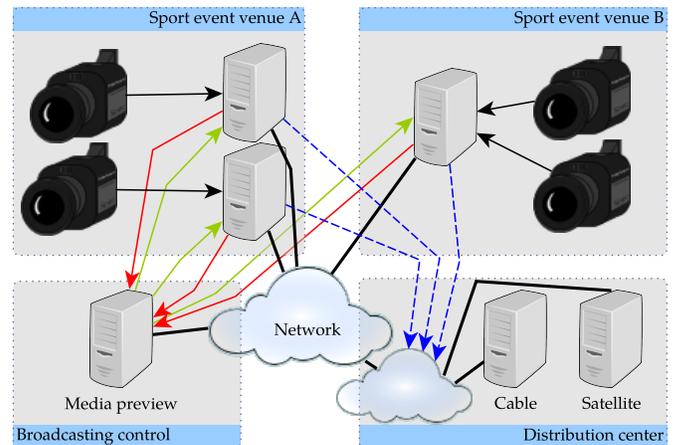


Figure 1. Scheme of live broadcasting infrastructure

B. Remote Interpreting of Sign Language for Education

Integrated education of deaf and hard-of-hearing students poses a challenging problem for any educational facility, but becomes especially difficult at universities. The narrow specialized fields of study use a lot of technical terms and courses are often taught in a foreign language. Should the deaf students have an option to visit lectures and seminars, highly skilled interpreters of sign language are required, who know the specialized vocabulary and/or are able to interpret the foreign language. Since such skilled interpreters are scarce, they are often required to inefficiently travel among universities. Videoconferencing can offer an effective and efficient alternative comfortable for students, interpreters, and teachers. Better education could be provided to more deaf students with lower costs.

The videoconference remote interpreting (VRI) has been studied mainly for legal purposes at the police station, in court, or healthcare. For example, see the Avidicus project [24]. Current research is focused rather on psychological implications of the remote interpreting than technological aspects. VRI usually uses only a single video and audio channel, but deployment in university lectures requires advanced solutions. The Support Center for Students with Special Needs at Masaryk University (SCSSN) designed the CoUnSiL, a specialized environment for remote interpreting in lectures, where each student has his own screen with picture of teacher, her presentation, interpreter, and other students. Since students need to follow and understand long lectures and seminars, the visual quality of the videos has to be maximized. As illustrated by picture-in-picture interpreting on TV, the sign language is comprehensible even if the picture is low quality. Yet, the low quality is not suitable for education, since it wears the deaf out very quickly and diminishes their ability to comprehend and retain information. Also due to a difficult setup of the environment, deployment of the CoUniverse is natural. In cooperation with SCSSN, we are adapting the CoUniverse for the education of deaf students. Testing of the prototype received a positive response from future users, and deployment for regular teaching across several Czech universities is expected during 2016.

IV. NEW PROBLEMS

The development of the environment for live broadcasting brought to light several new extensions of the problem. The camera switching feature requires a change of approach to transition between two solutions during a dynamic reconfiguration. Until now, our algorithms did not support any elaborate transition procedure. Yet, the live broadcasting has special requirements on cuts when switching between two sources. For good visual experience of spectators, the cut has to take place exactly between two video frames. But on the packet level, the frames cannot be so easily separated, and borders between frames are not aligned among individual sources. The problem of seamless switching can be solved by concurrent transmission of data from both sources to a specialized switching ALM agent. While receiving both data sources, such ALM agent can decide the right moment to stop transmission of the original stream and start the new one, when the cut shall be visually clear. But the concurrent reception doubles requirements on incoming network capacity of the processing node. The capacity can be reserved for two sources at a time during the entire broadcast, decreasing quality of received media all the time. Another solution of the problem could involve a multistep transition. First, the original source would transmit lower bandwidth (and quality as well) to the switching ALM agent. Second, transmission of the new source to the agent would start. Third, the original source would stop transmitting to the switching agent, and finally the freed capacity would be filled by the new source. As a result, spectators would see the lower quality picture only during the transition phase, not all the time. From the routing point of view, the multiphase transitions require not only switch between original and new solution, but four solutions deployed in relatively quick succession.

Visually clear cuts require concurrent reception of two sources at a switching ALM agent. There could be also additional requirements on such node, e. g., a specialized hardware or software equipment for video processing. For every source and subset of its destinations, there can be a predefined ALM agent (or a set of agents), which have to lie on path between the source and destination subset. Our ACO algorithm is open to such modification, as we only split tree construction in two phases: one to search for the predefined ALM agents, another to build the rest of the paths down to destinations.

The cuts between cameras occur in reaction to user-induced request. The dynamic reconfiguration of multicast trees is not caused by change of network topology, e. g., failure of a network link. Therefore, the planning algorithm should certainly guarantee preservation of transmission paths from a previous solution where a user requires so. The path preservation would not be only favoured by the objective function, but required by problem constraints. Again, the ACO algorithm can be run with the preserved path as an unmodifiable basis for further growth of the multicast tree.

Although the previous problems are inspired by the application in live broadcasting, their solution may find future use in other collaborative environments. The following problem of planning under strong network uncertainty is common to almost all our future applications. As mentioned in Section II, our algorithms support two layers of network representation: always available knowledge of overlay and optional (possibly partial) knowledge of underlying physical topology. The ACO

algorithm is also designed for cooperation with tools for bandwidth estimation and inference of underlying network topology with uncertain results. Unfortunately, the tools are not ready for deployment. According to our testing, their runtimes are often high, and results on high-capacity networks unreliable beyond admissibility. In order to partially overcome the issues, we are searching for methods of partial topology inference from information on data loss. Most often, our issues with network congestion are linked to incorrect assumptions that overlay links are independent while they actually share capacity of underlay links. In contrast to the commonly proposed methods, which are unobtrusive and aim to load the network as little as possible, we can use the data about actual congestion when one occurs. And once it occurs, we need to respond quickly and establish or restore transmissions, no matter if we are able to infer anything about the underlying network or not. Therefore, we need to develop strategies for the responses in various situations, like initialization of the collaborative environment, arrival of a new participant, or sudden drop of delivery rates without any user-induced cause. The strategies have to include selection of links in multicast trees where rates should be decreased, the decrease rate, and procedures for detection and utilization of free network capacity. We assume that the strategies will not modify the multicast routing algorithms, but rather find suitable parameters of the input network topologies for their execution.

ACKNOWLEDGMENT

We thank the Czech Science Foundation for support by project GAP202/12/0306. Work of Petr Holub is also supported by CESNET Large Infrastructure project LM2010005.

REFERENCES

- [1] C. Drioli, C. Allocchio, and N. Buso, "Networked performances and natural interaction via LOLA: Low latency high quality A/V streaming system," in *Information Technologies for Performing Arts, Media Access, and Entertainment*. Springer, 2013, pp. 240–250.
- [2] K. Zhang, W.-L. Liu, C. Locatis, and M. Ackerman, "Uncompressed high-definition videoconferencing tools for telemedicine and distance learning," *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, vol. 19, no. 8, 2013, pp. 579–584.
- [3] Digital Cinema Initiatives, "Digital Cinema System Specification v. 1.2," http://dcimovies.com/specification/DCI_DCSS_v12_with_errata_2012-1010.pdf, p. 155, 2012, accessed: 26/08/2015.
- [4] International Telecommunication Union, "One-way transmission time: ITU-T recommendation G.114," <http://www.itu.int/rec/T-REC-G.114/en>, p. 20, 2013, accessed: 26/08/2015.
- [5] "UltraGrid," <http://www.ultragrid.cz/en>, accessed: 26/08/2015.
- [6] "CoUniverse," <http://couniverse.sitola.cz>, accessed: 26/08/2015.
- [7] X. Chen, M. Chen, B. Li, Y. Zhao, Y. Wu, and J. Li, "Celerity: towards low-delay multi-party conferencing over arbitrary network topologies," in *Network and Operating System Support for Digital Audio and Video*. ACM, 2011, pp. 123–128.
- [8] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, 2000, pp. 78–88.
- [9] C. Yeo, B. Lee, and M. Er, "A survey of application level multicast techniques," *Computer Communications*, vol. 27, no. 15, 2004, pp. 1547–1568.
- [10] C. A. S. Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers & Operations Research*, vol. 32, no. 8, 2005, pp. 1953–1981.
- [11] C. A. Noronha Jr. and F. A. Tobagi, "Optimum routing of multicast streams," in *Proceedings of the IEEE INFOCOM*, 1994, pp. 865–873.

- [12] Y. Yen, H. Chao, R. Chang, and A. Vasilakos, "Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs," *Mathematical and Computer Modelling*, vol. 53, no. 11-12, 2011, pp. 2238–2250.
- [13] H. Wang, H. Xu, S. Yi, and Z. Shi, "A tree-growth based ant colony algorithm for QoS multicast routing problem," *Expert Systems with Applications*, vol. 38, no. 9, 2011, pp. 11 787–11 795.
- [14] P.-Y. Yin, R.-I. Chang, C.-C. Chao, and Y.-T. Chu, "Niche ant colony optimization with colony guides for QoS multicast routing," *Journal of Network and Computer Applications*, vol. 40, 2014, pp. 61–72.
- [15] L. Sanna Randaccio and L. Atzori, "Group multicast routing problem: A genetic algorithms based approach," *Computer Networks*, vol. 51, 2007, pp. 3989–4004.
- [16] P. Troubil and H. Rudová, "Integer linear programming models for media streams planning," in *International Conference on Applied Operational Research*, 2011, pp. 509–522.
- [17] P. Troubil, H. Rudová, and P. Holub, "Media streams planning with transcoding," in *IEEE Network Computing and Applications (NCA)*, 2013, pp. 41–48.
- [18] A. Malekzadeh and M. H. MacGregor, "Network topology inference from end-to-end unicast measurements," in *Advanced Information Networking and Applications Workshops (WAINA)*, 2013, pp. 1101–1106.
- [19] F. Thouin, M. Coates, and M. Rabbat, "Large scale probabilistic available bandwidth estimation," *Computer Networks*, vol. 55, no. 9, 2011, pp. 2065–2078.
- [20] D. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, 1998, pp. 768–778.
- [21] P. Troubil, H. Rudová, and P. Holub, "Media streams planning with uncertain link capacities," in *IEEE Network Computing and Applications (NCA)*, 2014, pp. 197–204.
- [22] —, "Media streams planning with uncertain link capacities," *Networks: Special issue on metaheuristics in network optimization*, 2014, under review.
- [23] "14th annual global lambda grid workshop," <http://www.glif.is/meetings/2014/>, accessed: 26/08/2015.
- [24] AVIDICUS Project, "Videoconference interpreting," <http://www.videoconference-interpreting.net/>, accessed: 26/08/2015.

An Iterative Method for Enhancing Text Comprehension by Automatic Reading of References

Amal Babour, Fatema Nafa, Javed I. Khan
 Department of Computer Science, Kent State University
 Kent, Ohio, USA
 Email: {ababour,fnafa,javed}@kent.edu

Abstract— Humans read references to increase understanding about a topic. In this research, we investigate an interesting algorithm which tries to mimic the human reading process. Given a free text about a topic, the algorithm iteratively discovers important references to illuminate the less understood part of the text at hand and then analyzes the reference text to add new knowledge paths. The algorithm also consults modern semantic dictionary through the analysis as needed. In this paper, we are going to share an experiment, which uses Wikipedia pages as reference and WordNet as the ontology engine to understand a news article. We display the knowledge gain via this algorithm.

Keywords- WordNet; Wikipedia; Semantic-Graph; Illuminated-Semantic-Graph.

I. INTRODUCTION

Some texts are very complicated and have very specialized concepts that are difficult to comprehend. This happens in many domains, such as science, politics, and language. Text comprehension requires not only the extraction of individual concepts in the text, but also the identification of the relations among the concepts [1]. Text comprehension can be defined as a cognitive process of understanding and comprehending concepts from a text and the relationship among them. The more relations discovered, the better the comprehension of the text. Sometimes the relation between two concepts is explicit, while the relation between two other concepts is implicit and needs to use external reference texts to extract it. An ontology engine is applied to find the relation directly or by detecting a path between the concepts. In some cases, using ontology engine alone does not help. In this case, using additional reference texts can help in finding the hidden relations among the text concepts and adding new knowledge. Most of the recent techniques in text understanding are based on ontology engines, reference texts or both together to find the relations among the text concepts. Although anyone who wants to get a deep comprehension about a specific text back to external references to read about it, the recent techniques do not use the reference texts themselves, but utilize their categories instead. On the other hand, the use of the ontology engines alone gives a short connection. Therefore, these techniques are considered very limited. In the proposed system, we mimic the human process of reading by making a computerized text comprehension to discover the most relevant references that illuminate the relation among the text concepts and add new knowledge about them by an automatic reading of the important reference texts to make the text comprehension easier for the reader.

The technique presented in this research falls in the category of free text based on ontology engine and the reference texts themselves. We demonstrate the way by which the system gives a deep text comprehension, by an iterative process through different reference texts. Thus, the information about the concepts relations is brought from more than one reference. This makes humans have a faster and wider knowledge. The singular value decomposition (SVD) algorithm is applied to extract the most important concepts in the given text and the references. This system can be used to enhance a lot of models, such as Toplelet, which is a deep structure of a specific topic [16].

The rest of the paper is structured as follows. Section II provides an overview of the related work. The main definitions and the system are presented in Section III. In Section IV, we present detailed experimental results demonstrating dramatic improvement in the extracted relations among the concepts after adding the external reference texts. Section V presents the conclusion and the future work.

II. RELATED WORK

There have been interesting researches using an ontology engine such as WordNet to extract relations among concepts. Kang and his associates [1] used Guided Agglomerative Hierarchical Clustering algorithm (GAHC) [2] to identify the semantic relationship among concepts in order to construct concepts of multi-branch hierarchy of a domain specific text corpus automatically. In addition to the “is-a” and “part-of” relations from WordNet, they incorporate a set of lexico-syntactic patterns extended from Hirst’s [3] to identify “is-a” relations from a domain specific corpus, and applied related lexico-syntactic patterns [4] to extract “part-of” relations. Yadav and his associates [5] construct a semantic graph from a text document, applying WordNet to find the direct relation among the concepts automatically to be used for different applications in text mining such as keyword extraction and knowing the nature of the document. Tu and his associates [6] proposed an approach based on text concepts and WordNet to build a multi-way concept hierarchy from a document corpus. They applied multi-way agglomerative clustering algorithm [7] to generate the concept hierarchy automatically. On the other hand, a number of researches applied reference texts such as Wikipedia to extract the concept relations. Taieb and his associates [8] presented the idea of using information content (IC) metric based on Wikipedia Category graph [8] as semantic taxonomic resource for extracting relations between concepts. Lipczak and his associates [9] utilized

Wikipedia category graph to propose a system that spots mentions of entities in a document and link the mentions to corresponding Freebase articles. Han and Mon [10] use the taxonomy of categories in Wikipedia to compute semantic relations using structured knowledge extracted from Wikipedia. They introduce strategy over the network of Wikipedia categories to evaluate semantic relations.

III. OVERVIEW OF THE SYSTEM

The purpose of our system is to create *Illuminated-Semantic-Graph* which represents hidden relations among each pair of concepts in a target text T , where T is the input to the system and the *Illuminated-Semantic-Graph* is the output. In this paper, the term concept refers to a particular sense of noun.

The *Illuminated-Semantic-Graph* is a directed graph $G=(V,E)$ where V is a set of concepts and E is a set of edges. Each concept v_n can have one or more senses $(S_{n1}, S_{n2}, \dots, S_{nk})$ where n is the concept number and k is the sense number. Each edge connects two concepts via a specific sense of each concept. A label represents the type of relation between any two concepts is shown over each edge. The concept is either in T , ontology engine, or reference text, while the edge between any two concepts represents the relation between the concepts via the following relation types (synonym (S), hyponym (H), meronym (M), and instance (I)). We do not mention the hypernym and holonym types because they are the inverse relations of hyponym and meronym respectively. Since we are dealing with noun concepts, we focused only on the mentioned relations. We define a *Knowledge Path* $P_{i,j}$ as a path which preserves the sense in G . It is a sequence of edges that connects a concept C_i with a concept C_j .

Figure 1 shows two examples of geometric paths: $C_1 - \{S_{11}-I-S_{21}\} - C_2 - \{S_{21}-S-S_{31}\} - C_3 - \{S_{31} -I-S_{51}\} - C_5$ and $C_1 - \{S_{11}-I-S_{21}\} - C_2 - \{S_{21}-S-S_{31}\} - C_3 - \{S_{32}-H-S_{41}\} - C_4 - \{S_{41}-H-S_{64}\}-C_6$, where $C_1, C_2 \dots$ etc. refer to the concept number. S_{11} refers to the first sense of the first concept, S_{21} the first sense of the second concept, etc. The Knowledge paths can be extracted from the geometric-paths. For example, $C_1 - \{S_{11}-I-S_{21}\} - C_2 - \{S_{21}-S-S_{31}\} - C_3 - \{S_{31} -I-S_{51}\} - C_5$ and $C_3 - \{S_{32}-H-S_{41}\} - C_4 - \{S_{41}-H-S_{64}\}-C_6$ are considered knowledge paths because the incoming and the outgoing senses for each concept is preserved. However, not all the knowledge paths are correct. Some of them are classified as noise paths. For example, C_1 is an instance of C_2 , C_5 is instance of C_3 , and C_2 and C_3 are synonyms, so C_1 and C_3 seem to be equivalent, while they are not. So $C_1 - \{S_{11}-I-S_{21}\} - C_2 - \{S_{21}-S-S_{31}\} - C_3 - \{S_{31} -I-S_{51}\} - C_5$ is considered a noise path.

The idea of the system is represented in six steps:

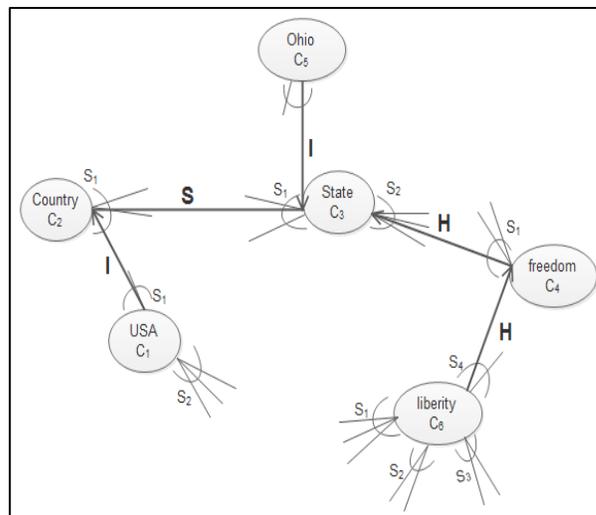


Figure. 1. Geometric-paths examples.

1. Extracting the most important concepts in T to create the *Target-Text-graph*.
2. Using an ontology engine to discover the relation between each pair of concepts extracted in step 1 by adding knowledge paths between each pair if found. A number of ontology engine concepts are added to the graph. The graph in this version is called *Semantic-graph*.
3. Extracting the most important concepts from a reference text i , which is a text related to T .
4. Repeat step 2 on the important concepts extracted in step 3. The graph in this version is called *Reference-Semantic-Graph*.
5. Add more knowledge paths from the *Reference-Semantic-Graph* created in step 4 to the *Semantic-Graph* created in step 2, while either of the added knowledge paths ends is a T concept and the other end is a new concept from the reference text concepts. A number of ontology engine and reference text concepts are added to the graph. The graph in this version is called *Illuminated-Semantic-Graph*.
6. Repeat 3-5 steps for the reference text $i++$.

Figure 2 explains the steps. Figure 3 illustrates an example that shows how adding more knowledge paths from the *Reference-Semantic-Graph* to the *Semantic-Graph* gives more text comprehension. Some of the knowledge paths give concept-illuminations by adding new knowledge to T concepts, while other knowledge paths give relation-illuminations by showing additional hidden relations between the knowledge paths ends of T concepts.

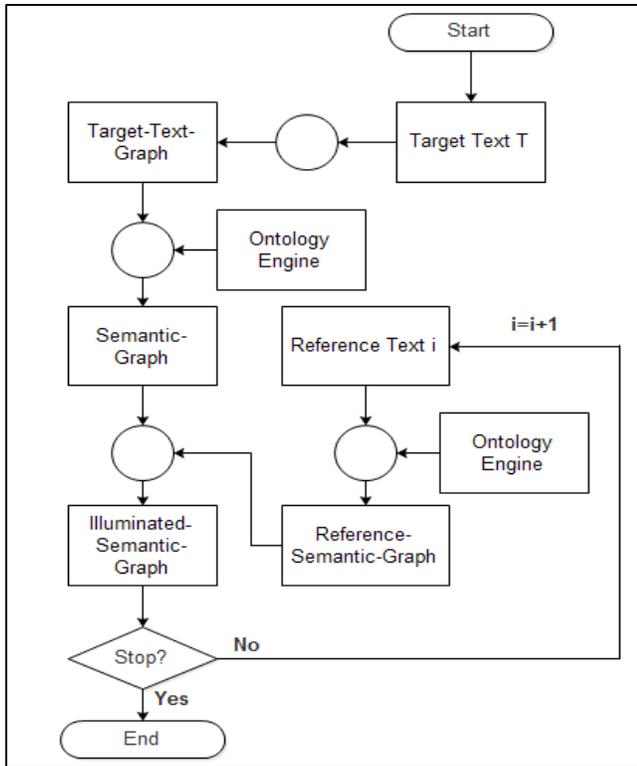


Figure 2. Workflow to generate the Illuminated-Semantic-Graph.

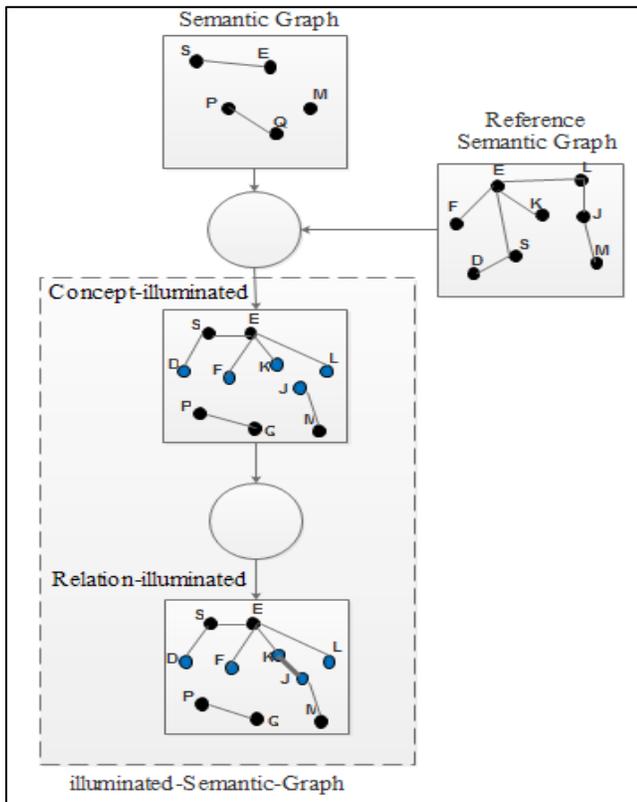


Figure 3. Example of Concept illumination and Relation Illumination.

The overall system uses two core techniques (i) **concepts extraction** and (ii) **knowledge paths detection**. Both techniques are applied on two phases **Target Text phase** and **Reference Text phase**, where the input of the Target Text phase is T and the output is the *Semantic-graph*. The *Target Text graph* is generated after applying the first technique in this phase, where the SVD algorithm is used to generate n^c which is the most important concepts in T . The input of the Reference Text phase is a text R_T from any reference such as Wikipedia. This text is selected based on n_c concepts where n_c is a list of T concepts that have knowledge paths among them in the *Semantic-Graph*, $n_c \subseteq n^c$. The output of the Reference Text phase is a *Reference-Semantic-Graph*. A number of knowledge paths generated from *Reference-Semantic-Graph* are added to the *Semantic-Graph* to generate the *Illuminated-Semantic-Graph* where either of the added knowledge paths ends is a T concept and the other end is a new concept from R_T . The more added knowledge paths, result in more concepts and relations illumination for T concepts.

Figure 4 explains the process in details. T and R_T go through various text preprocessing. They are pruned by removing stop words and stemming words using Porter stemmer algorithm.

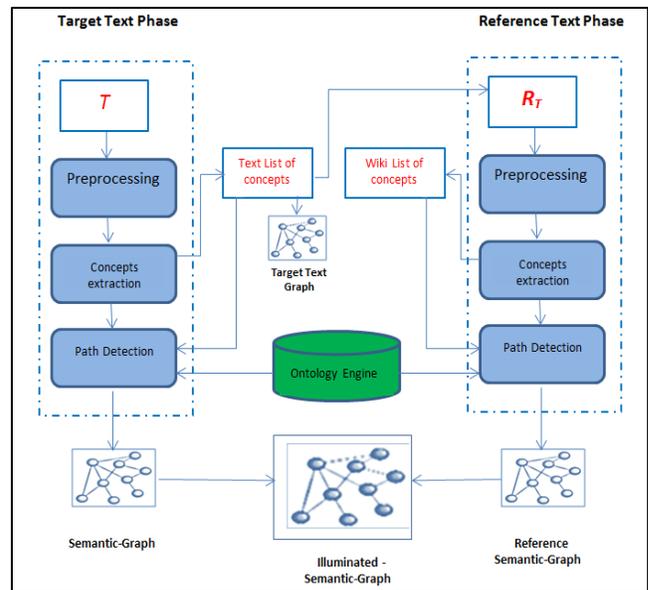


Figure 4. Overview of the System.

The **concepts extraction** technique generates a list of the most important concepts in T based on SVD method. **Knowledge path detection** technique finds the knowledge paths between each pair of concepts if found using an ontology engine.

A. *Concepts extraction:*

This technique uses SVD [11] to extract the most important concepts in the text using the following equation:

$$A=USV^T$$

Where A is n -by- n matrix and n represents all concepts/nouns in the text. Each cell in A represents the number of appearance of pair of nouns in each sentence in the text. U is an orthogonal matrix, S is a diagonal matrix, and V is the transpose of an orthogonal matrix. The SVD is used to reduce the high dimensionality of matrix U , emphasize the important concepts in T , and eliminate any noise. The first dimension of U is ordered in descending order to get the n of the most important concepts in the T . A number of pairs between concepts are generated from n . Table I gives an example of concept pairs.

TABLE I. CONCEPTS PAIRS DATA STRUCTURE

i	Concept pairs
1	(judge, charge)
2	(Egypt, Israel)
3	(case, family)
..	..

B. Knowledge paths detection:

Function *discover-knowledge-paths* in Figure 5 searches for knowledge paths of length less than or equal k connecting each pair of concepts/nodes if found using an ontology engine. Its input is *Sword* and *Eword*, where *Sword* and *Eword* are respectively the first and last nodes in the path. R is a dictionary of all relations between concepts in the ontology engine, and *relationalGraph* is a dictionary used to hold concepts that have any type of relations from R with the last node in the current path. The search for a knowledge path has been implemented as a breadth-first search (BFS). The function searches in all senses of *Sword*. For each sense, it searches for knowledge paths from *Sword* node to *Eword* node by searching the neighbors of the *Sword* that have any type of relations from R and have the same sense of *Sword*. Then, it searches the neighbors of the neighbors, and so on until it reaches *Eword*. The function returns the knowledge paths of length less than or equal k connecting *Sword* and *Eword*. If the knowledge path does not exist, the function returns 'Not Found'. The function does not return the shortest path between the pair of concepts as the knowledge path because it could be a path of multi senses concepts.

The function uses two queue data structures. One is *NodeQ* and the other one is *PathQ*. The *NodeQ* saves the current path in the BFS that has the node to explore the next. The *PathQ* holds the created paths until now. The function starts with *Sword* as the current path. The while loop iterates through the paths in *PathQ* searching for a knowledge path connects *Sword* and *Eword*. In each loop iteration, it dequeues the first path in *PathQ* and signed it in *NodeQ*. Then, it checks if the last node in *NodeQ* matches *Eword*. If so, the function saves the knowledge path in *Kpaths*. If not, it checks if the length of the *NodeQ* is less than k , if so, for the sense of the last node in *NodeQ*, the function gets all the

concepts that have one of the relation types from R , with the last node in *NodeQ* and add them to *relationalGraph*. A number of paths are created between each concept in the *relationalGraph* and the current path. The new created paths are saved in *PathQ*. If all paths in *PathQ* are checked and *Kpaths* does not exist, the function returns 'Not found'. Let us consider *Sword*= 'Egypt and *Eword* =Israel, and $k < 4$. Figure 6 illustrates the process of discovering the knowledge-path between the two concepts 'Egypt' and 'Israel'.

Function discover knowledge-paths

```

1. def discover-knowledge-path(relationalGraph, Sword,
   Eword,k,R):
2.   PathQ = []
3.   Kpaths=[]
4.   # push the first path into the PathQ
5.   PathQ.append([Sword])
6.   for sen in Sword.sense():
7.     while PathQ:
8.       # get the first path from the PathQ
9.       NodeQ = PathQ.pop(0)
10.      # get the last node from the NodeQ
11.      node = NodeQ[-1]
12.      # path found
13.      if node == Eword:
14.        Kpaths.append(NodeQ)
15.        return Kpaths
16.      else:
17.        if len(NodeQ) < k:
18.          s=list()
19.          for key, value in R.iteritems():
20.            n=value
21.            # get all concepts have relations from R with node and
            # have the same sense of node
22.            x=n(node, sen, key)
23.            s=s+x
24.            relationalGraph[node]=s
25.          end for
26.        end if
27.      end if
28.      # enumerate all adjacent nodes, construct a new path and
      # push it into the queue
29.      for adjacent in relationalGraph.get(node,[ ]):
30.        new_path = list(NodeQ)
31.        new_path.append(adjacent)
32.        if len(new_path) < k:
33.          PathQ.append(new_path)
34.        else:
35.          break
36.      end if
37.    end for
38.  end while
39. end for
40. if !(Kpaths)
41.   return 'Not found'
```

Figure 5. Discovering Knowledge-paths function.

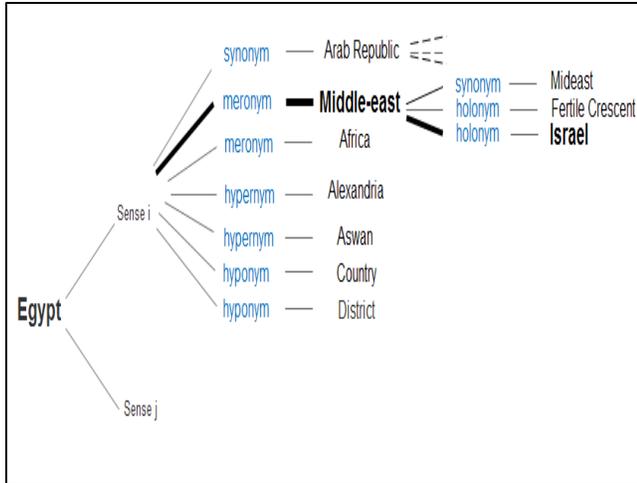


Figure 6. Knowledge Path example.

The following path is the knowledge path returned by the algorithm. **Egypt** (meronym) **Middle-east** (holonym) **Israel**.

IV. EXPERIMENTAL EVALUATION

This section describes an evaluation of our methodology based on the obtained results. We chose an interesting article from the media as the target text. Its title is “*Egypt: Ex-ruler Hosni Mubarak, accused in deaths of hundreds, cleared of charges*” [12]. The article was released on November 30, 2014 and has 45 sentences and 152 concepts where the concepts are only nouns. The used ontology engine is WordNet, version 1.7. We select the first 42 concepts from 152 concepts as the most important text concepts n^t . The used reference text is Wikipedia. The reference text was about ‘Hosni Mubarak’ [13]. It was selected based on the first concept in n_c . Table II shows the number of nodes in the three graphs Target-Text Graph (TG), Semantic-Graph (SG), and illuminated-Semantic-Graph (ISG).

TABLE II NUMBER OF NODES IN THE THREE GRAPHS

	TG	SG	ISG
T	42	42	42
OE	0	7	13
R_T	0	0	6
Total	42	49	61

Figure 7 shows the relations types of the three graphs with the ontology engine (OE) relations (synonym, hyponym and meronym). We see that the ISG set has the largest number of relations followed by SG set, while the TG has no relations. SG set shows that there are zero synonym, 18 hyponym, and 2 meronym relations which mean that using OE helps in discovering some of the hidden relations among T concepts. In ISG, three new synonym relations appear among T concepts, the number of hyponym relations reached to 29, while the number of meronym jumped to 4. Therefore, using OE and RT concepts discover more hidden relations among T concepts.

Humans discovered that not all the gained knowledge paths are correct and some of them are considered noise paths. Figure 8 shows a breakdown of the obtained geometric-paths in the three graphs. Evidently, the number of the correct knowledge paths in SG and ISG has risen gradually, while the number of noise paths is stable. These results confirm that our system is suitable for discovering the knowledge paths among T concepts.

Figure 9 shows an overview of the resulting of five well-known graph parameters (disconnected-components, giant component, average non-giant component, average degree, and diameter) on the three graphs. In TG, the 42 concepts are disconnected-components because there is no knowledge paths among them thus the average size of non-giant component is 1, while all the rest of graph parameters are zero. In SG, the number of disconnected-components is sharply decreased to 23. This refers to that the use of OE helps in discovering some relations among some of the T concepts by adding knowledge paths among them. As seen, the size of the giant component jumped to 7 and the average size of non-giant component [14] is 1.5 which means that 14.3% of the concepts in this graph including T and OE concepts are reachable from one another. The average degree [14] increased to 0.82 that means there is a high number of relations appear among the T concepts, while the diameter is 6, that refers to the longest knowledge path between any two T concepts. For ISG, all the results of the graph parameters are improved as the following: The number of disconnected-components is decreased to 18, which means that adding the OE and the R_T concepts adds more relations and knowledge among the T concepts. The size of giant component including T, OE, and R_T concepts increased to 12, which is almost 19.7% of the concepts in this graph and the average size of non-giant component is 1.9 that refers to the more reachability among the concepts. Thus, more of the hidden relations among the T concepts appeared and more knowledge added. Average degree increased to 1.11, which points to the much high connectivity among the T concepts, whereas the diameter reached to 9.

The SG and the ISG of our experiment are shown in Figure 10 and Figure 11. In Figure 10 the white nodes show the T concepts, the dark gray nodes represent the OE concepts, and edges among them represent the paths. The solid black edges illustrate the correct knowledge paths and the dashed edges depict the noise paths. Additionally in Figure 11, the light gray nodes are the new concepts added from RT and the solid gray edges show the knowledge paths among T and RT concepts. The double circle nodes are the shared nodes between T and RT. In Figure 10 and Figure 11, H refers to the Hyponym relation, M refers to the Meronym relation, and S represents the Synonym relation.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a computerized human text comprehension technique by automatic reading reference texts, that gives a deep comprehension about a specific text

and demonstrate how it can improve the text comprehension. In the process, we used the SVD algorithm to extract the most important concepts in the target text and the reference text to generate the illuminated-Semantic-Graph using ontology engine. The results show that the system improves the text comprehension by adding relations and new knowledge among the concepts. Although the system succeeded in finding the knowledge paths, it has a limitation with filtering the correct knowledge paths from the noise paths. Alternatively, we perform the filtration manually. We intend to handle this issue automatically in the future work.

REFERENCES

[1] Y. Kang, L. Zhou, and D. Zhang "An integrated method for hierarchy construction of domain-specific terms" IEEE/ACIS 13th International Conference on. IEEE, 2014, pp.485-490.

[2] P. Cimiano, and S. Staab "Learning concept hierarchies from text with a guided hierarchical clustering algorithm", ICML 2005 workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany, 2005, pp. 6-16.

[3] G. Hirst, and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms" In Fellbaum, C., ed., WordNet: An electronic lexical database, 1998, pp. 305-332.

[4] R. Girju, A. Badulescu, and D. Moldovan, "Automatic Discovery of Part- Whole Relations. Computational Linguistics", Vol. 32, 2006, pp.83-135.

[5] C. S. Yadav, A. Sharan, and M. L. Joshi, "Semantic graph based approach for text mining." Issues and Challenges in Intelligent Computing Techniques (ICICT), International Conference on. IEEE, 2014, pp. 596-601.

[6] D. Tu, L. Chen, and G. Chen. "Automatic multi-way domain concept hierarchy construction from customer reviews." Neurocomputing 147, 2015, pp.472-484.

[7] D. Tu, L. Chen, and G. Chen, "WordNet based multi-way concept hierarchy construction from text corpus", Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 1647-1648.

[8] M. A. H. Taieb, M. Ben Aouicha, M. Tamer, and A. Ben Hamadou "Wikipedia category graph and new intrinsic information content metric for word semantic relatedness measuring" Data and Knowledge Engineering. Springer Berlin Heidelberg, 2012, pp. 128-140.

[9] M. Lipczak, A. Koushkestani, and E. Milios "Tulip: lightweight entity recognition and disambiguation using wikipedia-based topic centroids" Proceedings of the first international workshop on Entity recognition & disambiguation. ACM, , 2014, pp. 31-36.

[10] M. S. Han, and E. E. Mon. "Computing Semantic Relatedness using Wikipedia Taxonomy by Spreading Activation" International Conference on Advances in Engineering and Technology (ICAET), Singapore, 2014, pp. 49-53.

[11] G. H. Golub, and C. Reinsch. "Singular value decomposition and least squares solutions" Numerische Mathematik 14.5, 1970, pp. 403-420.

[12] J. Hanna, S. Sirgany and H. Yan. (2015, July 25). Egypt: Ex-ruler Hosni Mubarak, accused in deaths of hundreds, cleared of charges. CNN. Reterived from: <http://www.cnn.com/2014/11/29/world/meast/egypt-mubarak-trial>.

[13] Hosni Mubarak (2015). From Wikipedia. Retrived July 25, 2015, from: http://en.wikipedia.org/wiki/Hosni_Mubarak.

[14] M. S. Haldas "Segmentation and Integration in Text Comprehension: A Model of Concept Network Growth" Diss. Kent State University, 2012.

[15] L. Wilkinson, A. Anand, and R. Grossman. "Graph-Theoretic Scagnostics", INFOVIS. Vol. 5, 2005, pp.21.

[16] A. Babour and J. I. Khan. "Tweet Sentiment Analytics with Context Sensitive Tone-Word Lexicon" Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Vol. 01. IEEE Computer Society, 2014, pp. 392-399.

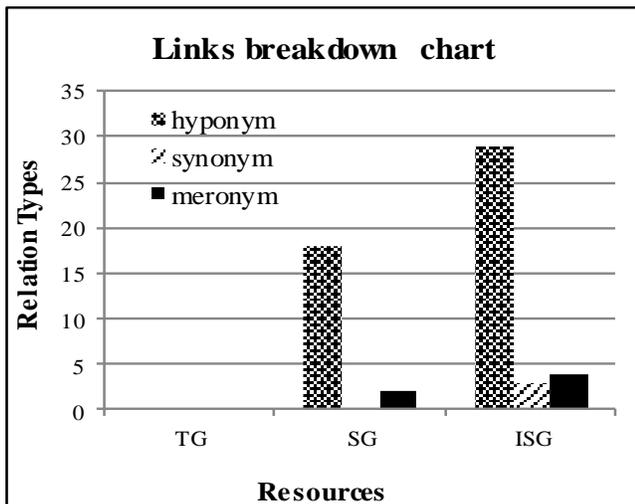


Figure 7. OE relation analysis for the three graphs.

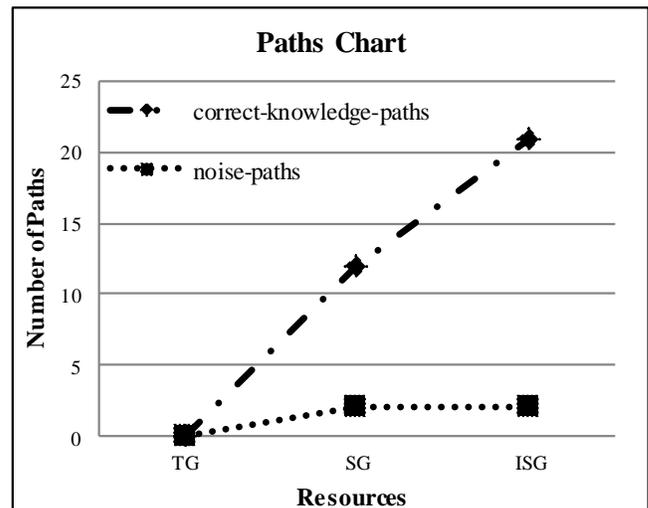


Figure 8. Paths analysis for the three graphs.

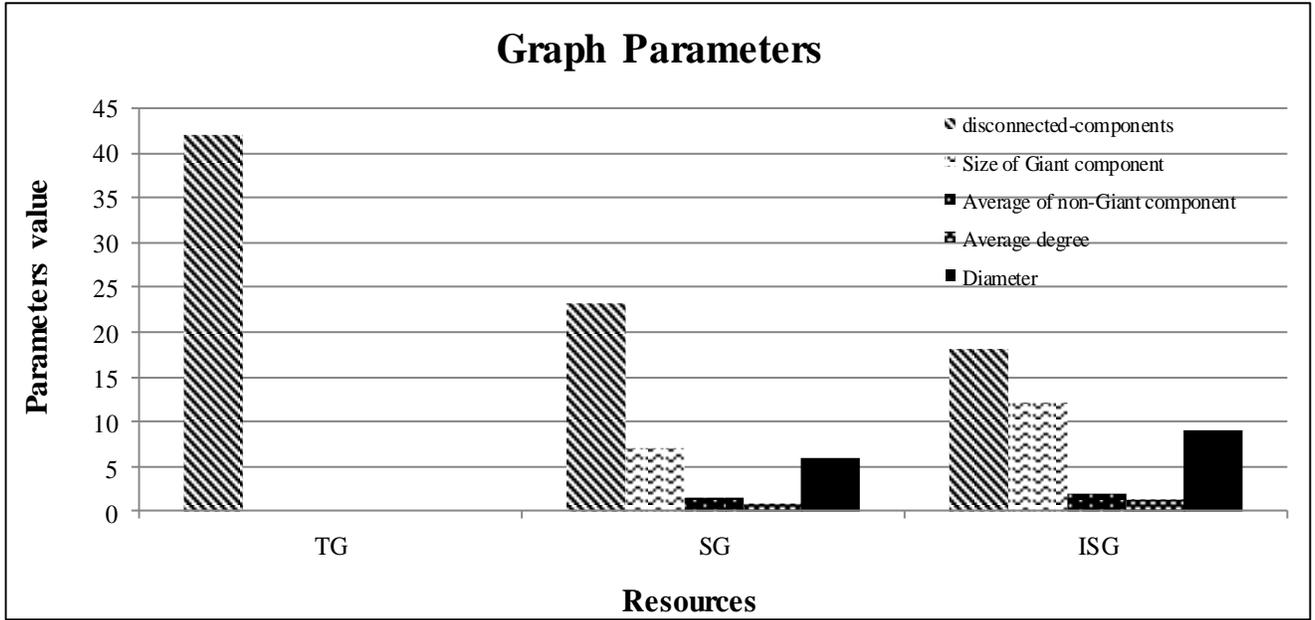


Figure 9. Measuring graph parameters for the three graphs.

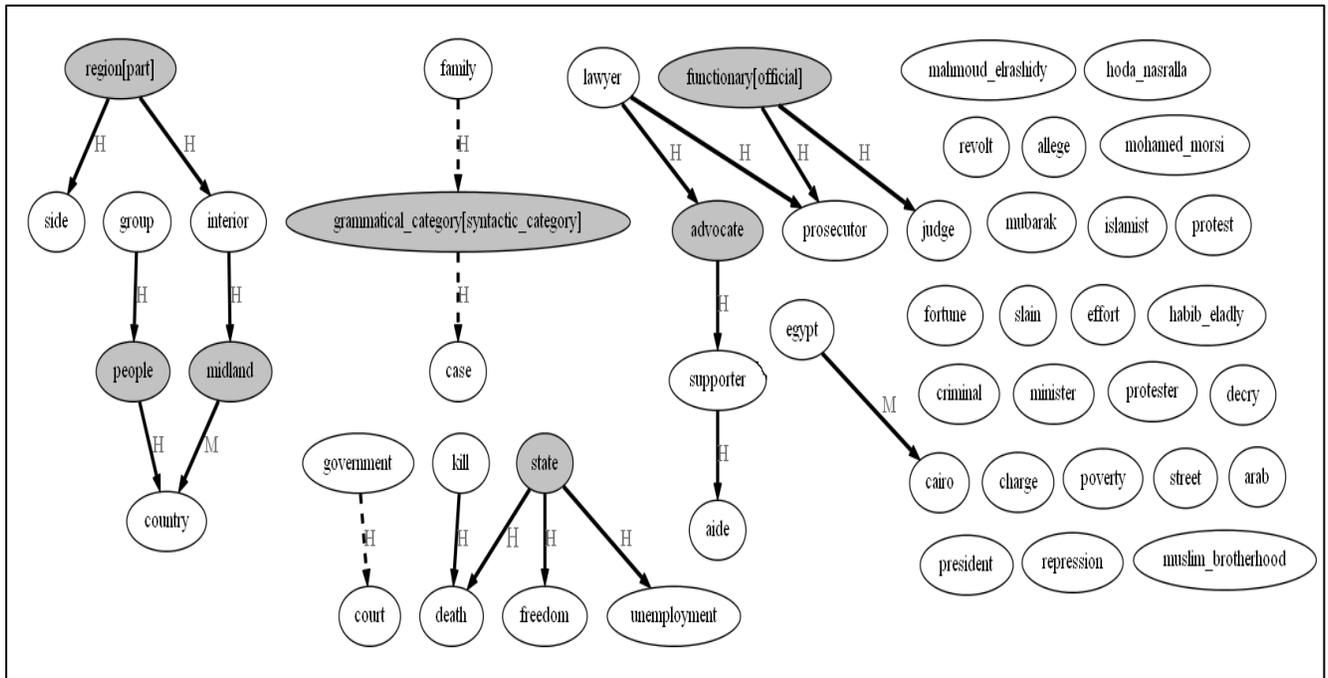


Figure 10. SG for the Target text [12].

Crowdsourcing-Based Multi-Layer Automated Ontology Matching

An approach and Case Study

Alexander Smirnov, Nikolay Shilov, Nikolay Teslya
 Laboratory of Computer Aided Integrated Systems
 SPIIRAS
 St.Petersburg, Russia
 International Laboratory «Intelligent Technologies for
 Socio-Cyberphysical Systems»
 ITMO University
 St.Petersburg, Russia
 e-mail: {smir, alexey, nick, teslya}@iias.spb.su

Alexey Kashevnik
 Laboratory of Computer Aided Integrated Systems
 SPIIRAS
 St.Petersburg, Russia
 Department of Computer Science
 Petrozavodsk State University (PetrSU)
 Petrozavodsk, Russia
 alexey@iias.spb.su

Abstract—This paper presents an approach and a case study for a multi-layer automated ontology matching based on the crowdsourcing technique. The main idea of our approach is that ontology matching is implemented automatically at first. In the case of non-adequate matching, the crowdsourcing technique is invoked, that involves crowd participants into the matching process. As a case study, the scenario of robot interaction is considered. The developed ontology matching approach allows providing for semantic interoperability between the robots for joint tasks solving.

Keywords-ontology; ontology-matching; crowdsourcing; robots; interoperability.

I. INTRODUCTION

Ontology matching plays an important role in the development of ontology-based information systems. If a system consists of several interacting components and each component is developed by different manufactures based on different ontologies, we need to implement a matching of ontology entities between these components in order to provide semantic interoperability between them.

To implement such systems, the smart space technology can be used, which allows information sharing between different services of the system. This technology [1][2] aims at the seamless integration of different devices by developing ubiquitous computing environments, where different services can share information with each other, make different computations and interact for joint task solving. The open source Smart-M3 platform [3] has been used for the organization of the smart space infrastructure of the robots self-organization case study presented in the paper. The use of Smart-M3 platform enables significant simplification of further system development, including new information sources and services, making the system highly scalable. The key idea of this platform is that the formed smart space is device-, domain-, and vendor-independent. Smart-M3 assumes that devices and software entities can publish their embedded information for other devices and software entities through simple shared information brokers. The Smart-M3 platform consists of two main parts: information agents and

kernel [4]. The kernel consists of two elements: Semantic Information Broker (SIB) and information storage. Information agents are software entities, installed on mobile devices of smart space users and other devices hosting smart space services. These agents interact with SIB through the Smart Space Access Protocol (SSAP). The SIB is the access point for receiving the information to be stored, or retrieving the stored information. All this information is stored in the information storage as a graph that conforms with the rules of the Resource Description Framework (RDF). In accordance with these rules, all information is described by triples "Subject - Predicate - Object".

The paper presents a multi-level automated ontology matching approach based on the crowdsourcing technique. Matching of ontology elements is implemented via automatic procedures and then enhanced manually if needed.

In the presented case study, the proposed algorithm is used to provide for interoperability support for robots solving a joint task.

The rest of the paper is structured as follows. The state-of-the-art of ontology matching systems is presented in Section II. Section III describes the proposed approach to ontology matching. Section IV presents the case study implemented using the proposed approach. The results are summarized in the Conclusion section.

II. STATE-OF-THE-ART

In order to analyze the existing ontology matching techniques, an extensive state-of-the-art review has been done, which covered systems/approaches/projects related to ontology matching. Among them the following ones are worth mentioning: GLUE System [5][6], Falcon-AO [7], MLMA [8], Hovy [9], SKAT [10], ONION [11], Prompt [12], H-Match [13], CTX-MATCH [14], SMART [15], Cupid [16], COMA [17], Similarity Flooding Algorithm [18], AgreementMaker [19], Pattern Based Approach [20], MinSMATCH [21], OntoView [22], Chimaera [16], VITRUVIUS [23][24], SAMBO [25], Falcon [26], DSSim [27], RiMOM [28], ASMOV [29], Anchor-Flood [30]. The following systems are

more interesting and, for this reason, we describe them in detail below.

The VITRUVIUS platform integrates multiple sensors and handles different sensor configurations, allowing applications to be installed dynamically and run concurrently. A benefit of the platform is that it provides capabilities for reuse and evolution of existing sensor networks and applications, and for extensions by adding new sensors and applications. The platform uses an ontology-based approach to achieve semantic interoperability between different components. The authors understand the ontology as a vehicle that unifies the data originating from different system components into a universal understanding [24]. Ontology mappers are used to translate the local ontology (local syntactic and structural representations) into the application ontology and vice versa. From the development perspective, by using the ontology mapper, a new component (e.g., a sensor driver) can be integrated into the platform easily. This does not require re-implementing the component; only the ontology mapper requires an updated specification of how local terminologies and structures that refer to the new sensor can be represented in terms of the application ontology. The mapping between the application ontology and the local ontology is performed by the mapper component, which also provides interfaces for the data communication and control (e.g., sensor configuration). The mapper is implemented as an Android service, which can quickly be implemented using a generic development pattern.

SAMBO is a system for matching and merging biomedical ontologies. It handles ontologies specified in OWL language and outputs *1:1* alignments between concepts and relations. The system uses various similarity-based matchers, including.

- Terminological: n-gram, edit distance, comparison of the lists of words of which the terms are composed. The results of these matchers are combined via a weighted sum with pre-defined weights.
- Structural, through an iterative algorithm that checks if two concepts occur in similar positions with respect to is-a or part-of hierarchies relative to already matched concepts, with the intuition that the concepts under consideration are likely to be similar as well.
- Background knowledge-based, using (i) a relationship between the matched entities in Unified Medical Language System (UMLS) and (ii) a corpus of knowledge collected from the published literature exploited through a naive Bayes classifier.

The results produced by these matchers are combined based on user-defined weights. Then, filtering based on thresholds is applied to come up with an alignment suggestion, which is further displayed to the user for feedback (approval, rejection or modification). Once matching has been accomplished, the system can merge the matched ontologies, compute the consequences, check the newly created ontology for consistency, etc.

Falcon is an automatic divide-and-conquer approach to ontology matching. It handles ontologies in RDFS and OWL. It has been designed with the goal of dealing with large ontologies. The approach operates in three phases.

- Partitioning ontologies

- Matching blocks.
- Discovering alignments.

The first phase starts with a structure-based partitioning to separate entities (classes and properties) of each ontology into a set of small clusters. Partitioning is based on structural proximities between classes and properties, e.g., how closely are the classes in the hierarchies of rdfs: subClassOf relations and on an extension of the Rock agglomerative clustering algorithm [31]. Then, it constructs blocks out of these clusters. In the second phase, the blocks from distinct ontologies are matched based on anchors (pairs of entities matched in advance), i.e., the more anchors are found between two blocks, the more similar the blocks are. In turn, the anchors are discovered by matching entities with the help of the I-SUB string comparison technique [32].

DSSim is an agent-based ontology matching framework. The system handles large-scale ontologies in OWL and SKOS (Simple Knowledge Organization System) and computes *1:1* alignments with equivalence and subsumption relations between concepts and properties. It uses the Dempster-Shafer [33] theory in the context of query answering. Specifically, each agent builds a belief for the correctness of a particular correspondence hypothesis. Then, these beliefs are combined into a single more coherent view in order to improve correspondence quality. The ontologies are initially partitioned into fragments. Each concept or property of a first ontology fragment is viewed as a query, which is expanded based on hypernyms from WordNet [34], viewed as background knowledge. These hypernyms are used as variables in the hypothesis to enhance the beliefs. The expanded concepts and properties are matched syntactically to the similar concepts and properties of the second ontology in order to identify a relevant graph fragment of the second ontology. Then, the query graph of the first ontology is matched against the relevant graph fragment of the second ontology. For that purpose, various terminological similarity measures are used, such as Monger-Elkan and Jaccard distances, which are combined using Dempster's rule. Similarities are viewed as different experts in the evidence theory and are used to assess quantitative similarity values (converted into belief mass functions) that populate the similarity matrices. The resulting correspondences are selected based on the highest belief function over the combined evidences. Eventual conflicts among beliefs are resolved by using a fuzzy voting approach equipped with four ad hoc if-then rules. The system does not have a dedicated user interface but uses that of the AQUA (An Ontology-Driven Question Answering System) able to handle natural language queries.

RiMOM is a dynamic multi-strategy ontology matching framework. It focuses on combining multiple matching strategies, through risk minimization of Bayesian decision and quantitatively estimates the similarity characteristics for each matching task. These characteristics are used for dynamically selecting and combining the multiple matching methods. Two basic matching methods are employed.

- Linguistic similarity (edit distance over entity labels, vector distance among comments and instances of entities).

- Structural similarity (a variation of Similarity Flooding [18] implemented as three similarity propagation strategies: concept-to-concept, property-to-property and concept-to-property).

In turn, the strategy selection uses label and structure similarity factors, obtained as a preprocessing of the ontologies to be matched, in order to determine what information should be employed in the matching process. Specifically, the strategy selection dynamically regulates the concrete feature selection for linguistic matching, the combination of weights for similarity combination, and the choice of the concrete similarity propagation strategy. After similarity propagation, the matching process concludes with alignment refinement and extraction of the final result.

Automatic Semantic Matching of Ontologies with Verification (ASMOV) is an automatic approach for ontology matching that targets information integration for bioinformatics. Overall, the approach can be summarized in two steps.

- Similarity calculation.
- Semantic verification.

It takes two OWL ontologies and an optional alignment as the input and returns an $n:m$ alignment between ontology entities (classes and properties) as the output. In the first step, it uses lexical (string equality, a variation of Levenshtein distance), structural (weighted sum of the domain and range similarities) and extensional matchers to iteratively compute similarity measures between two ontologies, which are then aggregated into a single one as a weighted average. It also uses several sources of general and domain specific background knowledge, such as WordNet and UMLS, to provide more evidence for similarity computation. Then, it derives an alignment and checks it for inconsistency. Consistency checking is pattern based, i.e., that instead of using a complete solver, the system recognizes sets of correspondences that are proved to lead to an inconsistency. The semantic verification process examines five types of patterns, e.g., disjoint-subsumption contradiction, subsumption incompleteness. This matching process is repeated with the obtained alignment as input until no new correspondences are found.

AgreementMaker is a system composed of a wide range of automatic matchers, an extensible and modular architecture, a multi-purpose user interface, a set of evaluation strategies, and various manual, e.g., visual comparison, and semi-automatic features, e.g., user feedback. It has been designed to handle largescale ontologies based on the requirements coming from various domains, such as the geospatial and biomedical domains. The system handles ontologies in XML, RDFS, OWL and outputs $1:1$, $1:m$, $n:1$, $n:m$ alignments. In general, the matching process is organized into two modules: similarity computation and alignment selection. The system combines matchers using three layers.

- The matchers of the first layer compare concept features, such as labels, comments, instances, which are represented as TF_IDF (TF — Term Frequency, IDF — Inverse Document Frequency) vectors used with a cosine similarity metric. Other string-based measures, e.g., “edit distance” or “substrings”.

- The second layer uses structural ontology properties and includes two matchers called descendants similarity inheritance (if two nodes are matched with high similarity, then the similarity between the descendants of those nodes should increase) and siblings similarity contribution (which uses the relationships between sibling concepts).

At the third layer, a linear weighted combination is computed over the results coming from the first two layers, whose results are further pruned based on thresholds and desired output cardinalities of the correspondences. The system has a sophisticated user interface deeply integrated with the evaluation of ontology alignment quality, being an integral part of the matching process, thus empowering users with more control over it.

III. PROPOSED ONTOLOGY MATCHING APPROACH

A. General Description

The below proposed approach allows matching of two ontologies for the interoperability purposes of appropriate services and is based on the ontology matching model illustrated in Figure 1. The approach takes into account that the matching procedure has to be done in three steps. The first two steps are performed “on-the-fly” and the third step is optional; it is performed on-demand if the matching results of first two steps are not satisfactory. When a service joins a smart space, it performs a matching of its own ontology with the smart space ontology. If all classes that characterize service capabilities and requirements are matched with smart space ontology, the third step is skipped. The proposed approach also includes a graph-based distance improvement model that allows to propagate similarity from matched elements to elements related to them.

The approach consists of the following steps:

1. Compare all elements between two ontologies and fill the matrix M using *similarity-based model*. The matrix M is of size m to n , where m is the number of elements in the first ontology and n is the number of elements in the second ontology. Each element of this matrix contains the degree of similarity between the string terms of two ontology elements using the fuzzy string comparison method.

2. Calculate *semantic distances*, using background knowledge, e.g., WordNet or Wiktionary [35] and fill the matrix M' . The matrix M' is of size m to n , where m is the number of elements in the first ontology and n is the number of elements in the second ontology. Each element of this matrix represents the degree of similarity between two ontology elements.

3. Update values in matrix M , where each new value of elements of M is the maximum value of (M, M') .

4. Improve distance values in the matrix M using the *graph-based distance improvement model*.

5. (Optional) Use the crowdsourcing technique for comparison of ontologies.

As a result, the matrix M contains the degrees of similarity between ontology elements of the two services. This allows determining correspondences between elements by selecting higher than the threshold value degrees of similarities.

B. Similarity-Based Model for Matching Ontology

The similarity-based model for the ontology matching is presented in Figure 2. It contains a stemming procedure to normalize words, improved fuzzy string comparison procedure, and normalization procedure.

To improve the matching quality, the application of the stemming procedure is proposed. This operation makes it possible to identify ontology elements even if they are written in different forms. The following conversions can be done: “looking” → “look”, “device” → “devic”, “vertical” → “vertic”, and “horizontal” → “horizont”. This procedure is uniquely tuned for each supported language.

The basis of the string comparison algorithm is the well-known conventional algorithm that calculates occurrence of substrings from one string in the other string. However, this algorithm does not take into account the length of the second string. As a result, it was decided to introduce the comparison based on the above algorithm twice: $FC_1 = \text{FuzzyCompare}(\text{Element}_1, \text{Element}_2)$ and $FC_2 = \text{FuzzyCompare}(\text{Element}_2, \text{Element}_1)$. After that we calculate the result as an aggregation of the above results in accordance with the following formula:

$$Re' = n * FC_1 + (1 - n) * FC_2, \text{ where } n \text{ is a weight, } n \in [0; 1].$$

$n = 0.5$ sets the same weight to the both strings, $n = 0$ searches only Request within Class, and $n = 1$ searches only Class within Request. It is proposed to set $n = 0.5$. Since the similarity metrics are obtained by different techniques they have to be normalized.

C. Model of searching semantic distances of ontology elements

To measure semantic distances between ontology elements we use the machine readable dictionary extracted by direct access to Wiktionary and WordNet.

This dictionary includes:

- 1) a set of words defined in dictionary along with,
- 2) definitions given for each word,
- 3) a set of synonyms for each word, and
- 4) a set of associated words for each word.

Words associated with a word are considered as hyperlinked words occurring in the Dictionary definition given for this word.

The nodes of ontology are linked to nodes representing their synonyms and associated words as this is given in the machine-readable dictionary. The links between the nodes are labeled by the distance of relations specified between the concepts represented by these nodes in the machine-readable dictionary. Weight w of a relation specified between two ontology elements t_i and t_j is assigned as:

$$w = \begin{cases} 0,5 & -t_i, t_j \text{ are synonyms} \\ 0,3 & -t_i, t_j \text{ are associated words} \\ \infty & -t_i, t_j \text{ are the same word} \end{cases}$$

The values for the weights were evaluated based on the following principles:

1. Semantic distances between synonyms are assumed to be smaller than semantic distances between associated words;

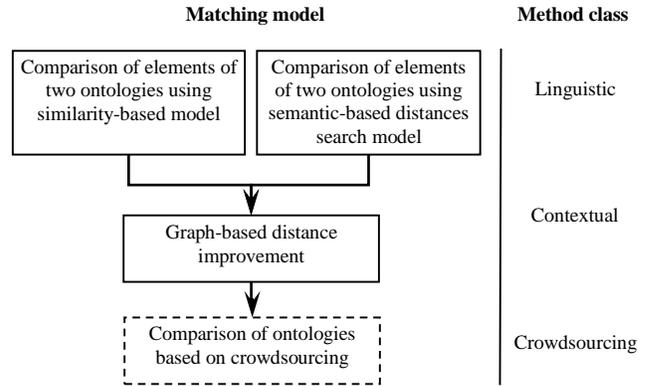


Figure 1. Multi-model approach to automated ontology matching.

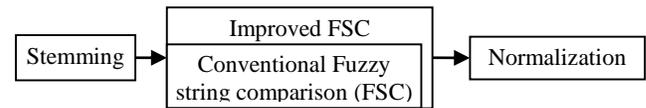


Figure 2. Similarity-based model

2. Semantic distance is proposed to be calculated as inversely proportional to weights raised to a power. The power is proportional to the path between the compared words. The longer the path, the greater the semantic distance for the two different words is expected to be. To meet this expectation with reference to the way of the semantic distance calculation, a weight of the relation between two different words should be in the range (0, 1) and ∞ . Taken into account the first principle, we empirically selected the weights: 0,5 - for the relation between the synonyms; and 0,3 - for the relation between the associated words;

3. The semantic distance between the same words is equal to 0.

To search the semantic distance between the elements of two ontologies, the nodes of the first ontology are checked for their similarity to nodes of the second ontology. As a measure of similarity, the semantic distance (Dist) is used.

$$Dist(t_i, t_j) = \frac{1}{\sum_S \prod_{k=s_i}^{s_j} w_k}$$

where t_i, t_j – ontology elements; w – weight of lexical relation existing between t_i and t_j ; S – a set of paths from t_i to t_j , where a path s is formed by any number of links that connect t_i and t_j passing through any number of nodes. The degree of similarity depends inversely on distance.

D. Graph-based distance improvement model

The graph-based improvement model for propagation similarities from one ontology element to another is presented in Figure 3 (see [36]). The main goal of this model is to propagate the degree of similarity between closely matching ontology elements to ontology elements related to them through RDF triples.

Let $X=(x_1, x_2, \dots, x_n)$ be the set of subjects and objects in the ontology of two knowledge processors. Let $D_x = (d(x_i, x_j), \dots)$ be a degree of similarity between x_i and x_j . Let $R = (r_1, r_2, \dots, r_n)$ be a set of predicates in the ontology of two knowledge processors. Let $D_r = (d(r_i, r_j), \dots)$ be a set of degrees of similarity between r_i and r_j . Constant Tr is a threshold value that determines whether two ontology elements mapped to each other or not.

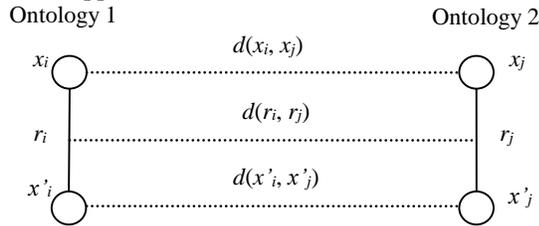


Figure 3. Matching of two ontology model

E. Crowdsourcing comparison of ontologies

Crowdsourcing is the process of obtaining information from an online group of crowd members. Technically, members are registered in a special application that provides them micro tasks and then summarizes their responses. For ontology matching process, this technique is used for making alignments between ontology elements if the methods presented above fail. If the service estimates that it needs more matching information it can use the crowdsourcing technique to improve the ontology matching with the help of a group of members.

Crowdsourcing system uses matrix M to show the ontology matching results found in steps 1–3 to the members, and provides a user friendly interfaces for them to see these alignments and to add, remove, or modify the matching between the ontology elements.

IV. CASE STUDY

The aim of the considered scenario is providing interoperability support for robots to solve a joint task. Two types of robots participate in the scenario: a pipeline robot and a manipulating robot (see Figure 4). The first one is stationary and has a pipeline that transfers objects from their current location to a predefined destination. It has a color sensor that determines the color of the transferred object. Robots interact in a smart space. To provide for semantic interoperability between robots, the proposed ontology matching approach is used. Each robot uploads its own ontology to the smart space when the robot joins it. The ontology matching service performs matching of the uploaded ontology with the smart space ontology and then extends the latter with the elements of the ontologies uploaded by robots.

The proposed interaction scheme of two robots in the smart space is presented in Figure 5. Example of robot interaction is shown in the example of sharing information about object transfer by the pipeline robot and color identification. When the pipeline robot is transferring the object, the pipeline velocity is shared with smart space by the following triple according to the pipeline robot ontology.

(“Pipeline”, “has_velocity”, [pipeline velocity]).

When the color is determined, it is shared with smart space as follows.

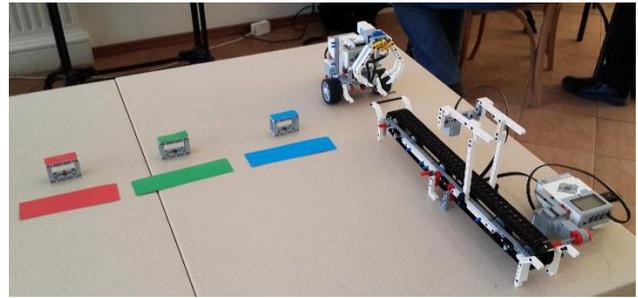


Figure 4. Pick-and-Place System Scenario

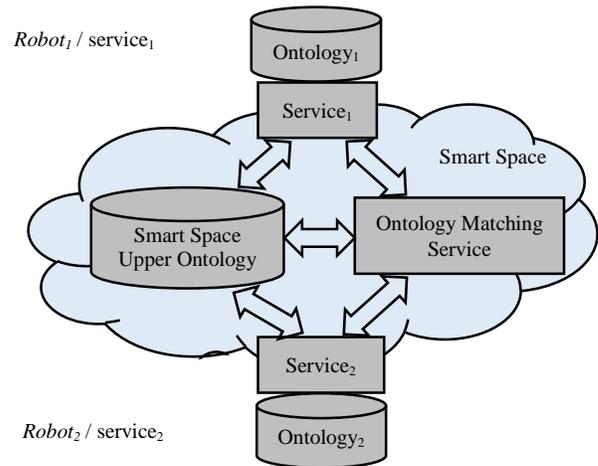


Figure 5. Robot Interaction in Smart Space Based on Ontology Matching

(“Object”, “has_color”, [object color]).

When the object has been moved to the destination point and is ready for manipulation by the manipulating robot, the related triple is shared with smart space by pipeline robot.

(“Object”, “is_ready_for_manipulation”, 1).

(“Pipeline”, “has_velocity”, 0).

V. CONCLUSION

The paper presents the state-of-the-art of ontology matching works and proposes the crowdsourcing approach for matching ontology elements of a service-based system. The approach allows to implement matching of ontology elements with the help of group of people that improves the “on-the-fly” ontology matching approach. The considered case study is based on smart space technology that provides ontology-based information sharing between different system components and implements the ontology matching approach.

ACKNOWLEDGMENT

The presented results are part of the research carried out within the project funded by grants # 13-07-00336, 13-07-12095 of the Russian Foundation for Basic Research.

REFERENCES

- [1] D. J. Cook and S. K. Das, "How smart are our environments? an updated look at the state of the art", *Pervasive and Mobile Computing*, vol. 3, no. 2, pp. 53-73, 2007.
- [2] S. Balandin and H. Waris, "Key properties in the development of smart spaces," *Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction*, Springer, 2009, pp. 3-12.
- [3] Smart-M3 at Sourceforge, Web: <http://sourceforge.net/projects/smart-m3> [retrieved: August, 2015].
- [4] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, "Smart-M3 Information Sharing Platform," *Proc. ISCC 2010, IEEE Comp. Soc.*; Jun. 2010, pp. 1041-1046.
- [5] D. AnHai, M. Jayant, D. Pedro, and H. Alon, "Ontology Matching: A Machine Learning Approach," *Handbook on Ontologies in Information Systems*, 2004.
- [6] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," *Proceedings of the 11th international conference on World Wide Web*, 2002, pp. 662-673.
- [7] W. Hu, N. Jian, Y. Qu, and Y. Wang, "GMO: A Graph Matching for Ontologies," *K-CAP Workshop on Integrating Ontologies*, 2005, pp. 43-50.
- [8] A. Alasoud, V. Haarslev, and N. Shiri, "An Effective Ontology Matching Technique," *17th International Symposium ISMIS 2008, Toronto, Canada, May 2008*, pp. 585-590.
- [9] E. Hovy, "Combining and standardizing largescale, practical ontologies for machine translation and other uses," *The First International Conference on Language Resources and Evaluation (LREC)*, Granada, Spain, 1998, pp. 535-542.
- [10] P. Mitra, G. Wiederhold, and J. Jannink, "Semi-automatic Integration of Knowledge Sources," *2nd International Conference on Information Fusion*, Sunnyvale, CA, July 1999.
- [11] P. Mitra, M. Kersten, and G. Wiederhold, "Graph-Oriented Model for Articulation of Ontology Interdependencies," *Proceedings of the 7th Int. Conf. on Extending Database Technology*, Springer-Verlag, 2000.
- [12] N. Noy and M. Musen, "Anchor-PROMPT: Using Non-Local Context for Semantic Matching," *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, USA, 2001.
- [13] S. Castano, A. Ferrara, and S. Montanelli, "H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems," *Proc. of the 1st VLDB Int. Workshop on Semantic Web and Databases*, 2003.
- [14] L. Serafini, P. Bouquet, B. Magnini, and S. Zanobini, "An algorithm for matching contextualized schemas via SAT," *Technical report*, DIT University of trento, Italy, 2003.
- [15] N. Noy and M. Musen, "SMART: Automated Support for Ontology Merging and Alignment," *12th Workshop on Knowledge Acquisition, Modeling, and Management*, Banff, Alberta, 1999.
- [16] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "An Environment for Merging and Testing Large Ontologies," *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000)*. Breckenridge, Colorado, USA, 2000, <http://www.ksl.stanford.edu/software/chimaera/> [retrieved: August, 2015].
- [17] D. Aumueller, H. Do, S. Massmann, and E. Rahm, "Schema and Ontology Matching with COMA++," *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 906-908.
- [18] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its application to schema matching," *Proceedings. 18th International Conference on Data Engineering, USA, 2002*, pp. 117-128.
- [19] I. Cruz, F. Antonelli, and C. Stroe, "Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods," *The Fourth International Workshop on Ontology Matching*, Washington DC., 2009.
- [20] D. Ritze, C. Meilicke, O. Šváb-Zamazal, and H. Stuckenschmidt, "A pattern-based ontology matching approach for detecting complex correspondences," *The Fourth International Workshop on Ontology Matching*, Washington DC., 2009.
- [21] [F. Giunchiglia, V. Maltese, and A. Autayeu, "Computing Minimal Mappings," *The Fourth International Workshop on Ontology Matching*, Washington DC., 2009.
- [22] M. Klein, W. Kiryakov, D. Ognyanov, and D. Fensel, "Ontology Versioning and Change Detection on the We," *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, Sigiienza, Spain, 2002.
- [23] V. Bui, R. Verhoeven, and J. Lukkien, "A body sensor platform for concurrent applications," *Proc. of IEEE Int. Conf. on Consumer Electronics*, 2012, pp. 38-42.
- [24] V. Bui, P. Brandt, H. Liu, T. Basten, and J. Lukkien, "Semantic Interoperability in Body Area Sensor Networks and Applications," *9th International Conference on Body Area Networks*, London, Great Britain, Sep. 2014, pp. 210-216.
- [25] P. Lambrix and H. Tan, "SAMBO – a system for aligning and merging biomedical ontologies," *Journal of Web Semantics*, vol. 4, no. 1, pp. 196-206, 2006.
- [26] W. Hu, Y. Qu, and G. Cheng, "Matching large ontologies: A divide-and-conquer approach," *Data and Knowledge Engineering*, vol. 67, no. 1, pp. 140-160, 2008.
- [27] M. Nagy and M. Vargas-Vera, "Towards an automatic semantic data integration: Multi-agent framework approach," *Semantic Web*, ch. 7, pp. 107-134, 2010.
- [28] J. Li, J. Tang, Y. Li, and Q. Luo, "Rimom: A dynamic multistrategy ontology alignment framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 8, pp. 1218-1232, 2009.
- [29] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Journal of Web Semantics*, vol. 7, no. 3, pp. 235-251, 2009.
- [30] M. S. Hanif and M. Aono, "An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size," *Journal of Web Semantics*, vol. 7, no. 4, pp. 344-356, 2009.
- [31] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *Proc. 15th International Conference on Data Engineering*, pp. 512-521, 1999.
- [32] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," *Proc. 4th International Semantic Web Conference (ISWC)*, 2005, pp. 624-637.
- [33] G. Shafer, "A Mathematical Theory of Evidence," *Princeton University Press*, 1976.
- [34] Wordnet, <https://wordnet.princeton.edu/> [retrieved: August, 2015].
- [35] Wiktionary, <https://ru.wiktionary.org/> [retrieved: August, 2015].
- [36] A. Smirnov, A. Kashevnik, N. Shilov, S. Balandin, I. Oliver, and S. Boldyrev, "On-the-Fly Ontology Matching in Smart Spaces: A Multi-Model Approach," *Proceedings of the Third Conference on Smart Spaces*, 2010, pp. 72-83.

Object-Oriented Communication Model for an Agent-Based Inventory Operations Management

Rafal Cupek, Adam Ziebinski, Lukasz Huczala
Institute of Informatics
Silesian University of Technology
Gliwice, Poland
Rafal.Cupek@polsl.pl

Daniel Grossmann, Markus Bregulla
Ingolstadt University of Applied Sciences
Ingolstadt, Germany
Daniel.Grossmann@thi.de

Abstract—this document presents the idea of an autonomous mobile platform that was developed for Inventory Operations Management support that is directly connected and acts as part of a Manufacturing Execution System. The proposed Multi-Agent System supports logistic operations in manufacturing. This article focuses on the object-oriented communication model for a multi-agent logistic system that is based on standardized MES services that are defined according to the ISA95 model and object-oriented communication services implemented in OPC UA standard.

Keywords—*Manufacturing Operations System; MES; Inventory Operations Management; Autonomous Systems; industrial communication; Industry 4.0; OPC UA; ISA95.*

I. INTRODUCTION

Inventory Operations Management (IOM) is the process of planning and directing that allows for the easy control and effective flow of materials, semi-products and final products, as well as to the information flow from the source of the information to its destination in order to fulfil a customer's requirements. One of the main goals of an IOM is to minimize the final cost of products by reducing the costs of transportation, management and storage in warehouses [1].

IOM execution is supported by logistics systems that help to improve the processes for the management of an enterprise and to perform the necessary analyses that are necessary to reduce costs. They include analyses of the loading, transferring, unloading and storage of products in the supply cycles for both the production and delivery stages. Logistics systems can be grouped according to different criteria [2] such as: functionalities (systems of supply, production or distribution), structural-decision-functional criteria (systems of planning, controlling and organizing) or hierarchical criteria (normative, strategic and operating systems). As was shown in [3], autonomous vehicles are one of key components in the development of flexible and efficient transport systems for logistics and industrial site management applications. Such a system needs the distribution of intelligence across devices as well as logistics and business applications.

An IOM cannot function in isolation from a Manufacturing Execution System (MES). One of the commonly accepted definitions of MES activities can be

found in the set of documents that is managed by MESA, the International Manufacturing Enterprise Solutions Association, and expressed as the ANSI/ISA95 (IEC/ISO 62264) norms that are the international standard for the integration of enterprise and control systems [4].

ISA95 defines the MES data structure and MES services that are related to manufacturing operations: defining the product, forecasting production, managing production capability and evaluating production performances. ISA95 consists of models and terminology and describes the information that is exchanged between the systems for sales, finance and logistics and the systems for production, maintenance and quality. This information is structured in the form of UML (Unified Modelling Language) models, which are the basis for the development of standard interfaces between ERP (Enterprise Resource Planning) and MES systems. ISA95 is built on an object-oriented model that defines the interface between the control systems and a business application. It also defines the services that are required for the manufacturing support that is designed according to the object-oriented model [5].

One of the leading examples of the Service-Oriented communication standards that are used in industry is OPC UA (Open Production Connectivity Unified Architecture). OPC UA is a service-based architecture that relies on Web Services for communication with enterprise management systems and TCP-based communication for communication with control and HMI (Human Machine Interfaces) [6, 7]. OPC is maintained by the OPC Foundation [8] and is recommended by Industry 4.0 guidelines. As was shown in [9], OPC UA can be applied as a communication interface between an MES (Manufacturing Execution System) and the real-time devices that are used for logistics operations. OPC UA is based on an object-oriented model that allows a flexible communication interface to be created that can be used in heterogeneous logistics systems. OPC UA supports logistics data modelling and annotating raw input data with useful semantic information that supports logistics decisions.

This article focuses on the object-oriented communication interface model for an intelligent agent system that is designed for logistic operations. The proposed model follows the existing manufacturing standards and reflects the system requirements that have been defined for cyber-physical systems that are developed in accordance

with the model defined by Industry 4.0 [10]. The authors propose an agent-based architecture that is connected to an MES according to the ISA 95 standard. The proposed communication interface is based on OPC UA with a special focus on the ISA95 OPC UA information model that is accepted in the guidelines of Industry 4.0.

The rest of this paper is organized as follows: the agent-based architecture concept of the Autonomous Mobile Platform (AMP) is presented in chapter two. Some of the details of the implementation of the AMP platform are given in chapter three. On the one hand, an AMP is an autonomous device that is designed to support material transportation, while on the other hand, an AMP is part of the inventory operations management support and from this point of view, it is part of a distributed agent-based MES. The concept of a data model that is based on the OPC UA communication interface and joins the AMP and MES according to the ISA95 architecture is described in chapter four. The conclusions are presented in chapter five.

II. AGENT-BASED ARCHITECTURE FOR INVENTORY OPERATIONS MANAGEMENT (IOM)

In order to implement our Distributed Agent-Based Architecture concept, we decided to use the inventory operations management activity model that was proposed in the third part of the ISA95 standard. This model reflects the four main information streams that are exchanged between an MES and enterprise management systems. Inventory definitions describe the rules and information that are associated with the movement and storage of materials. These rules may be location specific, equipment specific, physical asset specific or material specific. Inventory capability is a capability measure of the ability to handle materials for specific time horizons and is characterized by the type of material, available storage space (or volume) and type of storage. An inventory request is used to define requests to transfer materials. An inventory response is used to respond to an inventory request and indicates the completion status (successful or unsuccessful) of the request [11].

Although the structure of the activity that is proposed in the ISA95 seems fine for individual functions, it cannot be effectively implemented in multi-agent systems. ISA95 defines detailed inventory scheduling as a collection of activities that take inventory requests and generate work schedules for inventory. This reflects the hierarchical approach top to bottom that is used in planning systems but does not reflect the dynamics of a real logistics system. Instead of duplicating the structure that is proposed by the ISA95, the authors propose the division of the inventory operations management between agents as shown in Fig. 1.

The Logistics Agent (LA) is a communication entry point that is responsible for collecting inventory requests from the MES and is the owner of the Transportation Request list. Detailed inventory scheduling is created by Delivery Agents (DA) and is achieved by distributed and collaborative work. Each inventory request generates one instance of a Delivery Agent (DA) that is responsible for the realization of the delivery. Delivery Agents fulfil the detailed inventory

scheduling and Inventory dispatching functionality as defined by ISA95. They analyze transportation possibilities and the costs that are offered by Transportation Agents (TA). Transportation Agents are holons that are directly coupled with mobile platforms. Each mobile platform has its own TA, which manages the Confirmed Orders list and the Required Orders list. Each delivery request that is sent by a DA is placed on the Required Orders list. Deliveries that are confirmed by a DA are moved to the Confirmed Orders list. A DA can also cancel a delivery order that has been accepted by a TA for realization but has not been started yet. In such a case, the delivery record from the Confirmed Orders list is moved back to the Required Orders list.

A DA bases its decisions on the delivery costs that are exposed by a TA. A DA sends a response to the TA that can be either a Confirmation or a Cancellation of the delivery service that was proposed by the TA. Every new Order that is accepted by the TA can change the cost for other Confirmed Orders. This information is sent back from the TA to interested DAs including both owners of Confirmed Orders and Required Orders. Based on these, the DA can decide to change the TA and send the message to cancel the delivery. This may result in a cost change for other orders. To make the system more stable, the Cancellation cost, which is a function of the Time to start transportation service, has to be added to the total delivery cost.

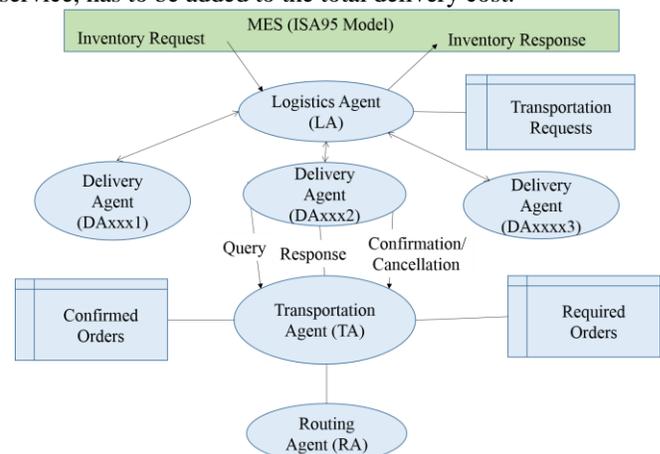


Figure 1. Proposed IT architecture for an agent-based IOM system.

III. ARCHITECTURE FOR AN AUTONOMOUS MOBILE PLATFORM (AMP)

An AMP is designed as a modular project that is based on a simple mobile platform, an engine control module and a movement control system. The authors assume that the mobile platform and engine controls are one of the commercially available solutions and the control system is realized on the Raspberry Pi system.

The AMP is supervised and monitored by the Distributed Logistics System of a Transportation Agent (TA). In the proposed solution, the TA operates as a holon for the AMP and is responsible for the execution of the transportation services that are ordered by the DAs. Motion control of the AMP is executed through an application running on the

RaspberryPI that interprets the signals that are collected from the sensors that are installed on the platform and they perform the communication tasks that are received from the TA. Communication between the logistics system and the TA based on the RaspberryPI platform is performed via a Wi-Fi/GPRS. The TA is able export its services to the logistics system.

The movement control system of an AMP allows it to drive a vehicle forward and backward, turn left and right and rotate in place. In order to achieve this functionality, an AMP can be equipped (depending on the AMP configuration) with a set of sensors that includes a Gyroscope, Accelerometer, GPS, Encoders, Short Range Radar and Lidar, Ultrasound and Camera. It was assumed that the AMP will be moved along a known route on a flat terrain – pavement or a corridor in a building. The Transportation Agent allows data to be exchanged between the AMP and other agents. A Routing Agent (RA) is responsible for preparing the route for the Transportation Agent. The RA will send part of the route parameters to the TA based on the receipt of subsequent jobs – short segments of the route include movement direction, velocity, distance, order of delivery, etc. Next the Transportation Agent sends this information to the AMP controller.

Another functionality is the continuous monitoring of the AMP route by Transportation Agent using the RA. This will allow information from the sensors, camera and the control signals to be written into the database. That data will allow the Transportation Agent to monitor and analyze the route that is travelled, The RaspberryPI application, which supports real time functions, is responsible for the movement, position, velocity, and detection of the distance to obstacles and current consumption.

An additional functionality is the continuous recording of the route and sending this information to the Transportation Agent. The information that is sent includes data from the camera, sensors, orders that were issued, power that was consumed, etc. When the system on the AMP detects an obstacle, it will try to bypass the current obstacle, and in the event of failure, the AMP will scan the surrounding area and try to find another way to drive to the designated point. In the event of failure to find good way in this mode, the Transportation Agent will ask the RA to prepare a new route. The system will go into manual mode if the RA cannot prepare a correct route.

IV. OPEN PRODUCTION CONNECTIVITY UNIFIED ARCHITECTURE (OPC UA) INTERFACE FOR INVENTORY OPERATIONS MANAGEMENT

OPC UA is an object-oriented and service-based communication interface. OPC UA joins the functionality that was offered by the previous sets of OPC specifications that are now integrated into a unified service set that is defined on an abstract level. The OPC UA services are organized into ten Service Sets: Discovery, SecureChannel, Session, NodeManagement, View, Query, Attribute, Method, MonitoredItem and Subscription. The service sets contain 37 services of which 16 are used for actual information

exchange and 21 are used to manage the communication infrastructure. All of the OPC UA functionality is based on these 37 services, which represent all of the possible interactions between a UA client and UA server applications.

The services that are defined by OPC UA are platform independent and are defined on an abstract level. There are various OPC stack implementations including binary UA TCP encoding and an XML encoding with the SOAP/ HTTP transport protocol. Information exchange is managed by a Secure Channel, which defines the long-running logical connection between an OPC UA Client and Server. This channel maintains a set of Public-key infrastructure (PKI) keys that are known only to the Client and the Server and that are used to authenticate and encrypt messages that are sent across the network.

Information exchange is started by establishing a secure connection between the Client and the Server so that a Client can then browse or query a server's address space if necessary. Client-Server Sessions are defined over underlying transport layers and may be managed by a Session Service Set. OPC UA communication uses the Session Service, while the OPC UA defines the session level services – Create Session, Activate Session and Close Session services. Such a structure supports the interfaces robustly in the event of communication errors because sessions can survive communication channel breaks. The services of OPC UA are defined on an abstract level. They have already been implemented with a binary UA TCP encoding and an XML encoding with the SOAP/ HTTP transport protocol. Information exchange is managed by a Secure Channel, which defines the long-running logical connection between an OPC UA Client and Server. This channel maintains a set of Public-key infrastructure (PKI) keys that are only known to the Client and the Server and that are used to authenticate and encrypt the messages that are sent across the network.

Although Read/Write services, which are defined in an Attribute Service Set, allow Clients to have direct information access, a more efficient communication may be established based on the well-known Publisher/Subscriber model that is supported by the MonitoredItem and Subscription service sets. In this model, a Client can subscribe to Server parameters for a given set of MonitoredItems – Sampling Interval (ms), Queue Size and Filter Conditions that define the Trigger condition (status, value/status, source timestamp/value/status) and Deadband (Absolute, Percent). According to the tracking conditions that are defined, an OPC UA Server Client send Publish requests (there is no longer call back mechanism) for given Session to the server. Data transfers are optimized by grouping modified MonitoredItems within sessions for efficiency. Published request responses are not sent immediately, but are queried by the Server, whereas a Publish Response is sent back according to the Subscription's publishing interval. NotificationMessages contain Notifications of any monitored items that have not yet been reported to the Client. If no notification is available, the server sends a life-ping to the client.

OPC UA is scalable and can be used with servers that have high computational power and rich resources but it can also be applied in smart sensors with very limited resources. The individual features are grouped into ConformanceUnits, which are further grouped into Profiles. This allows the actual service set to be adjusted to the application requirements on the one hand and to the hardware limitations on the other. One example of profiles with very low hardware requirements is the OPC UA Nano Embedded Device profile, which describes a reduced functionality and simplified set of services. This profile is fully compliant with the other parts of the OPC UA standard and can be run on platforms with very limited resources. In [12], it was demonstrated that such an environment can even be reduced to a simple FPGA platform that has very limited hardware and software resources.

The application profiles are defined based on the basic service sets. They reflect the main areas of the application of OPC UA – Data Access, which gives subscription-based information about the current values of the parameters that are required by the OPC UA Client; Alarms & Conditions, which are organized as event-based communications that are restricted to the area of interests that is defined by the client; a Historical Access interface for and Programs interface, which allows the server-side application to be controlled by commands that are sent by the client. OPC UA has been an IEC standard (IEC 62541) since 2012.

Nowadays, the OPC Foundation supports different working groups that are involved in the development of the industry-leading specifications, technologies, certifications and processes that are related to OPC UA. One of them is the ISA95 Working Group, which is responsible for defining and maintaining the ISA95 OPC UA information model. The first release of the ISA95 OPC UA specifications includes support for the following ISA95 [4] models: Physical Assets, Equipment, Personnel and Material Handling. This mapping of the abstract ISA95 model provides a high-speed, secure information flow from the lowest levels of the automation hierarchy to the Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems.

The features above allow a scalable OPC UA address space that can be implemented both a mobile platform with limited resources and on MES servers that have far more hardware and software resources to be designed. The proposed OPC UA-based communication model consists of a set of OPC UA servers. Each OPC UA server is directly connected with one agent: LA, DA, TA and RA. Each of the agents also has an OPC UA client that is used for communication with the OPC UA servers as shown in Fig. 2.

The OPC UA address space is, by definition, distributed and can include many servers and many clients that are connected by references (if part of information is stored on another server) and subscriptions (to define the information that is important to a given client). Three powerful mechanisms that are offered by OPC UA were used for this purpose:

An OPC UA-type definition offers the possibility to create the object-oriented data structures that their clients need with the possibility to discover the object data

structures that are managed by servers. The OPC UA address space includes knowledge about the organization of the available information and presents this knowledge to any application that is connected on the client side. A client can use object-oriented data types to discover their definitions, which are stored on the server. The references and cross-server references are part of the type definition. Types organize data into a structure that includes object-oriented mechanisms such as subtyping and type inheritance. Types can be used for objects, variables, data and reference definitions.

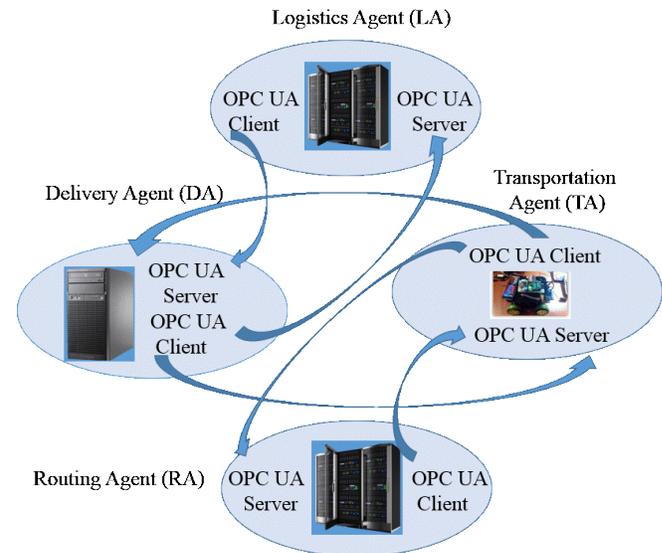


Figure 2. OPC UA-based architecture for a multi-agent IOM system.

The AMP address space defines its own types, which are related to its IOM functionality, such as details about materials, production orders, production schedule and other information that is related to logistics operations. These types are defined according to the functionality of the AMP, the ISA 95 model and the IOM information documents that are exchanged with the Enterprise Management System (ERP), which are based on SAP software (as presented in an example below). The OPC UA address space part that is defined for an AMP reflects the information that is related to route planning such as road maps, obstacle identification and information about other vehicles, etc. Actual information is presented by objects that reflect the operations that are carried by the IOM on the one hand and from the objects that describe the current situation that at the place of operation of AMP on the other hand.

Type definitions reflect the real situation that is presented by the objects that are class instances according to given types. Objects reflect actual information about variables, properties and methods. Variables are used to present the real process signals that change during the execution of the process, the properties that are used to describe an object and in a case in which more complicated activity on the object is needed, the methods that are used. The object-oriented structure is also supported by events mechanisms, which

means that there is no longer a need for cyclical data pulling and continuous object state checking. The information is automatically sent to the client that has subscribed to a given event and that receives the required message in the event of its evidence. The subscription mechanism is used to track all kinds of data changes in a consistent way. These include Variables Values, Aggregated Values and Events. According to the SOA model, subscriptions can be managed by the OPC UA MonitoredItem Service Set.

The OPC UA address space for the whole MES can be very complicated. The physical memory of an AMP and communication bandwidth makes it impossible to access all of the IOM information in every TA. This problem can be solved by using a reference-based view mechanism. The view mechanism, which is supported by OPC UA, allows the whole address space that is managed by the OPC UA servers to be reduced to only the information that is necessary for a given client. Views help to organize large data structures in order to present the information that is important in given contexts. In the case of an AMP, it can be used to limit the presentation of information to only the scope that is necessary for a given mobile platform. The information about the IOM is limited to the area that is in the domain of a given TA including any possible transport services. The selected part of OPC UA address space that presents the type definition of the TA and AMP objects is shown in Fig. 3.

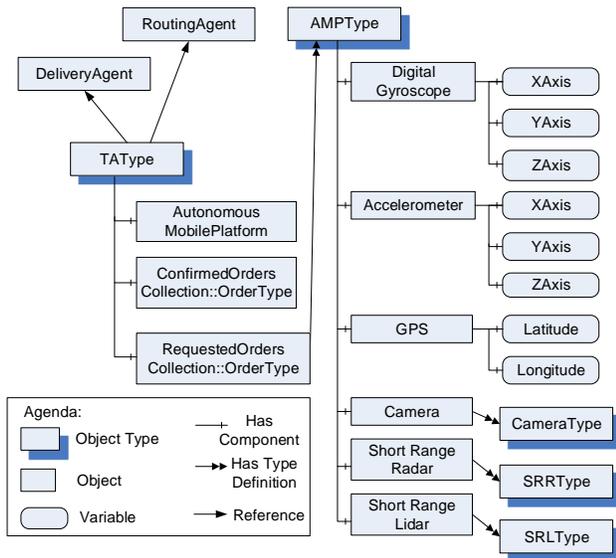


Figure 3. OPC UA address space for an AMP (selected view).

Each TA creates one instance of TAType that is hosted by the OPC UA server. The Information that is presented is used by the DA (collections of Orders) as well as by the AMP in order to exchange information about the ordered transport services and is used for optimal delivery routing. In addition, the TAType contains the AMPType object that presents information related to the location and movement of the mobile platform. This object is also hosted by an OPCUA Server that is owned by the TA. This information is required for the TA to calculate delivery cost.

The productive implementation of the system requires data exchange with the SAP Warehouse Management system, and therefore the SAP standard message types had to be created in the OPC UA address space. The SAP transfer order contains all of the information that is required in order to execute the physical transfer of materials into the warehouse, out of the warehouse or from one storage bin to another storage bin within the warehouse. When you confirm a transfer order, you inform the system that it has been processed and that the goods have arrived at the intended destination.

A transfer order contains all of the necessary information on a planned movement of goods. Additionally, a material definition consisting of the size, weight, transport requirements, etc. is accessible via separate message channel. The SAP data is mapped into an ISA95 MaterialClassType. The required messages in IDOC standard are listed in Table 1, which is based on SAP requirements.

TABLE I. INFORMATION MODEL FOR TRANSFER ORDER IN OPC UA

SAP IDoc message		IDoc data segment	
Transfer orders	WMTORD	E2LTORH	Transfer order header
		E2LTORI	Transfer order items
Confirmation of transfer orders	WMTOCO	E2LTCOX	Confirm entire storage
		E2LTCOH	unit (multiple orders)
		E2LTCOI	Confirm entire transfer order
			Confirmation of a transfer order item
Cancellation request	WMCATO	E2LTCAH	Header data
		E2LTCAI	Item data
Material master	MATMAS	E1MARCM	Material Class segment
		E1MARDM	Warehouse / batch segment
		E1MARMM	segment
		E1MEANM	Units of measure
			European Article Number

There are two possible scenarios for creating a message transfer between the Logistics Agent and the Warehouse Management system: OPC UA connection by means of SAP Plant Connectivity (PCo) – IDOC transformation on the SAP side and EDI connection by means of, e. g., IBM InfoSphere Information Server – IDOC transformation on LA side.

The OPC UA data model that is selected has to reflect both the ISA95 and SAP transfer order business process and the data types of the management system. The information

model that is proposed for a Transfer Order in the OPC UA address space is presented Fig. 4.

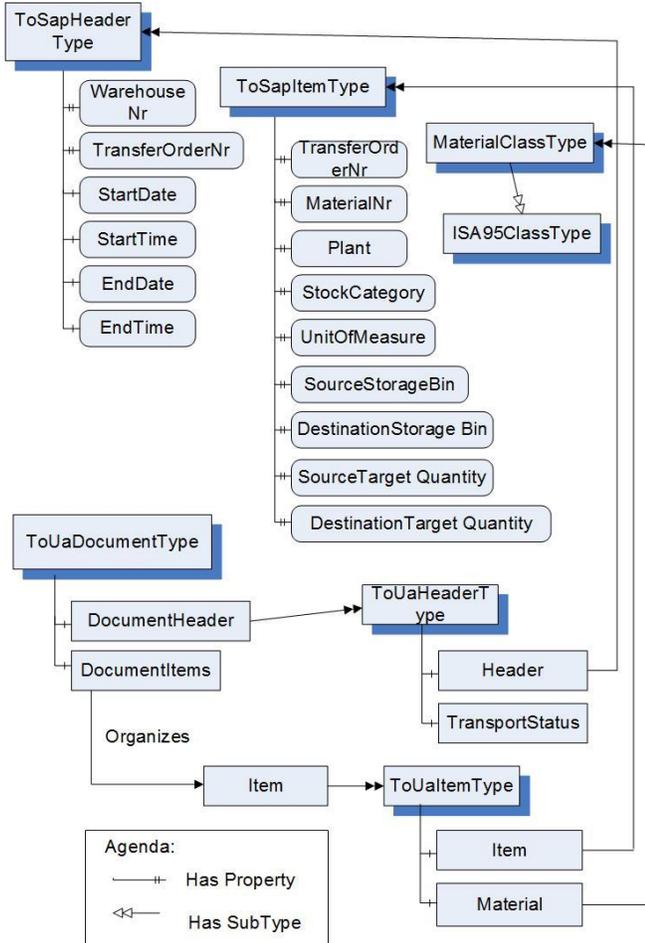


Figure 4. Information model for a Transfer Order in OPC UA [23].

The operations request model for ISA 95 [13] defines the following attributes for operations request objects:

- ID – A unique identification for the operations request
- Description – additional information and descriptions of the operations request
- Operations type – in our case Inventory
- Start Time – When the operation is to be started
- End Time – When the operation is to be completed
- Priority – The priority of the request
- Hierarchy Scope – Identifies where the exchanged information fits within the role-based equipment hierarchy
- Operations Definition ID – Identifies the associated Operations definition that are to be used
- Request State – Indicates the state of the operations request.

V. CONCLUSIONS

The proposed model for IOM systems is based on a multi-agent architecture that is controlled by the flow of events. It is based on the hierarchical model that was proposed by ISA95. For the ERP (SAP) information is presented in a model that is compatible with an ISA95 Operations request. Detailed Inventory scheduling and Inventory dispatching are created and modified dynamically via the interaction of Delivery Agents and Transportation Agents. A detailed communication model is proposed based on the distributed OPC UA address space definition that was adjusted for a multi-agent environment. The next planned step is to validate the stability and efficiency of the model. The authors plan to reach this goal by performing a simulation.

ACKNOWLEDGMENT

This work was supported by the European Union from the FP7-PEOPLE-2013-IAPP AutoUniMo project “Automotive Production Engineering Unified Perspective based on Data Mining Methods and Virtual Factory Model” (grant agreement no: 612207).

REFERENCES

- [1] C. F. Daganzo, Logistics Systems Analysis, Springer-Verlag, 1999.
- [2] T. Gudehus, H. Kotzab, Comprehensive Logistics, Springer-Verlag, 2009.
- [3] Andreasson, Henrik, et al. "Autonomous Transport Vehicles: Where We Are and What Is Missing. " Robotics & Automation Magazine, IEEE 22. 1 (2015): 64-75.
- [4] <http://www.isa-95.com>.
- [5] B.Scholten, The road to integration: A guide to applying the ISA-95 standard in manufacturing. Isa, 2007.
- [6] W. Mahnke, S. -H. Leitner and M. Damm, OPC Unified Architecture, Springer-Verlag Berlin Heilderberg New York, 2009.
- [7] J. Lange, F. Iwanitz, T. J. Burke, “OPC – From Data Access to Unified Architecture”, VDE Verlag, 2010.
- [8] <https://opcfoundation.org>
- [9] A.Maka, R.Cupek, and J.Rosner, "OPC UA Object-Oriented Model for Public Transportation System. " Computer Modeling and Simulation (EMS), 2011 Fifth UKSim European Symposium on. IEEE, 2011.
- [10] N.Jazdi, "Cyber physical systems in the context of Industry 4. 0. " Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on. IEEE, 2014.
- [11] ANSI/ISA95. 00. 03-2013 (IEC 62264-3 Modified) Enterprise-Control System Integration – Part 3: Activity Models of Manufacturing Operations Management; Approved 8 July 2013.
- [12] R. Cupek, A. Ziebinski, and M. Franek, "FPGA BASED OPC UA EMBEDDED INDUSTRIAL DATA SERVER IMPLEMENTATION. " Journal of Circuits, Systems and Computers 22. 08 (2013).
- [13] ANSI/ISA95. 00. 02-2010 (IEC 62264-2 Mod) Enterprise-Control System Integration – Part 2: Object Model Attributes; Approved 13 May 2010.

Granular Meta-Ontology and Extended Allen's logic: Some Theoretical Background and Application to Intelligent Product Lifecycle Management Systems

Valery B. Tarassov Alena V. Fedotova

CIM Department

Bauman Moscow State Technical University

Moscow, Russia

e-mail: vbulbov@yahoo.com, afedotova.bmstu@gmail.com

Rainer Stark

Faculty of Mechanical Engineering and Transport Systems

Berlin University of Technology (TU Berlin)

Berlin, Germany

e-mail: rainer.stark@tu-berlin.de

Baurzhan S. Karabekov

Institute of Informational and Computational Technologies

Ministry of Education and Science of Kazakhstan

Almaty, Kazakhstan

e-mail: bskarabekov@mail.ru

Abstract—A hierarchical system of ontologies is considered, where the concepts of meta-ontology and upper ontology are of primary concern. The concept of meta-ontology is discussed; the distinction between meta-ontology and upper ontology is shown. Various methodologies for constructing formal ontologies are analyzed. There is a need for a generalized approach to ontological modeling based of Maltsev's algebraic systems is justified. Basic principles of information granulation and granular ontology construction are formulated; some formal definitions of granular meta-ontologies together with fuzzy and linguistic ontologies (based on extended linguistic variables) are introduced. An application of granular meta-ontology and upper ontology concepts to lifecycle modeling is considered in the context of building new generation product lifecycle management systems – intelligent Product Lifecycle Management (PLM) system. Circular and sequential lifecycle representations are constructed and interpreted as coarse-grained and fine-grained ontologies. The use of extended Allen's interval logic as an integrated parallel-sequential lifecycle ontology is suggested.

Keywords—*Ontological engineering; granular meta-ontology; Allen's logic; lifecycle ontologies; intelligent PLM.*

I. INTRODUCTION

A main theoretical purpose of this paper consists in bridging the gap between two different scientific areas, named by the same term – “Ontology”. On one hand, ontology is a classical philosophical discipline that studies the nature of existence, the basic categories of being and their relations. It faces problems, such as what is a thing, why various entities exist, and how these entities may be grouped, related within a hierarchy, and subdivided according to similarities and differences [7]. On the other hand, in computer science and artificial intelligence,

ontology is a description (like a formal specification of a program) of the concepts and relationships [9] that can formally exist for an agent or a community of agents.

It can be argued that in computer science «exist» means «be represented in a computer model». So, any ontology defines a set of representational primitives in order to model a domain of knowledge or discourse. These representational primitives are typically classes (or sets), attributes (or properties) and relationships (or relations among class members) [1].

From a methodological viewpoint, ontological engineering is an adoption of modern systemic, highly interdisciplinary approach in computer science. This field studies the methods and methodologies for building various ontologies: formal representations of a set of concepts within a domain and the relationships between those concepts. A large-scaled representation of the most abstract concepts, such as system, relation, time, space, action, event, etc., is the core of ontological engineering [13], [14].

Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies. For this reason, ontologies are said to be at the "semantic" level, whereas database schema are models of data at the "logical" or "physical" level.

A practical problem to be solved consists in developing product (system's) lifecycle models based on granular meta-ontology and extended Allen's logic. These granular models will contribute to the development of intelligent Product Lifecycle Management (PLM)-system enabling enterprise knowledge management using lifecycle ontological engineering [29].

The paper is organized in the following way. In Section II, we consider the transition from descriptive ontological systems to formal granular ontologies. First of all, in subsection A we present our modification of Guarino's hierarchical system of ontologies that enables top-down ontology engineering. Here, the concepts of meta-ontology

and upper ontology are of primary concern. To clarify the first concept, some principles of meta-logic and meta-mathematics are considered, and the distinctions between meta-ontology and upper ontology are discussed.

Sub-section *B* is devoted to a presentation of various methodologies for constructing formal ontologies. The main emphasis is made on discussing and comparing basic approaches suggested by representative of both classical philosophical-mathematical and novel computer science communities. Furthermore, the usefulness of abstract algebraic approach to ontological modeling is shown, and the perspectives of Maltsev’s algebraic systems in ontological modeling are outlined.

In sub-section *C*, we consider some basic principles of information granulation, specify the concepts of granule and granularity, give the classification and interpretation of granules and mention fundamental components of ontology granulation. In Sub-section *D*, a formal granular meta-ontology together with fuzzy and linguistic ontologies (based on extended linguistic variables) are introduced.

Section 3 is devoted to applications of granular meta-ontologies to lifecycle modeling. In sub-section *A*, the crucial role of lifecycle ontological modeling and engineering for developing intelligent product lifecycle management systems. In this context, two basic time metaphors and time theories are briefly analyzed. In sub-section *B*, circular lifecycle representations as coarse-granular ontologies are given. In sub-section *C*, formal lifecycle models are proposed, and sequential lifecycle ontologies based on Allen’s relations and their extensions are constructed.

II. FROM DESCRIPTIVE ONTOLOGICAL SYSTEMS TO FORMAL GRANULAR ONTOLOGIES

A. On a Hierarchical System of Ontologies

In this paper, we consider ontological modeling and engineering as a generic tool of knowledge management in multi-agent systems and intelligent organizations. Ontological modeling encompasses both cognitive and communicative sides of knowledge management. On one hand, ontological investigations deal with the problems of knowledge generation on the basis of some entities. On the other hand, their objective is to support communication processes, i.e., enable knowledge sharing and reuse.

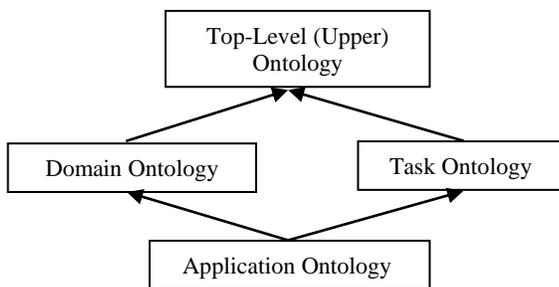


Figure 1. Guarino’s Types of Ontologies

It is often difficult or even impossible to construct single, comprehensive, coherent and practically useful ontology. To simplify ontology development, integration and reuse, a modular ontological engineering technique is suitable. In these cases, it is possible to exploit the principle «divide and rule»: decompose the subject domain into sub-domains with consequent composition, introduce upper ontologies, tasks ontologies and application ontologies [1] (see Fig. 2).

Here, domain ontology (for instance, maintenance ontology) encompasses domain concepts together with basic relations between them, whereas task ontology is related to main tasks or actions. For example, in maintenance we deal with such tasks as inspection, disassembly, substitution, assembly, diagnostics. The position of application ontology may be different. On one hand, application area specifies main requirements to domain and task ontologies. On the other hand, application ontology (for instance, aircraft maintenance) describes concepts depending both on a particular domain and tasks, which are often have specializations of *both* the related ontologies. This concept often corresponds to roles, played by domain entities while performing a specific activity. All these three types of ontologies may be referred to as low level ontologies, which are domain-dependent.

An upper ontology (or a top-level ontology) [1], [2] is a model of the common objects (or concepts) that are generally applicable across a wide range of domain ontologies. Such ontologies describe very general concepts like space, time, matter, resource, event, action, etc., which are independent of a particular problem or domain. It seems quite reasonable to start enabling mutual understanding and joint work of agents by specifying basic upper ontologies.

Nevertheless, this classical Guarino’s ontological system [1] as depicted in Fig. 1 corresponds to a bottom-up approach in ontological engineering.

Below we present our three-leveled hierarchy of ontologies that includes low-level ontologies, upper ontologies and meta-ontology (Fig. 2). It illustrates a top-down approach to ontological engineering. Here, the concept of Meta-Ontology is of primary concern. Unfortunately, the concepts of meta-ontology and upper ontology are often confounded.

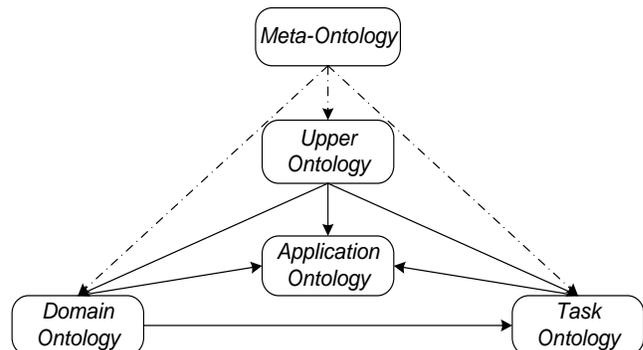


Figure 2. Representation of ontological hierarchy

To explain the concept of meta-ontology, we can use the analogy with meta-logic and meta-mathematics. About a hundred years ago, Russian scientist N.A. Vasiliev (see [3]) suggested a two-leveled logical structure, the sort of simple hierarchy of empirical logic and meta-logic. He considered human everyday logic as dual, semi-empirical, semi-rational area, and by contrast he envisaged a sort of generalized logic, called meta-logic. According to Vasiliev, we ought to make difference between two levels of knowledge: 1) an empirical level, based on real-world's events; 2) a conceptual level, depending on our thinking.

In his turn, S.C. Kleene defined meta-mathematics as the study of intrinsic capabilities of our formal models by mathematical methods.

Similarly, a meta-ontology (ontology of ontologies) studies the nature of ontological problems and specifies a family of ontologies. It provides the investigation of basic ontology properties and establishes the links between various ontologies. It means, that meta-ontology framework includes a family of mathematical theories, used for formal ontology description. In particular, it contains methods and forms of representing, integrating and merging different ontologies.

Below we shall present the concept of granular meta-ontology on the basis of Mal'tsev's algebraic system [4]. At first, we should discuss the notion of formal ontology.

B. Formal Ontologies

The idea of formal ontology was coined by E. Husserl who differentiated between formal logic on one hand and formal ontology on the other. In his «Logical investigations» E. Husserl has shown, that formal logic deals with the interconnections of truths (or propositional meanings in general) – with consequence relations, inference, consistency, proof, validity, whereas formal ontology considers the interconnections of things, objects and properties, parts and wholes, relations and collectives [5]. Husserl's formal ontology stands on three pillars: mereology, topology and the theory of dependence. Later on this idea was developed by St. Leśniewski – the author of three nested formal systems called respectively protothetic, ontology and mereology [6] and B. Smith [7], who suggested the concept of mereotopology as a theory of parts and boundaries.

Another classical approach rises to Tarski's concept of logical notion [8]. To define it, Tarski drew on Klein's Erlangen Programme that classified the various types of geometry (Euclidean geometry, affine geometry, etc.) by the type of one-one transformation of space onto itself that left the objects of this geometrical theory invariant. Generalizing, he specified the concept of logical notion by considering all possible one-to-one transformations (automorphisms) of a domain onto itself.

In the context of ontological engineering a basic proposal is to interpret model-theoretic languages (abstract logics) as formal ontologies [9]. Here, an abstract logic

consists of: (1) a collection of structures closed under isomorphism, (2) a collection of formal expressions, and (3) a relation of satisfaction between the two [10].

In modern computer science and artificial intelligence the term «ontology» stands for clear and formal specification of problem domain structure. According to T.R. Gruber, it is often referred to as an explicit specification of shared conceptualization [11]. In other words, ontology is a conceptualization of a domain into a human understandable, but machine-readable format. A formal definition of conceptualization has been given in [12]. It is a structure $CON_1 = \langle D, R \rangle$, where D is a domain and R is a set or relevant relations on D .

In his turn, N. Guarino [13] defines ontology as a logical theory that specifies some conceptualization; it includes some basic terms forming taxonomy, their definitions and attributes, related axioms and inference rules. In other words, ontology is seen both as a formal view to semantics and a thesaurus used by logical theory. In essence, any ontology expresses some convention about shared methods of constructing and using conceptual models. It can play the role of knowledge representation and reuse method, knowledge management tool, learning technique, etc.

He also focuses on the need of transition from extensional to intensional relations. A standard way to represent intensions (and therefore conceptual relations) is to see them as functions from possible worlds into sets. While ordinary relations are defined on a certain domain, conceptual relations are defined on a domain space – a structure $\langle D, W \rangle$, where D is a domain and W is a set of possible worlds. Then a conceptual relation ρ^n of arity n on $\langle D, W \rangle$ is given as a function $\rho^n: W \rightarrow 2^{D^n}$ from W into the set of all n -ary (ordinary) relations on D .

Thus, a conceptualization for D can be now defined as an ordered triple $CON_2 = \langle D, W, P \rangle$ [13], where P is a set of conceptual relations on the domain space $\langle D, W \rangle$.

Nowadays, some generalized approaches to building formal ontologies based on universal algebras and category theory are worth noticing [14]. Moreover, a new ontological framework Basic Formal Ontology (BFO) [15] that consists in a series of sub-ontologies at different levels of granularity is of special concern. Here, the basic concepts are subdivided into Continuants (e.g., Objects and Functions) and Occurents (e.g. Processes and Events).

To construct granular ontology (and meta-ontology) both, new formal ontological models and fundamentals of information granulation theory, ought to be developed.

C. Information Granulation and Granular Ontologies

Let us consider some basic principles of information granulation in the context of creating granular meta-ontology. The main concepts of granulation theory are granules, granularities, hierarchies, levels, granular structures and theirs mappings.

Information granulation is a basic capacity of cognitive agent that supposes processing information on such level of

abstraction, which is consistent with the allowable level of imprecision. The term “granule” is originated from Latin word “granum”, that means grain, to denote a small particle in the real or imaginary world. Typical interpretations of granules are: part of the whole, sub-problem of the problem, cluster, variable constraint, uncertainty area, etc.

According to L. Zadeh, granule is seen as a collection of objects which are drawn together by indistinguishability equivalence, similarity or functionality [16]. Information granules are complex dynamic information entities which are formed to achieve some goal. The arrival of information granulation means the transition from ordinary machine-centric to human-centric approach in information gathering and knowledge discovery [17]. By selecting different granulation levels one can construct heterogeneous ontological models with modifiable abstraction degrees.

It is easy to clarify the sense of the term “granular” by comparing it with the antonym “singular”. For example, one of the founders of multi-valued logics J. Lukasiewicz specified the basic concepts of Truth and Falsity as singular objects, whereas Zadeh’s consideration of linguistic variable Truth with such linguistic hedges as «more or less true», «rather true than false» and so on supposes the shift from singular to granular truth values.

The same idea underlies granular ontology: the transition from singular (pointwise) representation primitives to interval and regional representation primitives is the essence of ontological granulation. For example, the transition from fine-grained low-level ontology, given by an ordinary graph, to coarse-grained upper ontology, given by a hyper-graph, may be fulfilled.

Some classical approaches to ontological granulation are presented in [18, 19].

Now let us discuss some basic components of ontology granulation theory. These are: 1) ontology granulation principles and criteria; 2) interpretation and classification of granules; 3) approaches and methods of granulation; 4) formal models of ontological granules; 5) Ontological granular structures; 6) mappings of granular structures; transitions from fine-grained to coarse-grained ontologies and vice versa; 7) quantitative indices for granular ontologies and granulation process itself.

D. Formal Granular Meta-Ontologies.

From the systemic viewpoint, meta-ontology makes appeal to the most universal domain-independent categories, such as *concepts, relations, changes*. The timely adaptation for changes and the management of these changes characterizes a dynamic meta-ontology.

A natural mathematical basis for specifying meta-ontologies is Maltsev’s theory of algebraic systems [4]. Below we shall recall the concept of algebraic system and extend it to take into account granularity and fuzziness.

Definition 1 [4]. An algebraic system is a triple

$$AS = \langle X, O, \Pi \rangle, \quad (1)$$

where X is a non-empty set of objects, called the underlying set (or basis) of the algebraic system, O is an operation set, i.e., the set of finitary operations on X and Π is predicate set. Here, $O = \{o_i^j\}$, $i = 1, \dots, m$, $j = 0, \dots, n$, $o_i: X \rightarrow X$, $o_i^2: X \times X \rightarrow X, \dots, o_i^n: X^n \rightarrow X$. Constants are also included into O as 0-ary functions. $\Pi = \{\pi_k^l\}$, $k=1, \dots, p$, $l=1, \dots, q$, $\pi_k: X \rightarrow \{0,1\}$, $\pi_k^2: X \times X \rightarrow \{0,1\}, \dots, \pi_k^q: X^q \rightarrow \{0,1\}$.

The union of operation set and predicate set $O \cup \Pi$ is called a signature (type) of algebraic system. The algebraic systems with coinciding signatures have the same type.

The algebraic system can have multiple basis, for instance, $X = (X_1, \dots, X_n)$. When underlying set, provided with the structure of topological space and operations, are continuous, we obtain topological algebraic system. Various topological spaces may be represented as sets equipped with closure operation.

If the operation set may include partial operations, then algebraic system is called partial algebraic system.

Two special cases of algebraic system are: a relational system (model) for $O = \emptyset$ and a universal algebra for $\Pi = \emptyset$.

Remark 1. In case of fuzzy relations, the concepts of relation and predicate coincide.

Definition 1*. A Meta-Ontology is given by an algebraic system

$$MONT = \langle C, R, O \rangle, \quad (1^*)$$

where C is a non-empty set of concepts, R is a set of relations on C and O is a set of operations over concepts and/or relations.

Remark 2. It is worth noticing, that formal specification of meta-ontology (ontology) by relational system is not sufficient for ontology integration and investigation. The intersection of various concept sets allows us to specify the kernel of ontology (degree of sharing), their union gives its range and difference helps to compare ontologies. Moreover, specific operations of generalization and specialization provide the changes of ontology granularity.

Definition 2. Let C be a non-empty set of concepts. We call a conceptual granule any subset $g \in 2^C$, where 2^C is a power set of C .

Definition 3. For any two conceptual granules $g, g' \in 2^C$, if $g \subseteq g'$, then g is called a sub-granule of g' , and, in its turn, g' is a super-granule for g .

Definition 4. Let us denote $G \in 2^C$ a non-empty set of conceptual granules. Then a pair $GS = \langle G, \subseteq \rangle$ is called a granular structure if \subseteq is set inclusion.

Now, let us give an unfolded definition of granular meta-ontology [20].

Definition 5. A granular meta-ontology is a quadruple

$$GMONT = \langle C, G^C, R_G, O_G \rangle, \quad (2)$$

where C is a non-empty set of concepts, G^C is a basis of ontological granulation, R_G is a set of granular relations on C_G , O_G is a set of operations over C_G . and/ or R_G . Among main bases of ontological granulation we take the ordinary power set 2^C , the set of fuzzy subsets $[0,1]^C$, the set of sub-lattices L^C of a lattice L .

Typical ways of specifying granular sets of concepts are the following: 1) a set of concepts C together with a quotient set C/E , denoted by $C_{G1}=(C, C/E)$, where E is an equivalence relation; 2) a set of concepts C with a family of nested sets $F = \{A_0, \dots, A_n\}$, $C_{G2} = (C, F)$, $F = \{A_0, \dots, A_n\}$, where $A_i \subseteq C$, $i=0, \dots, n$, $A_0 = X$, $A_0 \supseteq A_1 \supseteq \dots \supseteq A_n$ or more generally as a set of α -cuts defined on the lattice L , $A_\alpha: L \rightarrow 2^X$, $\alpha \in L$; 3) a set C with a family of fuzzy subsets $[0,1]^C$, $C_{G3} = (C, [0,1]^C)$. 4) a universal set C together with a rough set, given by lower and upper approximation.

A special granular computing view of an ontology, based on rough set methodology, is developed in [21].

A good example of granular ontology is fuzzy ontology, where fuzzy concepts and/or fuzzy relations and/or fuzzy attributes are considered. Two definitions of lightweight and heavyweight fuzzy ontologies are given below

Definition 6. A fuzzy ontology is a quadruple

$$FONT = \langle I, C_F, H, R_F \rangle, \quad (3)$$

where I is the set of individuals (instances of concepts), C_F is the set of fuzzy concepts, H is the hierarchy, R_F is the family of fuzzy relations sets.

Definition 7. A completely fuzzy ontology is a quintuple

$$FONT = \langle I, C_F, R_F^k, O_F^j, AX \rangle, \quad (4)$$

where I is the set of individuals (instances of concepts), C_F is the set of fuzzy concepts, R_F^k is the family of fuzzy relations sets, $k = 1, 2, \dots, s$; O_F^j is the set of finite operations over fuzzy concepts and/or fuzzy relations, $j=0, \dots, n$, AX is the set of axioms.

Fuzzy ontologies were already extensively studied (see, for instance, [22]) On the contrary, the specification of ontologies on the basis of Zadeh's linguistic variable remains a rather rare case. Below a fuzzy linguistic ontology is introduced on the basis of extended linguistic variable.

Definition 8 [20]. An extended linguistic variable is given by a tuple

$$LV_{ex} = \langle L, T, U, G, M, R_T, R_U, O_g, TR_U \rangle, \quad (5)$$

where L is the name of linguistic variable, T is its term set, U is the universal set (numerical scale), G is the set of syntactic rules (grammar), M is the set of semantic rules, R_T is the set of relations on T , R_U is the set of relations on U , O_g is the set of granulation operations, TR_U is the set of universe transformations.

Definition 9. A fuzzy linguistic ontology based on extended linguistic variable is a tuple

$$LVONT = \langle I, C_A, C_F, R, U, [0,1]^U, R_F \rangle, \quad (6)$$

where I is the set of individuals (instances of concept), $C_A = \{c_A\}$ is an abstract concept (singleton) that corresponds to the name of linguistic variable, C_F is the set of fuzzy concepts (the term set of linguistic variable), $R = \{r \mid r \subseteq C_F \times C_F\}$ is the set of binary relations between fuzzy concepts. Let us note, that the strict order relation $<$ is of special concern. The pair $\langle C_F, < \rangle$ generates an ordered structure. Here, U is the universal set, $[0,1]^U$ is the set of fuzzy subsets on U , R_F is the set of fuzzy relations on $[0,1]^U$.

III. ON THE USE OF GRANULAR META-ONTOLOGY: LIFECYCLE UPPER ONTOLOGIES

A. The Role of Lifecycle Modeling for Intelligent PLM

At present, the concept of ontology development lifecycle is thoroughly studied (see, for instance, [23], [24], [29]), but the problems of system's lifecycle ontology and lifecycle ontological modeling still remain open.

The aim of cyclic product definition is to realize both products and processes, and economic solutions that are better and more intelligent by integrating lifecycle philosophy into technology and economy.

The lifecycle concept may be analyzed from various viewpoints; different variants of specifying its phases and activities were suggested. In marketing theory products follow such stages as introduction, growth, maturity, and decline. In industry, all products or systems have a particular life span considered as a sequence of stages, which is called product lifecycle (or complex system lifecycle).

The term «system's lifecycle» expresses the idea of a circulation of produced artifacts between the fields of design, production and usage (consumption). One of fundamental resources for lifecycle management is time. Any cycle as a whole is characterized by the presence of finite and repetitive parts on some temporal intervals; here, key parameters are durations.

Nowadays, Product Lifecycle Management is viewed as a basic manufacturing strategy for XXIst century [25]. It is deployed as a process of managing the entire lifecycle of a product (system) from its conception, through design and manufacture, to service, disposal and dismantling. An implementation of PLM-system means integration of data, processes, personnel and organizations to provide a product information backbone for modern computer integrated (in particular, virtual and extended) enterprises.

We point out, that PLM initiative considers both questions: «how an enterprise works» and «what is being created». An effective PLM improves the ability of manufacturing enterprise to make better and faster product-related decisions. It enables the formation of a consistent set of concerted industrial solutions that support the collaborative creation, management, dissemination and use of product-definition information [26], [31].

It seems quite reasonable, that an advanced PLM-system has to support various lifecycle representations. Nevertheless, even the most popular industrial PLM-systems like Teamcenter Enterprise or Agile 9 lack this capacity and are in fact product data management systems. In our opinion, this situation is mainly explained by the absence of lifecycle ontological subsystem. The creation of intelligent PLM-system supposes the development of lifecycle engineering methods based on lifecycle ontologies [30].

B. Time Metaphors and Theories for Lifecycles

Basic time theories should be envisaged in the context of lifecycle modeling: substantial and relational, static and dynamic, pointwise and interval time. Two well-known time metaphors – «time wheel» and «time arrow» – bring about lifecycle’s circular and sequential models respectively. On one hand, sequential linear models express such time properties as course, ordering facility, irreversibility. On the other hand, circular time models make emphasis on alternations, reiterations, rhythms, self-sustaining processes. We shall support such a pluralism of lifecycle ontologies by constructing and analyzing both circular and sequential representations.

A specific lifecycle feature is its heterochronous character, i.e., irregularity related to the different vision of temporal criteria and constraints on various stages [27]. In fact, we try both to accelerate design and manufacturing time and slow down usage time. For instance, during the design stage a basic criterion is to decrease design time, e.g. by using concurrent design strategies. Contrarily, on the usage stage we tend to keep or increase reglamentary period, for example, by improving maintenance system.

C. Circular Lifecycle Representations: Coarse-Grained Ontologies

In case of system’s lifecycle, two basic granule types are lifecycle stages and phases. Lifecycle stages are usually divided into lifecycle phases, where each phase corresponds to a specific system’s state. So, the stage is viewed as a coarse-grained lifecycle part, whereas the phase is a fine-grained part.

At first, we shall represent lifecycle stages in the framework of set-theoretic approach as granules obtained by partition. Let us introduce natural denotations for systems’ lifecycle stages: D – design; M – manufacturing; U – use; R – recycling. Then we have

$$LC_1 = D \cup M \cup U, D \cap M = \emptyset, M \cap U = \emptyset, U \cap D = \emptyset \quad (7)$$

$$\text{or } LC_2 = M \cup U \cup R, M \cap U = \emptyset, U \cap R = \emptyset, R \cap M = \emptyset \quad (7^*)$$

Here, the structure of LC_2 (7*) expresses the «ecological imperative» of modern manufacturing being tightly related to Kimura’s lifecycle inversion concept [27]. The first lifecycle partition LC_1 (6) may be depicted by sectors of the circle (Fig. 3).

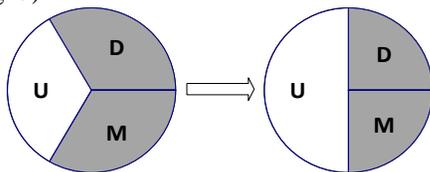


Figure 3. A Circular Representation of System’s Lifecycle: an Illustration of Reducing Lead (Design and Manufacturing) Time and Increasing Period of Usage

It is worth noticing that the representation of lifecycle by ordinary partition is rather simplistic and does not express many existing interrelations and co-operation links between partially overlapping stages. Moreover, this simultaneous

work enables very important functions. For example, on the crossroad of usage and design system’s specification is made, production technologies ought to be discussed on the edge of design and manufacturing, whereas maintenance requires the collaboration of users and manufacturers. With taking into consideration such factors, we obtain the circular lifecycle model with fuzzy boundaries. For these cases lifecycle granulation is based on covering (Fig. 4). Here,

$$LC_1 = D \cup M \cup U, \text{ but } D \cap M \neq \emptyset, M \cap U \neq \emptyset, U \cap D \neq \emptyset \quad (4)$$

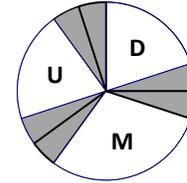


Figure 4. A Circular Lifecycle Representation on the Basis of Covering: the Presence of Collaborative Works at All Stages

D. Sequential and Parallel- Sequential Lifecycle Representation : Fine-Grained Ontologies Based on Extended Allen’s Relations

Generally our approach is based on relational time model and interval time primitives. We use a fuzzy extension of well-known Allen’s temporal logic [28] to model the links between lifecycle phases (or lifecycle stages and phases). These are mainly two types of relations: consequence and overlapping relations (see Table I).

Let us recall, that fuzzy quantity is defined as a fuzzy set of the real line. Fuzzy quantities are more suitable to describe flexible requirements on lifecycle parts duration.

We introduce a formal model of lifecycle ontologies ONT_{LC} as a quadruple

$$ONT_{LC} = \langle C_{LC}, R_{LC}, \Omega_{LC}, T_{LC} \rangle, \quad (8)$$

where C_{LC} is the set of concepts related to lifecycle, R_{LC} is the set of relations between these concepts, Ω_{LC} is the set of operations over concepts and/or relations, T_{LC} is the set of temporal characteristics for lifecycle.

Basic concepts for lifecycle are its phases and stages; therefore, the triple below can be taken as lifecycle systemic kernel

$$ONTS = \langle S, R_s, O_s \rangle, \quad (9)$$

where S is the set of lifecycle stages (phases), R_s is the set of relations between these stages (phases), O_s is the set of operations used on these stages (phases).

It is worth noticing that each lifecycle phase may be seen as an interval primitive $s = [a^-, a^+]$, where a^- is the starting point and a^+ is the end point of the interval. A fuzzy interval extending the concept of an interval is a special kind of fuzzy quantity that is represented by a convex fuzzy subset of a real line. As a special case, we have

$$ONT_{S1} = \langle S, <_f, \approx_f \rangle, \quad (10)$$

where $<_f$ is a fuzzy strict linear order relation that is non-reflexive, asymmetric, transitive and linear, \approx_f is a fuzzy simultaneity relation, i.e., fuzzy reflexive, symmetric relation.

More generally, we can use the linguistic variable «Time» with a linguistically ordered term set such as {almost simultaneously, a bit later, later, much later, very much later}.

A general representation of lifecycle ontology can be depicted by a mind map (Fig. 5). Here, such ontology characteristics as its goal, role, language, representation form and basic relations are of special concern.

IV. CONCLUSION

In this paper, some links between two different scientific areas called “Ontology” have been established through formal granular ontologies. The concept of granular meta-ontology has been discussed, formal models of granular, fuzzy and linguistic ontologies have been developed. An application of granular meta-ontology and extended Allen’s logic to system’s lifecycle ontological engineering has been considered. Our future work will be focused on specifying basic indices for granular ontologies and developing an ontological sub-system for intelligent PLM-system.

ACKNOWLEDGMENT

This work has been supported by Russian Fund for Basic Research (Project No 14-07-00846), Ministry of Education and Science of Kazakhstan (Project 0115 RK 00532) and DAAD and Russian Ministry of Education (Project HM3754).

REFERENCES

- [1] N. Guarino, “Formal ontology and information systems”, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS’98, Trento, Italy, June 6-8, 1998, Ed. by: N. Guarino, Amsterdam, IOS Press, 1998, pp. 3-15.
- [2] J.F. Sowa, “Top-level ontological categories”, International Journal of Human-Computer Studies, vol.43, No.5-6, 1995, pp. 669-685.
- [3] G. Priest, “N.A. Vasiliev and imaginary logic”, History and Philosophy of Logic, vol.21, No.1, 2000, pp. 135-146.
- [4] A.I. Mal'tsev, "Algebraic Systems", Berlin: Springer, 1973.
- [5] E. Husserl, “Logical Investigations”, London: Routledge and Kegan Paul, 1970.
- [6] St. Leśniewski, “Collected Works”, ed. by S.J. Surma, J.T.J. Szrednicki, D.I. Barnett, Dordrecht: PWN - Kluwer, vol.1 and 2, 1992.
- [7] B. Smith, “Mereotopology: a theory of parts and boundaries”, Data and Knowledge Engineering, vol.20, 1996, pp. 287-303.
- [8] A. Tarski, “What are logical notions?”, History and Philosophy of Logic, vol.7, 1986, pp. 143-154.
- [9] E. Dragalina-Chernaya, “Abstract Logics as Formal Ontologies of Relations”, L'ontologie de la relation, Caen: l'Université de Caen, 2012.
- [10] J. Barwise, “Model-theoretic logic: background and aims”, Model-Theoretic Logic, ed. by J. Barwise. and S.Feferman, New York, 1985, pp. 3-23.
- [11] T.R. Gruber, “A Translation approach to portable ontologies”, Knowledge Acquisition, vol.5, No.2, 1993, pp. 199-220.
- [12] M.R. Genesereth and N.J. Nilsson, “Logical Foundation of Artificial Intelligence”, Los Altos, Ca: Morgan Kaufmann, 1987.
- [13] N. Guarino, “Formal Ontology, Conceptual Analysis and Knowledge Representation”, International Journal of Human-Computer Studies, vol.43, No.5-6, 1995, pp. 625-640.
- [14] M. Johnson and R. Rosenbrugh, “Ontology engineering, universal algebra and category theory”, Theory and Applications of Ontology: Computer Applications, ed. by R.Poli et al., Heidelberg: Springer, 2010, pp. 565-576.
- [15] R. Arp, B. Smith and A.D. Spear, “Building Ontologies with Basic Formal Ontology”, Cambridge, MA: MIT Press, 2015.
- [16] L. Zadeh, ”Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic”, Fuzzy Sets and Systems, vol.90, 1997, pp. 111-127.
- [17] A. Bargiela and W. Pedrycz, “Granular Computing: an Introduction”, Dordrecht: Kluwer Academic Publishers, 2003.
- [18] T. Bittner and B. Smith, “Vagueness and granular partition”, Formal Ontology in Information Systems, ed. by C.Welty, B.Smith, New York: Sheridan Press, 2001.
- [19] T. Bittner and B. Smith, “Granular spatio-temporal ontologies”, Proceedings of the AAAI Spring Symposium on Foundations and Applications of Spatio-Temporal Reasoning (FASTR), Menlo Park, CA: AAAI Press, 2003, pp. 12-17.
- [20] V.B. Tarassov, A.P. Kalutskaya and M.N. Svyatkina, “Granular, fuzzy and linguistic ontologies to enable mutual understanding between cognitive agents”, Proceedings of the 2nd International Conference on Open Semantic Technologies for Intelligent Systems, OSTIS-2012, Minsk, Belarus, February 16-18, 2012, Minsk: BSUIR Press, 2012, pp. 267-278.
- [21] S. Calegari and D. Ciucci, “Granular computing applied to ontologies”, International Journal of Approximate Reasoning, vol. 51, 2010, pp. 391-409.
- [22] S. Calegari and D. Ciucci, “Towards a fuzzy ontology definition and a fuzzy extension of an ontology editor”, Enterprise Information Systems. Lecture Notes in Business Information Processing, vol.3, Heidelberg: Springer, 2008, pp. 147-158.
- [23] A. Gomez-Perez, M. Fernandez-Lopez and O. Corcho, “Ontological Engineering”, Heidelberg: Springer, 2005.
- [24] A. Borisov, G. Kulshova and T. Zmanovska, “Introduction to Ontology Engineering”, Riga: RTU Press, 2014.
- [25] J. Stark, “Product Lifecycle Management: 21st Century Paradigm for Product Realization”, 2nd ed., London: Springer, 2011.
- [26] A. Smirnov and N. Shilov, “Ontology Matching in Collaborative Recommendation System for PLM”, International Journal of Product Lifecycle Management, vol.6, No.4, 2013, pp. 322-338.
- [27] A. Fedotova and V. Tarassov, ”Development and interpretation of spiral lifecycle’s model: a granular computing approach, part 1. Lifecycle granulation and spiral representation”, Proc. of the 7th International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, ICSCCW’2013, Izmir, Turkey), Kaufering: b-Quadrat Verlag, 2013, pp. 432-440.
- [28] J.F. Allen, “Maintaining knowledge about temporal intervals”, Communications of the ACM, vol.26, 1983, pp. 832-843.
- [29] R. Stark and A. Pfortner, “Integrating ontology into PLM-tools to improve sustainable product development”, CIRP Annals - Manufacturing Technology, vol. 64, Issue 1, 2015, pp. 157-160.
- [30] R. Stark, H. Grosser, B. Beckmann-Dobrev and S. Kind, “Advanced Technologies in Life Cycle Engineering”, Proc. of the 3rd International Conference in Through-life Engineering Services. Cranfield, UK: Elsevier, 2014 pp. 3-14.
- [31] R. Stark, F.-L. Krause, C. Kind, U. Rothenburg, P. Müller, H. Hayka and H. Stöckert, “Competing in Engineering Design – the Role of Virtual Product Creation”, Journal of Manufacturing Science and Technology Manuscript, Elsevier Verlag, 2010 pp. 157-184.

TABLE I MAIN TEMPORAL RELATIONS BETWEEN LIFECYCLE PHASES AND STAGES: A CRISP MODEL

Notation	Relations and Their Inversion	Illustrations	Examples
r ₁	Phase <i>a</i> is performed before (precedes) phase <i>b</i>		Detailed Design phase precedes Maintenance phase
r ₂	Phase <i>b</i> is performed later (follows) phase <i>a</i>		Maintenance phase follows Detailed Design phase
r ₃	Phase <i>a</i> immediately precedes (is adjacent to) phase <i>b</i>		Preliminary Design phase immediately precedes Basic Design phase
r ₄	Phase <i>b</i> immediately follows phase <i>a</i>		Basic Design phase immediately follows Preliminary Design phase
r ₅	Phase <i>a</i> partially overlaps with phase <i>b</i>		Detailed Design phase partly overlaps with Production Planning phase
r ₆	Phase <i>b</i> partially overlaps with phase <i>a</i>		Production Planning phase partly overlaps with Detailed Design phase
r ₇	Phase <i>a</i> lies inside stage <i>s</i>		Maintenance phase lies inside Usage stage
r ₈	Stage <i>s</i> comprises phase <i>a</i>		Usage stage comprises Maintenance phase
r ₉	Phase <i>a</i> lies inside stage <i>s</i> , so that their starting points coincide		Production Specification phase lies inside Development stage, so that their starting points coincide
r ₁₀	Stage <i>s</i> comprises phase <i>a</i> , so that their starting points coincide		Development stage comprises Production Specification phase, so that their starting points coincide
r ₁₁	Phase <i>a</i> lies inside stage <i>s</i> , so that their endpoints coincide		Removal from Usage phase lies inside Usage stage, so that their endpoints coincide
r ₁₂	Stage <i>s</i> comprises phase <i>a</i> , so that their endpoints coincide		Usage stage comprises Removal from Usage phase, so that their endpoints coincide
r ₁₃	Phase <i>a</i> coincides with phase <i>b</i>		Detailed Design stage coincides with Basic Design and Work phase

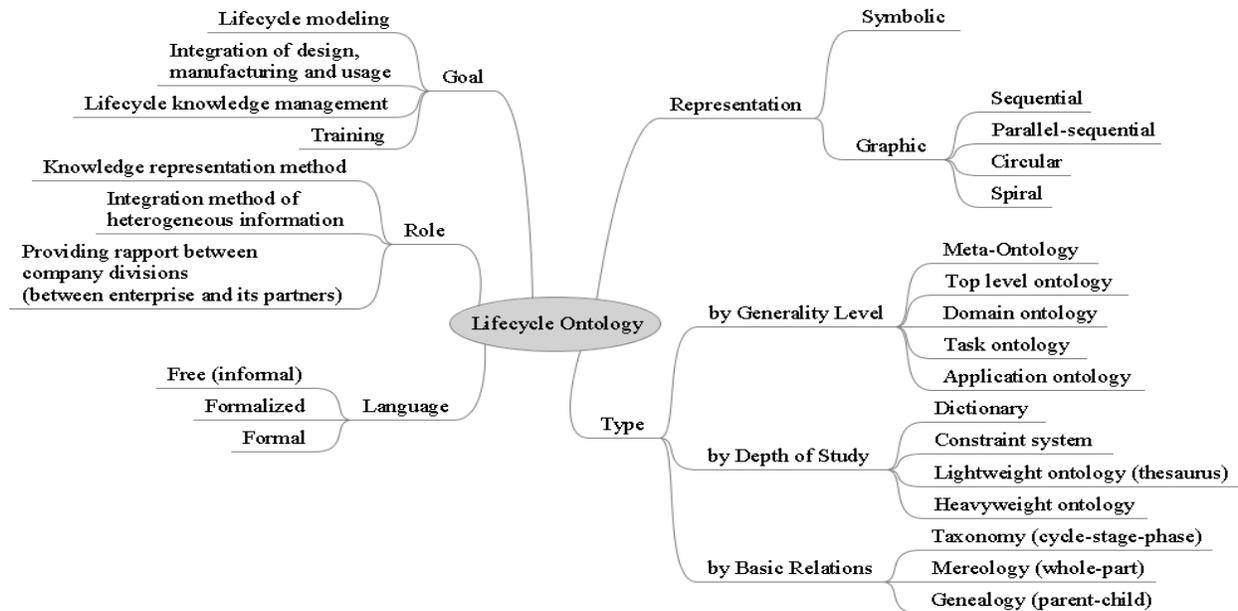


Figure 5. General Lifecycle Representation by a Mind Map

A Lightweight Simulator for Autonomous Driving Motion Planning Development

Tianyu Gu

Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA
Email: tianyu@cmu.edu

John M. Dolan

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
Email: jmd@cs.cmu.edu

Abstract—A good simulation environment will facilitate motion planning algorithm development for urban autonomous driving. The first requirement of such a simulator is to be able to replicate a complex urban environment, including road network, curb, general objects, etc. The second requirement is to simulate a realistic host vehicle, which includes perception, control and vehicle dynamics, to recreate imperfect inputs and non-accurate execution of the planner. The third requirement is to model traffic participants (other on-road vehicles) for microscopic traffic simulation. Intelligent-agent-based techniques are used to allow the traffic participants to interact with the environment and each other. In this paper, we present an open-source lightweight simulation environment, FastSim, which is designed to meet the three requirements above.

Keywords—Traffic simulator; Intelligent agent; Motion Planning; Autonomous driving

I. INTRODUCTION

Autonomous passenger vehicles (APV) have demonstrated promising social impacts that touch nearly all aspects of modern transportation. Motion planning (MP) algorithms are one of the most important components in such autonomous systems. The development of a planning algorithm is typically first performed in a simulator before applying it on the actual robot for convenience and safety reasons. Many simulation environments are either too complex or overly-simplified [1][2][3], and only a few are freely distributed for community usage. In this paper, we present the development of a lightweight and real-time simulation environment designed specifically for quick motion planning algorithm prototyping in urban environments, where the host vehicle operates on roads or freeways with structured lane information.

A. Related Work

MP algorithms take inputs from the upper-level perception processing module and generate outputs to the lower-level controller modules. From the planner's perspective, three factors must be reproduced in a simulator: the ground-truth of the surrounding environment, the perception of the ground-truth, and the host vehicle dynamics where the actual execution result of the planned actions is evaluated.

In terms of the environment ground-truth, Carnegie Mellon University Grand Challenge team [2] proposed a simulation package for desert vehicles with general object representations for field navigation, but no capability to model common objects in urban environments. The Tartan Racing Urban Challenge System (TRUCS) [3] explicitly represented different types of moving objects in urban environments, but was not capable of modeling the interactive capabilities of many objects. For a simulation environment that aimed at creating realistic microscopic traffic, [1] proposed a lane changing and merging model

for on-road vehicles. However, these models used simplified assumptions, which can only react to other in-lane vehicles, but not to other environment objects like static objects and pedestrians, etc.

Self-localization and sensing the surrounding environment are the two pillars of perception. Prior simulators typically assume perfect localization. In order to reproduce realistic imperfect localization, a closer look at actual localization methods used in reality and the nature of output dynamics is required. For environment sensing, the majority of prior simulation environments directly feed the motion planning algorithms with complete knowledge (directly pass the simulated ground-truth). However, on real robot, sensing is never perfect, e.g., [4] investigated the real-world perception failure cases with real Light Detection and Ranging (LIDAR)-based ranger. For MP to behave robustly on robot, imperfect sensing (e.g., sensor limitations) is important to simulate for MP development.

In terms of the host vehicle dynamics, there is a huge body of literature in vehicle modeling [5]. Many prior simulators used an overly simplified vehicle model. On the other hand, too-complicated vehicle models would be computationally unjustifiable. Meanwhile, lower-level vehicle controllers (e.g., path tracking and speed regulation) are external to the motion planner, hence must also be simulated, preferably with the actual controllers [6] implemented on the robot itself.

In the remainder of this paper, we explain the design of the proposed real-time simulation environment FastSim for MP algorithm development. The organization of this paper is as follows. Section II explains the implementation details of the proposed simulation environment FastSim. Section III presents the computationally efficient implementation of intelligent-agent-based microscopic traffic simulation. Section IV describes the user interface design. Section V concludes with our contributions and future work.

II. SIMULATION ENVIRONMENT

Based on the requirements from section I-A, the FastSim simulator consists of three simulation engines (Figure 1):

- The environment simulation engine models different invariant (e.g., road network, curb, etc.) and varying world elements (e.g., general static or moving objects).
- The perception simulation engine models the imperfect self-localization and environment sensing.
- The host vehicle simulation engine models the vehicle dynamics and low-level tracking controllers.

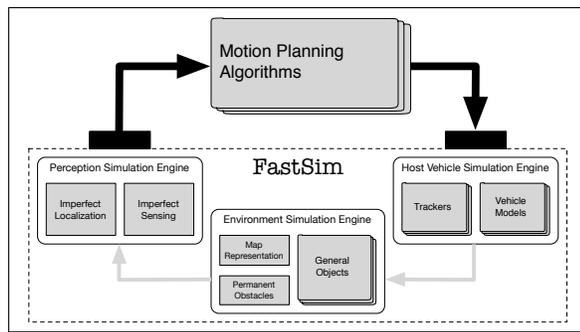


Figure 1. System diagram of FastSim.

A. Environment Simulation

1) *Road Network*: A road network provides the inter-connectivity of roads and roads' lane-level information that specifies the drivable regions. The route network definition file (RNDF) [3] is a robust way to store road network information. It makes use of road segments, where each segment contains one or more parallel lanes. Each lane is specified by a series of global way-points. Connectivity among lanes is defined by pairs of exit/entry way-points. One main limitation of the RNDF is its restriction to a fixed lane width and speed limit for each lane.

FastSim uses a similar waypoint-based lane definition. But for each waypoint, lane width (w) and speed limit (v_{lim}) are added to global position (x and y). An ever-increasing station coordinate (s) is first calculated for each waypoint by calculating piecewise-linear cumulative distance along-road. Cubic polynomial or step signal could be used for smooth or immediate interpolation:

$$\begin{cases} X(s) = \sum_{i=0}^3 p_X^i \cdot s^i \\ X(s) = X^i |_{s^i \leq s < s^{i+1}} \end{cases} \quad (1)$$

when X is global position (x, y), it is commonly interpolated by cubic polynomials. The first-order (heading θ) and second-order (curvature κ) geometric information is also easily obtained via analytic differentiation. When X is lane width (w) or speed limit (v_{lim}), either interpolation could be used according to the specific situation.

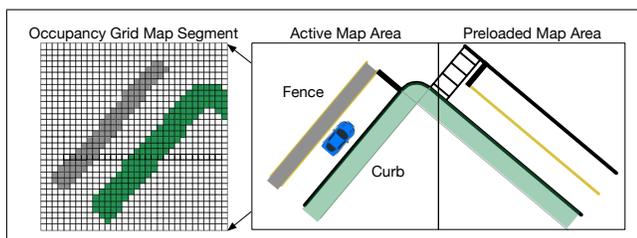


Figure 2. Permanent obstacles.

2) *Permanent Obstacles*: Permanent obstacles (Figure 2) refer to the stationary environment constraints that make certain regions non-traversable, such as curb and lane fences. Unlike general objects below, permanent obstacles typically do not have a separable shape. Hence, we use an occupancy

grid representation as an off-line map file, which is further rasterized at a larger scale to break a large area into smaller pieces. As the host vehicle moves, segments of occupancy grids maps in proximity to the vehicle are loaded.

3) *General Objects*: Various static and moving objects must be modeled in the urban environment. Objects of different types have different motion dynamics. The trivial non-movement model is for static objects (e.g., trash bins), which only contains unchanging pose information; a particle movement model is used for objects whose motion can be omnidirectional (e.g., pedestrians); the kinematic bicycle model can be used to model objects with non-holonomic kinematic constraints (e.g., bicyclists and other passenger vehicles).

As for the motion of objects, it is sometimes useful for other objects to follow a predetermined trajectory and not react to other simulated objects, for objects like the leading car in a queue of traffic, or a reckless pedestrian crossing the street disregarding traffic. In other cases, it is more important to create more realistic on-road traffic by enabling traffic participants with some intelligence to interact. Section III will explain more details on this matter.

B. Perception Simulation

In a realistic robot system, both localization and environment sensing have errors and limitations. One of the main design goals of the simulator is for it to be sophisticated enough to model such imperfect conditions for MP algorithm design purposes. The environment simulation engine above provides the "ground-truth", hence we need a separate perception simulation module to mimic realistic perception outcomes.

1) *Imperfect Localization*: The majority of localization methods are based on Extended-Kalman-Filter (EKF), Monte-Carlo-Filter (MCF) or Simultaneous-Localization-and-Mapping (SLAM) algorithms. They output best estimates of the vehicle state, along with covariance matrices describing the confidence of measurement. However, localization error in reality is largely situation-dependent, e.g., the vehicle loses GPS in an urban canyon or enters an area where the environment's features are quite different from the map. It is extremely difficult to model these realistic scenarios in a simulation environment. However, to make sure the perception outputs to planners are compatible with that of a real perception system, we add arbitrarily biased white-noise to the ground truth, and apply an EKF to maintain a filter-based perception output.

2) *Imperfect Object Sensing*: Ranger-based sensing units are the most commonly used on an APV. Two main sources of imperfect (partial) perception are the limited field of view (FOV) and occlusion. We simulate these limitations by the placement of virtual sensors at different configurations on the host vehicle (Figure 3). At each time-stamp, a constant-horizon line-tracing algorithm is used to simulate the sensor scanning, and only the objects that are reached by the simulated detection rays are made visible to the MP algorithm.

C. Host Vehicle Simulation

In a realistic robot system, the plan is never executed perfectly due to actuation errors and unmodeled vehicle dynamics. It is important for FastSim to simulate the execution of motion plans with adequately sophisticated host vehicle models for MP algorithm evaluation purposes.

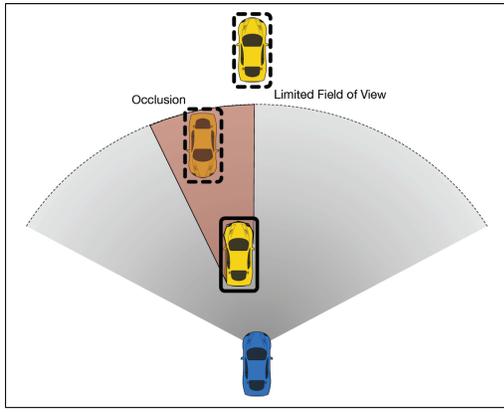


Figure 3. Imperfect sensing due to limited field of view and occlusion.

1) *Tracking Control*: In most APV systems, a decoupled planning and control scheme is used: the output of MP algorithms is fed to lower-level tracking controller and execution components. From the motion planner's perspective, the controller contributes partially to the overall vehicle dynamics. It is hence necessary to model the controllers. Two commonly used tracking controllers, a pure-pursuit and Linear Quadratic Regulator (LQR)-based trajectory tracker [6], are implemented.

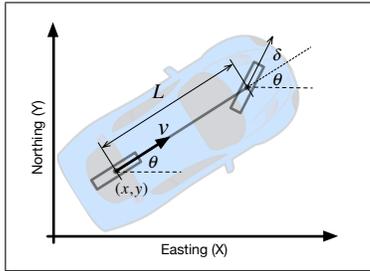


Figure 4. Dynamic bicycle model.

2) *Vehicle Dynamics*: A simplistic kinematic model has been used for many ground robot simulations [3]. While suitable for low-speed navigation applications, they cannot model realistic high-speed vehicle dynamics. Two of the most important factors are latency and vehicle skidding dynamics. Hence, we adopt a dynamic bicycle model (Figure 4):

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{\theta} = G_s \cdot \frac{v}{L} \cdot \tan \delta \\ \dot{\delta} = \frac{1}{T_\delta} \cdot (\delta_c(t) - \delta) \\ \dot{v} = a \\ \dot{a} = \frac{1}{T_a} \cdot (a_c(t) - a) \end{cases} \quad (2)$$

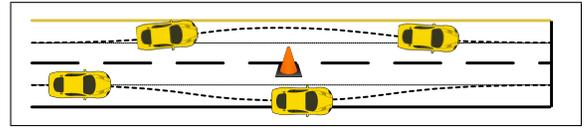
where $G_s \in [0, 1]$ is the slipping coefficient, T_δ and T_a are actuation latency coefficients of first-order low-pass filters, x , y and θ are the global pose, v is the speed scaler, δ and a are the actual steering/acceleration scalars, and δ_c and a_c are the commanded steering/acceleration (model inputs).

III. MODELING TRAFFIC WITH INTELLIGENT-AGENTS

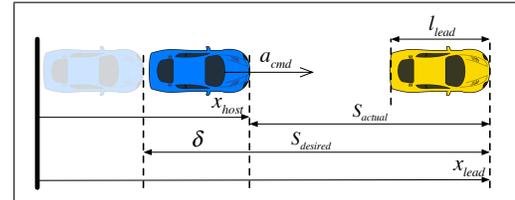
As explained in Section II-A3, it is sometimes important to simulate basic interactive intelligence of other traffic participants in order to recreate realistic traffic behavior. In this paper, we are primarily concerned with surrounding on-road vehicles, particularly, interested in modeling three basic maneuver capabilities:

- \mathcal{M}_1 : Swerve avoidance of static obstacles.
- \mathcal{M}_2 : Longitudinal avoidance of/distance keeping to a leading object.
- \mathcal{M}_3 : Lane-changing maneuver.

The challenge is to model these behaviors in a computationally efficient manner. It is natural to think of using a planner-based approach for each simulated on-road vehicle. However, this is generally not scalable if the number of on-road vehicles is large. In this section, we propose a computationally efficient interaction model for other on-road vehicles capable of performing the three maneuvers above.


 Figure 5. \mathcal{M}_1 : Swerve avoidance of static obstacles

For \mathcal{M}_1 , the key is to plan a vehicle-independent reference trajectory once per lane per cycle, and reuse this plan for multiple on-road vehicles. We make use of the elastic-band algorithm [7] to generate one reference trajectory per lane of interest per cycle, so that all the moving objects in that lane can reuse this planned trajectory (Figure 5).


 Figure 6. \mathcal{M}_2 : Longitudinal avoidance /distance keeping to a leading object

Then the individual vehicles only need to implement cheap-to-evaluate lateral and longitudinal controllers to track the planned reference. For lateral control, tracking controllers described in section II-C1 could be reused. For longitudinal control (\mathcal{M}_2), a constant-time adaptive cruise controller [8] is implemented to perform distance keeping and slowing-down based on the relative distance to the leading object (Figure 6):

$$\begin{cases} S_{desired} = l_{lead} + T \cdot \dot{x}_{host} \\ S_{actual} = x_{lead} - x_{host} \\ \delta = S_{desired} - S_{actual} \\ a_{cmd} = -\frac{1}{h} \cdot (-\dot{S}_{actual} + \lambda \cdot \delta) \end{cases} \quad (3)$$

where $S_{desired}$ and S_{actual} are the desired and actual longitudinal gaps between two vehicles, δ is the difference between these two gaps, l_{lead} is the length of the leading vehicle, x_{host} and x_{lead} are the longitudinal positions of host and leading

vehicle, T is the time coefficient of the controller, h and λ are the tunable coefficients to modify the aggressiveness of the controller, and a_{cmd} is commanded acceleration, which is the output of the controller.

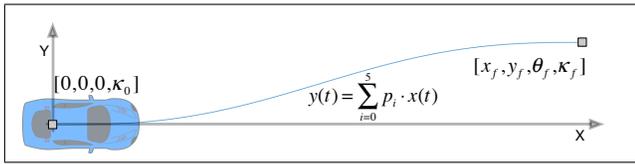


Figure 7. \mathcal{M}_3 : Lane-changing maneuver.

For \mathcal{M}_3 , a single lane-change path is generated using a polynomial [9] that connects from the current state of the object to a look-ahead state in the target lane (Figure 7):

$$y(t) = \sum_{i=0}^5 p_i \cdot x(t) \quad (4)$$

where the polynomial coefficients p_i can be found analytically, hence computationally trivially. The look-ahead distance is an empirical function of its current speed.

Depending on the nature of traffic, lane change can be free or cooperative. The former is trivial. For the latter, if the same controller for \mathcal{M}_2 is used, a significant change in the spacing (after lane-change) between two simulated vehicles will cause huge deceleration and generate a slowing-down shock-wave effect on all following vehicles in the lane. We adopt the controller proposed in [1] to simulate smoother cooperative lane change in dense traffic. Refer to the original paper for more details.

IV. RESULTS

The proposed simulator has been used to develop urban driving motion planning algorithms [7] for the autonomous Cadillac SRX testbed [10]. Compared with the simulator used in the 2007 DARPA Urban Challenge [3], "FastSim" is capable of modeling host vehicle with more accurate dynamics by using dynamic bicycle model, so that the vehicle response at higher-speed can be replicated. Meanwhile, by directly modeling perception system, "FastSim" is capable to recreate the non-perfect sensing limitation imposed by realistic sensors to create challenging test cases for the motion planner. Finally, "FastSim" can model various dynamic objects with more flexible motion patterns, and traffic pattern which is important for simulating urban driving scenarios.

A graphic user interface for FastSim is implemented to facilitate real-time monitoring and manipulation (Figure 8). It consists of six main functional components: world plotter (A), XML-based scenario loading area (B), historical host vehicle measurement (C), world plotter zoom/panning tool (D), simulator/planner stop/go toggle tool (E) and external trigger control panel (F). More description and example usage of FastSim can be found in [11].

V. CONCLUSION

In this paper, we propose a lightweight simulation environment FastSim for rapid MP algorithm development for urban autonomous driving. Three cornerstone simulation components, i.e., surrounding environment, perception and host

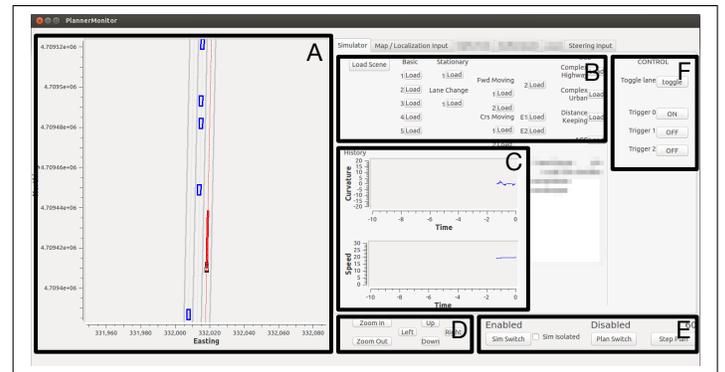


Figure 8. The graphic user interface of FastSim.

vehicle control/execution, are modeled to create a sufficiently complex environment for MP algorithm evaluation. We further proposed efficient models to reproduce basic interaction intelligence of other on-road traffic participants to create a more realistic simulation environment. FastSim is designed with modular programming, hence the models of different simulation components can be swapped for different research projects easily.

In the future, the simulation environment will be further expanded to be compatible with unstructured environments like a parking lot. More moving object models will also be implemented, such as the trailer model for trucks. In addition, more interaction intelligence of other non-vehicle traffic participants like pedestrians and bicyclists will also be investigated and modeled.

REFERENCES

- [1] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," Transportation Research Part C: Emerging Technologies, vol. 13, no. 1, 2005, pp. 37–62.
- [2] C. Urmson et al., "High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004," Robotics Institute, CMU, Tech. Rep. CMU-RI-TR-04-37, 2004.
- [3] M. McNaughton et al., "Software infrastructure for an autonomous ground vehicle," Journal of Aerospace Computing, Information, and Communication, vol. 5, no. 12, 2008, pp. 491–505.
- [4] R. MacLachlan, "Tracking moving objects from a moving vehicle using a laser scanner," Robotics Institute, CMU, Tech. Rep. CMU-RI-TR-05-07, 2005.
- [5] W. Milliken and D. L. Milliken, Race car vehicle dynamics. Society of Automotive Engineers Warrendale, 1995, vol. 400.
- [6] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08, 2009.
- [7] T. Gu et al., "Tunable and stable real-time trajectory planning for urban autonomous driving," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015). To appear.
- [8] R. Rajamani, Vehicle dynamics and control. Springer Science Business Media, 2011.
- [9] J.-W. Lee and B. Litkouhi, "A unified framework of the automated lane centering/changing control for motion smoothness adaptation," in Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. IEEE, Conference Proceedings, pp. 282–287.
- [10] J. Wei et al., "Towards a viable autonomous driving research platform," in Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, Conference Proceedings, pp. 763–770.
- [11] T. Gu and J. Dolan, "Github page for FastSim." [Online]. Available: <http://www.tianyugu.net/publications.html>

A NAO-based Intelligent Robotic System for a Word Search-like Game

Víctor Lobato-Ríos, Angélica Muñoz-Meléndez and José Martínez-Carranza

Computer Science Department

Instituto Nacional de Astrofísica, Óptica y Electrónica, México

Email: vlobato@ccc.inaoep.mx, {munoz, carranza}@inaoep.mx

Abstract—In this paper, we introduce a novel application based on the NAO robotic platform and inspired by the *word search puzzle*. In this scenario, NAO is presented with a worktable with letter tokens on it. The goal of this game is for NAO to be given a word that has to be assembled by using the letter tokens on the table. Thus, the robot has to recognise, reach, grasp and bring, to the bottom of the worktable, the tokens following the order of the letters in the word. For NAO to solve this task, we propose a computational strategy based on a vision system for the letter recognition and a motion planning architecture that will enable him to reach and manipulate the tokens. Our results indicate that our approach is adequate and effective to implement this intelligent robotic system, which also provides the basis for the implementation of a more sophisticated robotic system.

Keywords—Planning; humanoid; grasping; object recognition.

I. INTRODUCTION

Among the several platforms available on the market, the humanoid NAO¹ offers an aesthetic design which makes it appealing among people. Its friendly appearance makes this robot a suitable candidate for companion, however, beyond the cuteness and toy-like appearance, NAO is a genuine robotic platform that can be exploited in several robotic tasks such as object recognition, object manipulation, speech recognition and human-robot interaction.

In this work, we present a novel application of the NAO robotic platform, which contributes to the efforts made to build friendly robots that can sit along humans at home or in social environments for different purposes, *e.g.*, companion, assistance, interaction, etc. In this spirit, we have developed a NAO-based robotic system that aims at solving a game inspired by the *word search puzzle*². In this task, NAO is presented with a worktable where a set of tokens are laid on it, see Fig. 1. Each token has a letter drawn on it and the goal in this game is for NAO to be given a *word* whose compounding letters are found among the tokens.

From the above, in a one-by-one fashion driven by the order of the letters in the requested word, NAO has to: i) recognise, with the help of his vision system, a token whose letter is part of the word; ii) move his left or right hand towards the recognised token on the worktable; iii) grasp the token; iv) move the token towards the bottom part of the worktable and release it in the corresponding position indicated by the order of the letters in the word. The task ends when the word is fully formed as illustrated in Fig. 1, where it can be seen that, at the bottom of the worktable, NAO has formed the word *CAT*.

¹NAO robot is developed by the company Aldebaran. For more information consult: <https://www.aldebaran.com/en/humanoid-robot/nao-robot>

²Word search puzzle: A puzzle consisting of letters arranged in a grid, containing several hidden words written in any direction [1].



Figure 1. NAO robot assembling the word *CAT*: the tokens with the letters corresponding to the word are placed at the bottom of the table in the respective order.

For NAO to accomplish the task, we have developed a visual recognition system and an efficient planning mechanism that enables NAO to decide which arm to use and the trajectory that it has to follow in order to grasp a token and where to release it. The relevant tokens are recognised with the help of our vision system, which analyses the imagery obtained with NAO's onboard camera. Also, our proposed planner allows us to include relevant constraints in the configuration space such as: possible hand rotations; grasping execution; passing the token from one hand to the other and updates in the configuration space due to accidental (or intentional) changes in the position of the tokens on the worktable.

Our experiments indicate that our approach is effective and the obtained results are promising. They indicate that we are on track in terms of developing an intelligent robotic system that could be used at home or in a social environment in the future.

In order to describe our approach, this paper has been organised as follows: Section II presents the related work; the building blocks of our intelligent robotic system are presented in Section III; Section IV describes our experiments and results; Section V discusses our work and its scope; and Section VI presents our conclusions and future work.

II. RELATED WORK

Several works related to the NAO robot are focused on improving its performance in the RoboCup Standard Platform League³. The main topics here are object recognition, color

³<http://www.tzi.de/spl/bin/view/Website/WebHome>

TABLE I. COMPARISON WITH CLOSEST RELATED WORK

	Domain	Resources & capabilities	Real time adaptation
Kovacic et al. [14]	Tic Tac Toe	-Tokens recognition -Tokens' manipulation -Arms motion	No
Jost et al. [13]	Simon's game	-Speech recognition -Arms motion	No
Our work	Word search-like game	-Letters recognition -Tokens' manipulation -Arms motion	Yes

segmentation, gait improvement and fall protection [2][3][4][5][6]. In this context, many works try to improve specific movements of the robot such as its grasping or its balance while performing a kick, whereas others focus on modelling the kinematics of the robot [7][8].

However, because of its nice appearance, the NAO robot has become a perfect candidate for tasks involving interaction with people, especially in those involving children, elders or people with special needs. For example, Janssen et al. [9] developed a game to motivate children to learn arithmetic through imitation activities between the robot and the child. In [10], NAO is used to learn a set of physical exercises from an expert trainer and then NAO is used to teach the moves to elder people. Additionally, in [11] NAO has been used with the Kinect 3D vision system seeking to imitate upper limb movements of stroke rehabilitation patients.

Further, applications for NAO have also included daily life activities like bringing a cup of coffee to someone [12], where they made the motion plan for each arm in order to accomplish the grasping of a cup and then release it when someone wants to take it.

However, the closest works related to our research have been developed by Jost et al. [13] and Kovacic et al. [14]. The former adapts the Simon's game to a scenario where a NAO robot presents sequences of color that must be repeated and extended by a user. In the latter, a NAO robot plays Tic-Tac-Toe against a person; for that, the robot recognises the game elements visually. Table I compares the main features of these and our work.

III. ASSEMBLING WORDS WITH NAO ROBOT

In order to assemble a word with the NAO robot, which involves letter recognition, grasping and arm motion planning, two main modules were developed. The first module focuses on visual recognition, which analyses the images captured by the onboard camera in order to find the letters, among the tokens on the worktable, that compose the requested word. The second module is responsible for generating the motion plan for the robot's arms with three main goals: i) to move the closest arm towards the token with the recognised letter; ii) to grasp the token; iii) to move the arm whilst holding the token towards the corresponding position at the bottom of the worktable, where the token has to be placed/released. All of the previous steps are necessary in order to assemble the requested word. Both modules and the movement constraints that must be considered will be explained in detail below, but first, a general description of the robot activities will be presented in the platform setup.

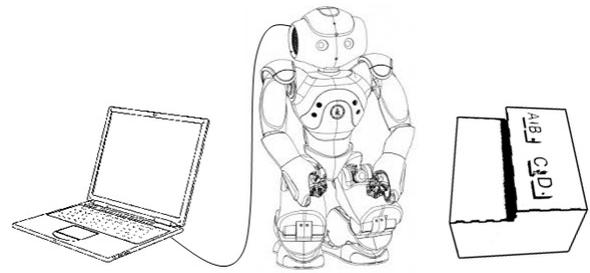
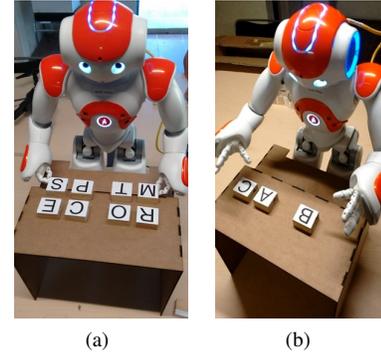


Figure 2. Components of the experimental platform.

Figure 3. NAO robot with its worktable with: (a) *StandInit* position. (b) Ready position.

A. Platform Setup

The platform used in this research consists of a NAO robot v4 using the software NAOqi 1.14.4⁴. The robot is connected through an Ethernet cable or through WiFi to a laptop with an Intel Core 2 Duo P8400 processor (2.26 GHz) operating under Windows 8.1. It is worth to remark that the processing of images captured by the robot as well as the planning of arm movements are done on the laptop locally. For a sketch of the platform see Fig. 2.

To begin its activities, the robot must be placed in front of its worktable as depicted in Fig. 3(a). The posture of the NAO is the one defined into the NAO robot environment as *StandInit*, which gives stability and motion freedom to the robot. Immediately after, the robot assumes a little different posture, first, tilts its head to look at the worktable and then raises both arms to chest height as shown in Fig. 3(b).

At this point, the robot is ready to start the visual recognition and obtain the coordinates of the letters that must be reached. As it was stated above, the robot can assemble words by recognising and reaching the letters among the tokens on the worktable as illustrated in Fig. 3(a), where the robot will attempt to assemble the word *MORE*. Furthermore, this same problem is equivalent to assembling a sequence of letters such as it is shown in Fig. 3(b), where NAO will attempt to assemble the letter sequence *A-B-C* from the tokens on the worktable. Either way, the requested word or letter sequence must be previously known by the robot.

The worktable has two levels, the upper level is where the candidate letters to be recognised are laid on. The lower level is where the robot will bring and release the recognised tokens in order to assemble the word or letter sequence. Once

⁴Documentation of NAOqi 1.14.4 can be found here: <http://doc.aldebaran.com/1-14/index.html>

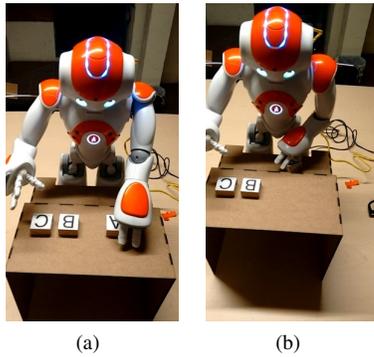


Figure 4. (a) Taking the letter. (b) Releasing the letter.

the coordinates of the letters are known, the robot will grasp these one by one as depicted in Fig. 4(a). NAO will recognise, reach, grasp and bring the tokens by following the order of the letters in the word or in the letter sequence, thus placing these in that order on the lower level (at the bottom) of the worktable as shown in Fig. 4(b).

This procedure ends when the robot has placed every letter of the word or the sequence at the bottom of the table and in the right order. The next section will describe in more detail the vision module, which helps to recognise and locate the letters among the tokens, and how their spatial positions are located.

B. Visual Recognition

Once letters that compose the requested word or sequence are known, the next step is to recognise these letters on the image retrieved by its onboard lower camera. We used this camera because it is located at the mouth of the robot and this makes it easier to see all the surface of the worktable than when using the upper camera located at the forehead of the robot. The captured images have a resolution of 320×260 pixels. However, when using the lower camera, due to its position with respect to the worktable the objects observed with this camera exhibit slight changes in appearance w.r.t. their original appearance, *i.e.*, objects at the bottom of the image are slightly bigger whereas objects at the top are slightly smaller and objects close to the left or right of the image suffer moderate affine transformation. Therefore, the visual recognition model has to be robust against such transformations even if these are small.

To deal with the transformations mentioned above, we use the skeleton of the letter as the template to be sought out on the image where the recognition has to be carried out. To create the template, each token is placed on the worktable in such a way that the letter appears centred on the camera image. Once the letter is well located a segmentation algorithm, based on colour segmentation and morphological operators, is used to extract the white area on the image, corresponding to the token, and then over that area the skeleton is extracted. The resulting template has an average size of 38×42 pixels. Thus, at recognition time, for each letter of the requested word its template is sought out in all the image, the pixel position with the highest similarity score, obtained with normalised crossed correlation, is used as the found image position indicating that the letter has been recognised in that image position.

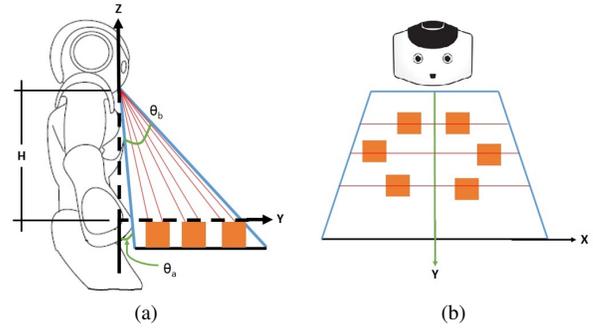


Figure 5. (a) Side view of the vision cone of the NAO robot. (b) Top view of the vision cone of the NAO robot.

The procedure described above will return the (x_p, y_p) image position of a recognised letter and in order to obtain the metric token's position (x, y) on the worktable's surface, we use a simple but effective interpolation method that converts a coordinate (x_p, y_p) into (x, y) . For the coordinate y , the method uses the angles θ_a and θ_b depicted in Fig 5(a), which correspond to the angles formed in between the camera optical centre and the lower and upper part of the worktable whose sides are observed in the first and last row of the camera image, hence $\theta_y = y_p \frac{\theta_b - \theta_a}{h}$, where h is the total number of lines in the image. From the latter and knowing the camera's height H w.r.t. the worktable, we have that $y = H \tan(\theta_y)$. A similar procedure is carried out in order to calculate x . In this case the vertical length of the worktable is used instead of H , see Fig. 5(a), where the known value of y can be used to simplify the calculations.

C. Movement Constraints

The next module in our work is the motion planning algorithm used to reach and grasp, with the robot's arms, the located recognised letters. This module includes some constraints related to the robot's arm motion, specifically, to its wrist which only has one degree of freedom (DOF) in roll. As a consequence, the robot's hand can pick a token only if the arm and wrist are parallel to the worktable's y axis, see Fig. 5(b). Another constraint is that related to the area that can be reached with either hand. This is mainly due to the DOF of the servo-motors in each shoulder. Fig 6 indicates what tokens on the worktable can be reached with what hand. Due to these constraints and the physical size of the tokens used in this work, the maximum number of letters that the robot is able to assemble is four. Nevertheless, we should highlight that this number could be increased by using smaller tokens or having a humanoid with more DOF but in either case, our proposed methodology would remain the same.

Also, observe that there might be cases when a recognised letter may be located on the left side of the worktable but then, the robot has to place it on the bottom right side. In this case, the corresponding token will be reached and picked by the left hand and then this must be exchanged to the right hand so that the right hand can bring the token to the bottom right side. These steps are depicted in Fig. 7, which demonstrates the ability of our system to deal with these type of situations due to the fact that such cases are considered within our motion planning algorithm.

Finally, we should highlight the fact that our motion

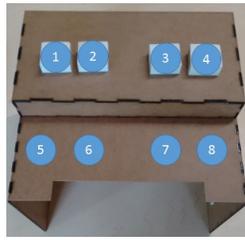


Figure 6. Possible token positions (1-4) and approximate goal positions (5-8).

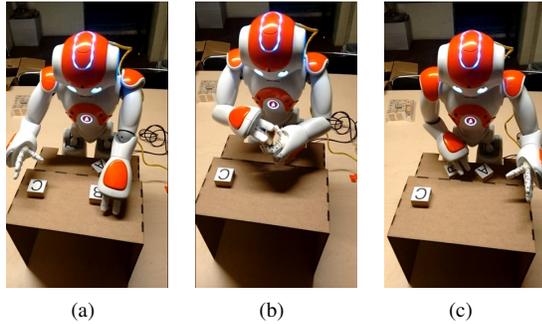


Figure 7. (a) Taking the letter. (b) Exchanging the letter. (c) Releasing the letter.

planning algorithm includes in its configuration space the trajectories that the robot’s arm has to traverse in order to reach a token. It also includes the situations when either hand has to rotate (usually when there is an exchange of the token from one hand to the other), and the trajectories that will bring a token to its corresponding position at the bottom of the worktable.

D. Motion Planning

As mentioned at the end of the last section, our motion planning algorithm takes into account the arm constraints in combination with the token positions on the worktable, this is, those positions where the tokens have to be reached and picked and those where these have to be released. In order to model all of the arm motions involved during the game, we design a *search tree* which is shown in Fig. 8.

Each node of the tree represents the goal physical positions where the robot’s hand has to arrive. This position includes the hand’s translation and orientation. In this sense, the labels of nodes indicate the motions for reaching and leaving a token position, represented with the prefixes *R* and *L*, respectively. Also, in order to perform an exchange, nodes are labelled as *REL* to represent the right hand that has to reach the left hand, and as *RER* to represent the left hand that has to reach the right hand. Finally, if an exchange was performed, then the tree includes the nodes with the required motions in order to leave the exchange position, these are labelled with the *LEL* and *LER* in similar fashion to the previous nodes.

Note that the nodes are coloured in either orange or blue, which indicates which hand is manipulating the token. For instance, orange nodes indicate that during the motion execution driven by any of the orange nodes, the left hand could grasp a token or, if it is already holding it, release it in its corresponding position. The numbers associated with each node correspond to the physical positions of the tokens on the worktable, see Fig. 6.

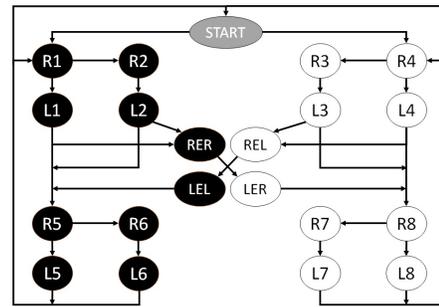


Figure 8. Search tree for planning.

From the above, a node with the label *R4* means that the right arm has to traverse certain trajectory in order to reach the position specified by *R4* and at such position the arm’s hand could grasp a token, assuming that a token is physically in that position. If no grasping is required, the arm can move towards another accessible position which is indicated by the arrows in the tree, in this case, from *R4* the arm can move towards *R3*, a token position, and if a token has been grasped then a leaving motion has to be performed in order to bring the token to the bottom of the worktable. The motion planning for this execution will be indicated by traversing the nodes *L3* and any of the other nodes that drive the arm towards a release position, for instance *R8*.

Note that the nodes representing the hand exchanges will contain the required motion for one arm’s hand to reach the other arm’s hand so that the exchange can be executed. For instance, let’s say that the left hand has grasped a token from the position indicated by the node *R2* and it has to be released on the other side of the worktable, then the arm has to traverse the position indicated by the node *L2* and then towards the exchange position indicated by the node *RER*, where the left hand will reach the right hand so the exchange can be executed. Immediately after, the right hand will leave the exchange position following the motion indicated by *LER* and then another node or set of nodes in the tree will have to be traversed to define the motion that the arm will have to follow in order to bring the token to the bottom of the worktable, for instance the sequence *R8-R7*.

In this way, our proposed tree represents all the paths that could be traversed by the robot’s arms. Therefore, once a letter has been recognised and its physical position on the worktable calculated by the vision module, the tree will be useful to determine which path is the shortest path in order to drive the arm towards the corresponding token in order to reach it, grasp it and bring it towards its position where it has to be released. This path is found by executing a depth-first search, which produces all the possible paths and that will return the shortest.

A remarkable feature of the proposed methodology based on our search tree is that once a token has been released in a certain position, defined by a node, for the next recognised letter the search for the shortest path does not have to begin from the *start* robot’s position, but it can start from the actual position, *i.e.*, from the current node in the tree. These will save some computational time in the search, but more importantly, it will avoid unnecessary arm motion since having to return to the start position will waste time and energy (NAO’s servo motors decrease their performance as time goes by due to

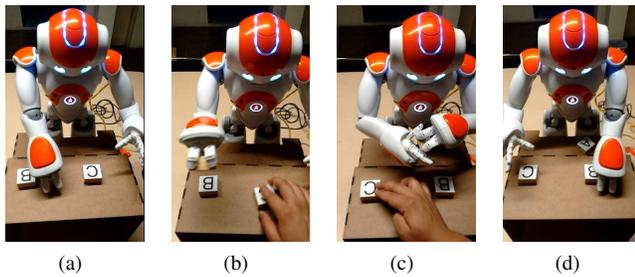


Figure 9. (a) Taking the first letter. (b-c) Changing the expected intermediate state from C- \boxtimes -B to B- \boxtimes -C. (d) Dealing with changes.

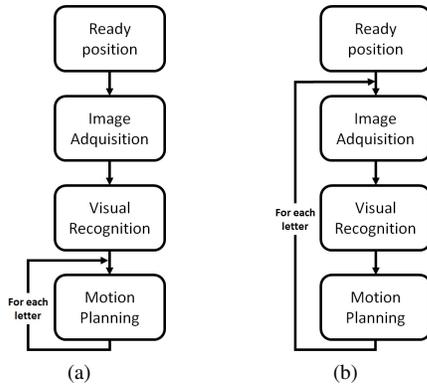


Figure 10. Computational strategies implemented in this work: (a) Static, the basic strategy used to test our proposed application; (b) Dynamic, which is robust to changes in positions of the tokens during the game.

overheat, hence it is convenient to move the arm as efficiently as possible).

To implement the planner described above, we followed two strategies, see Fig. 10. In a *static strategy*, a picture of the environment is taken only once at the beginning of the game. This image will be used to recognise each one of the letters in the word or letter sequence. However, if the tokens change their position later in time then the planner will not be able to correct the plan since it will believe that the tokens remain always in the same place. The second strategy is the *dynamic strategy* where a picture of the worktable is taken soon after a token has been released, *i.e.*, the planner calculates the position of the next token just before going for it. This will bring robustness against changes in the position of the tokens since it will enable the planner to correctly assign the shortest path given that it will use the updated token's position. Fig. 9 shows an example of the dynamic strategy where a user intentionally changes the position of the tokens while the robot is attempting to bring a grasped token to the position where it has been released. Note that, regardless the change in position, the robot manages to identify correctly the updated positions, hence calculating and performing an adequate plan, something that would fail with the *static strategy*.

IV. EXPERIMENTS

Three experiments were conducted in order to test our strategies. The first one is a time invested comparison between both strategies in static environments. The second experiment is a time analysis of the dynamic strategy in dynamic environments. Both experiments were performed for assembling

TABLE II. TIME INVESTED BY STATIC AND DYNAMIC STRATEGIES FOR REACHING THE GOAL SEQUENCE A-B-C.

Initial state	Average time (seconds)	
	Static strategy	Dynamic strategy
A-B-C	76.66 \pm 0.40	76.83 \pm 0.15
B-A-C	92.59 \pm 0.53	92.69 \pm 0.11

sequences of three letters. The third experiment is a test of the performance of the dynamic strategy to achieve the long case.

The worktable used for these experiments has these dimensions: 30 cm of length, 15 cm of width of the upper level, 12 cm of width of the lower level, 24.5 cm of height of the upper level and 20.5 cm of height of the lower level. Also, wooden tokens of $3.5 \times 4 \times 2$ cm and weight of 11g with the letters printed on both sides were used.

All the experiments were run 5 times in similar conditions, and averages and standard deviations were then calculated. It is important to mention that the NAO robot has an automatic monitor for the temperature of its joints. An alert message warns about a situation of high temperature that might cause an unexpected behavior of the robot. Thus, a condition was also that any warning message appeared at the beginning of the experiments.

A movie of these experiments can be watched here [15].

A. Simple Static Environments

The goal of the first experiment was to compare the time invested by both strategies to achieve a sequence of three letters in static environments. The goal sequence was A-B-C and two initial positions were tested, A-B-C and B-A-C. The first one was an easy case because the robot just has to take the tokens and carry them to the goal position in the same order, whereas the second case was harder because it requires two exchanges. The results of this experiment are shown in Table II.

The intuitive hypothesis suggests that the dynamic strategy might be more expensive than the static one since the former invests more time in the process of double-checking the conditions of the environment. However, this hypothesis was not verified by our experiments. Even though the static strategy scored better times than the dynamic strategy, 170 ms and 100 ms for the easy and hard case, respectively, the difference is not statistically significant.

As expected, the harder case takes a longer time, that results from the number of exchanges required to sort the tokens. Note also that the dynamic strategy is in general more consistent in terms of the variation of time invested for solving a problem. That can be explained by the brief pause introduced by the dynamic strategy for the process of double-checking, that benefits at the same time the stability of the robot for starting next movements.

B. Dynamic Environments

The second experiment is intended to verify the ability of the dynamic strategy to react to unexpected changes that might happen in the environment. For that, the goal sequence and the initial states were defined as for the previous experiment, A-B-C, A-B-C and B-A-C, respectively. Once the first token had

TABLE III. TIME INVESTED BY THE DYNAMIC STRATEGY FOR REACHING THE GOAL SEQUENCE *A-B-C* IN CHANGING ENVIRONMENTS.

Initial state	Intermediate state		Average time (seconds)
	Expected	Actual	
A-B-C	☒-B-C	B-C-☒	91.96 ± 0.20
		C-☒-B	85.99 ± 0.14
B-A-C	B-☒-C	B-C-☒	98.78 ± 0.19
		C-☒-B	92.58 ± 0.06

TABLE IV. TIME INVESTED BY THE DYNAMIC STRATEGY FOR REACHING THE GOAL SEQUENCE *A-B-C-D*.

Initial state	Average time (seconds)
A-B-C-D	91.61 ± 0.12
D-C-B-A	123.93 ± 0.07

been taken by the robot, the rest of the tokens were manually resorted in a different configuration. The ability of the dynamic strategy to double-check the state of tokens must be able to deal with these changes. The results of this experiment are shown in Table III.

Two important remarks can be highlighted from these results. First, that the dynamic strategy is effectively able to deal with unexpected changes in the environment, since for all cases including intentional changes of the tokens the robot was able to achieve the goal sequence. And second, that the time invested by this strategy to solve unexpected changes depends more on the specific configuration of the tokens than on the number of exchanges of tokens.

To illustrate the second remark, note that from the initial state, *B-A-C* and the intermediate state *C-☒-B* for reaching the goal sequence *A-B-C* (fourth row of Table III), the robot takes practically the same time invested for going from *B-A-C* with the intermediate state *B-☒-C* for the same goal sequence (second row of Table II). However, solving the same case with the intermediate state *B-C-☒* increases the time in 6 seconds, on average. We have noticed that the time for solving a case is increased particularly for tokens located in middle positions of the worktable, independently of the distance to reach the right position of the token in the goal sequence.

C. Testing with a Large Case

The goal of this experiment was that of testing the performance of the dynamic strategy for solving a larger case, *i.e.*, assembling sequences of four letters that is the maximum number of letters that can be solved by the robot in the current settings.

In this case, the goal sequence was *A-B-C-D* and the initial states were *A-B-C-D* and *D-C-B-A*, an easy and a hard case as for previous experiments. The dynamic strategy was chosen over the static one for its capability to corroborate the conditions of the environment that is convenient when dealing with a large sequence. The results of this experiment are shown in Table IV.

From these results, we confirm that the dynamic strategy is able to solve the largest possible goal sequences in the current settings, as well for easy than for hard cases. Also, it is worth to notice that this strategy scores in general results with small variations, which makes it a stable strategy.

V. DISCUSSION

Programming intelligent robots able to interact with people successfully requires the sum of small efforts in different fields such as improving, for instance, visual recognition algorithms, speech recognition skills, and grasping capabilities. However, a challenging issue that is particularly relevant for this kind of robots is that of how to achieve a flexible goal-oriented behaviour, *i.e.*, how to combine the capabilities for long-term planning with the capabilities for prompt reaction.

A Word Search-like Game using a NAO robot was chosen as a case study for investigating both: issues related to technical aspects of the robot, such as image processing and grasping, and issues related to behavioural aspects of the robot, such as programming strategies for dealing with static and dynamic environments.

The problem itself, a Word Search-like Game, was also defined for a number of reasons that are summarised as follows: i) it is a bound problem, *i.e.*, it has a well-defined set of rules, basic tokens, and initial and target positions; ii) it is a problem whose difficulty can be gradually increased, *i.e.*, variations of the game can be easily extended, for instance by giving NAO spoken commands; iii) it is a problem naturally involving human-robot interaction, *i.e.*, a scenario where a NAO robot plays with children, or assists teachers and helps students to learn spelling is absolutely thinkable.

From the above, a Word Search-like Game using a NAO robot is a problem that contains important ingredients for becoming a solid benchmark for studying human-robot interaction and designing intelligent robots.

VI. CONCLUSION AND FUTURE WORK

We have presented a novel application of the NAO robotic platform in the form of a Word Search-like Game. For this, we have developed a strategy that involves two main modules: a vision system that recognises letter tokens on a worktable, and a motion planning module that resolves what motion the robot's arms have to execute to reach and manipulate the tokens in order to solve the game.

Our proposed strategies enable the robot to solve a problem requiring goal-oriented behaviour as well as reactivity against dynamic changes of the tokens' positions. These capabilities are crucial for designing intelligent robots able to interact successfully with people.

Programming automatic players able to successfully play board games and puzzles is a challenging problem in the field of Artificial Intelligence. In effect, there are many non-trivial competences involved in the way people learn and play board games, such as representation of knowledge, identification and refinement of game strategies, and recognition of the opponent's expertise -in the case of interactive games- to mention a few examples.

Programming robotic players able to successfully play board games is a very appealing problem that combines issues encountered in Artificial Intelligence research with physical

and tangible considerations, such as motion constraints, perceptual limitations, time of response, among other.

We strongly believe that the platform and problem addressed in this research are on the track for developing intelligent robotic systems.

In the future, we plan to enhance our approach in order to exploit the walking capabilities of the NAO platform in order to address tasks where the robot has to bring a token to a different location outside the worktable. We will also find potential research scenarios where our NAO-based application can be exploited, for instance educational robotics and robots companions.

ACKNOWLEDGMENT

The first author is supported by the Mexican National Council for Science and Technology, CONACYT, under the grant number 336541. This work was partially supported by the RAFAGA project, funded by the Royal Society-Newton Advanced Fellowship, 2015-2017.

REFERENCES

- [1] Oxford Dictionaries, "Definition of: word search". Available on: <http://www.oxforddictionaries.com/definition/english/word-search?q=word+search>
- [2] T. González, "Artificial Vision in the Nao Humanoid Robot", Master's Thesis, Department of Computer Science and Mathematics, Rovira I Virgili University, Sept. 2009, chapter 5-8, pp. 30-77, URL: http://upcommons.upc.edu/pfc/bitstream/2099.1/7722/1/MT_TomasGonzalezSanchez-URV.pdf [accessed: 2015-05-29].
- [3] C. Graf, A. Härtl, T. Röfer, and T. Laue, "A Robust Closed-Loop Gait for the Standard Platform League Humanoid", in Proceedings of the 4th Workshop on Humanoid Soccer Robots (Humanoids 2009), Paris, France, 2009, pp. 30-37.
- [4] S. Liemhetcharat, B. Coltin, and M. Veloso, "Vision-Based Cognition of a Humanoid Robot in Standard Platform Robot Soccer", in Proceedings of the 5th Workshop on Humanoid Soccer Robots (Humanoids 2010), Nashville, USA, 2010, pp. 47-52.
- [5] J. Ruiz-del-Solar, R. Palma, R. Marchant, S. Parra, and P. Zegers, "Learning to fall: Designing low damage fall sequences for humanoid soccer robots", *Robotics and Autonomous Systems*, vol. 57, Issue 8, 2009, pp. 796-807.
- [6] J. Strom, G. Slavov, and E. Chown, "Omnidirectional Walking Using ZMP and Preview Control for the NAO Humanoid Robot", *RoboCup 2009: Robot Soccer World Cup XIII. Lecture Notes in Computer Science*, vol. 5949, 2010, pp. 378-389.
- [7] N. Kofinas, E. Orfanoudakis, and M. Lagoudakis, "Complete Analytical Inverse Kinematics for NAO", in Proceedings of the 13th International Conference on Autonomous Robot Systems (Robotica) Lisbon, Portugal, 2013, pp. 1-6.
- [8] H. Mellmann and G. Cotugno, "Dynamic Motion Control: Adaptive Bimanual Grasping for a Humanoid Robot", *Fundamenta Informaticae*, vol. 112, no. 1, 2011, pp. 89-101.
- [9] J. Janssen, C. Wal, M. Neerinx, and R. Looije, "Motivating children to learn arithmetic with an adaptive robot game", *Social Robotics. Lecture Notes in Computer Science*, vol. 7072, 2011, pp. 153-162.
- [10] B. Görer, A. Salah, and H. Akin, "A Robotic Fitness Coach for the Elderly", *Ambient Intelligence. Lecture Notes in Computer Science*, vol. 8309, 2013, pp. 124-139.
- [11] J. Ibarra, A. Malo, A. Gómez, J. Lavín, L. Rodríguez, and W. Sierra, "Development of a system based on 3D vision, interactive virtual environments, ergonomic signals and a humanoid for stroke rehabilitation", *Journal of Computer Methods and Programs in Biomedicine*, vol. 112, Issue 2, 2013, pp. 239-249.
- [12] J. Müller, U. Frese, T. Röfer, R. Gelin, and A. Mazel, "GRASPY – Object Manipulation with NAO", *Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe: Springer Tracts in Advanced Robotics*, vol. 94, 2014, pp. 177-195.
- [13] C. Jost, M. Grandgeorge, B. Le PÈVÈdic, and D. Duhaut, "Robot or Tablet: Users' behaviours on a Memory Game", *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, Scotland, 2014, pp. 1050-1055.
- [14] Z. Kovacic, F. Petric, D. Miklic, A. Babic, and K. Hrvatinic, "NAO Plays a Tic-Tac-Toe Game: Intelligent Grasping and Interaction", *University of Zagreb*, Feb. 2014. Available from https://www.fer.unizg.hr/_download/repository/nao_book.pdf [accessed: 2015-05-25].
- [15] V. Lobato-Ríos, A. Muñoz-Meléndez, and J. Martínez-Carranza, "Video: A NAO-based Intelligent Robotic System for a Word Search-like Game". Available on: <http://youtu.be/xvMX5glu7Jk>

LQG Control of a Two-Wheeled Mobile Pendulum System

Ákos Odry, Ervin Burkus, Péter Odry

Department of Control Engineering and Information Technology

College of Dunaújváros

Dunaújváros, Hungary

E-mail: {odrya, burkus, podry}@mail.duf.hu

Abstract—In this paper, the linear-quadratic-Gaussian control of a mechatronic system will be studied. The mechatronic system is a special mobile robot (called two-wheeled mobile pendulum) having two-wheels, two contact points with the supporting surface and its center of mass is located under the wheel axis. Due to the mechanical structure, the inner body (which acts as a pendulum between the wheels) tends to oscillate during the translational motion of the robot, thus the application of modern control methods is essential in order to stabilize the dynamical system. In the first part of the paper, the mechatronic system and the corresponding mathematical model are introduced, while in the second part different controllers are designed for the plant. The achieved control performances are analyzed based on simulation and implementation results.

Keywords—LQG control; Kalman-filter; mobile robot; self-balancing robot; future transportation system

I. INTRODUCTION

The linear-quadratic-Gaussian (LQG) technique is a beloved method in the control of dynamical systems since it provides the optimal state feedback gain based on the well-developed mathematical algorithm [1]. Numerous researches have been dealt with its application and control performance in real embedded environments such as references [2] - [9]. Divelbiss and Wen [2] presented their experimental results of the tracking control of a car-trailer system, where linear quadratic regulator was used to track the trajectory. Ji and Sul [4] proposed an LQG-based speed control method for torsional vibration suppression in a 2-mass motor drive system, which gave satisfying performance and robust behavior against parameter variations. Bouabdallah, Noth and Siegwart [5] compared the control performances of the PID and linear-quadratic (LQ) techniques applied to an autonomous four-rotor micro helicopter, and emphasized that the control performance of the later technique was influenced by model imperfections. Recent efforts broaden further the set of experimental research results regarding the LQG control, including the control of inverted pendulum type assistant robot [6], self-balancing unicycle robot [7], unmanned helicopter in an uncertain environment [8], inverted pendulum system [9], and quadrotors [3] as well. All the mentioned papers show that the LQG technique have proved its competitive performance in the control of dynamical systems.

Our research goal is to analyze critical balance tasks through the design, optimization and validation of modern control methods. Thus, as a first step, the design and

implementation of the well-known LQG controller had been performed, where the resultant control performances form the initial results of our research analysis. These initial results can be used in the design optimization or in the comparison with other control methods (such as fuzzy control), which is left for another publication.

On the other hand, it can be observed that nowadays technological efforts focus on the development and application of modern control solutions by which mechanical or other kind of anomalies are compensated in dynamical systems (the result of such aspirations is for example the transportation vehicle Segway). Since the two-wheeled mobile pendulum system shall be stabilized by feedback in order to achieve translational motion, we also strengthen and develop this tendency by the analysis and design of modern, optimized control solutions for the plant.

The remainder of this paper is organized as follows. In section II, the mechatronic structure of the robot is introduced, while in section III, the corresponding nonlinear mathematical is derived. In section IV, the control tasks and the LQG control method are reviewed. From section V, the control strategies are elaborated, namely, in section V, the anti-sway speed control-, while in section VI, the balance control of the robot are designed with simulation results. Section VII deals with the state estimation, while in section VIII, the application of the complete LQG control strategies is described. The implementation results of the elaborated control strategies are given in section IX. Section X contains the conclusions and the future work recommendations.



Figure 1. Photograph of the fabricated robot.

II. THE FABRICATED MECHATRONIC SYSTEM

The mechatronic system is a special mobile robot (called two-wheeled mobile pendulum) consisting of two wheels and a steel inner body (chassis). The wheels are actuated through DC motors that are attached to the body. As it can be seen in Figure 1, the diameter of the wheels is bigger than the diameter of the intermediate body, thus the robot has only two contact points with the supporting surface. Due to this mechanical structure, the inner body behaves as a pendulum between the stator and rotor of the applied DC motors, and tends to oscillate when the robot performs translational motion.

Since the location of the center of mass of the robot can be under and above the wheel axis, two equilibrium points can be distinguished. Namely, the robot stays around its stable equilibrium point when the center of mass is located under the wheel axis. Therefore, around this state the translational motion of the robot is affected by the damped oscillation of the inner body. On the other hand, the robot is operated around its unstable equilibrium point when the center of mass is stabilized above the wheel axis. Around the unstable equilibrium point, the robot simultaneously performs translational motion and balances its inner body, which acts as an inverted pendulum. For video demonstration see the website [10].

The electronic construction is built around two 16-bit ultra-low-power Texas Instruments MSP430F2618 microcontrollers (hereinafter MCU1 and MCU2). The applied sensors are summarized in Table 1. The actuators are 3V geared DC micromotors (type: 1024N003S) manufactured by Faulhaber. The motors are driven with pulse width modulation (PWM) signals through Texas Instruments DRV592 drivers. The electronic system is supplied from stabilized 3.3V, the source is a 1 cell lithium-polymer (Li-Po) battery. A 16 MHz quartz oscillator is used as the system clock.

Similar construction was built at the McGill University's Centre for Intelligent Machines [11]. It was proven that the two contact point construction is characterized by the so called quasi-holonomic property that eases the control of nonholonomic systems. Another corresponding two contact point construction is the electric diwheel built by the School of Mechanical Engineering at the University of Adelaide [12].

III. MATHEMATICAL MODEL

To be able to efficiently design the control algorithms of the system, its mathematical model has to be obtained first. Most of the electrical and mechanical parameters that characterize the robot dynamics (such as wheel radius or resistance of the motor) are quite accurately known from direct measurements, datasheets or from calculations performed by Solidworks, the rest of the parameters were experimentally tuned based on the measurements.

We indicate with θ_1 and θ_2 the angular displacements of the wheels, while with θ_3 the inclination angle of the pendulum (inner body). The parameters that characterize the robot are summarized in Table 2 in the appendix. The following notations will be also used: $\dot{\theta}$ as the mean value of $\dot{\theta}_1$ and $\dot{\theta}_2$, $\dot{\psi}$ as the change of yaw angle of the robot, and \dot{s} as

TABLE I. THE APPLIED SENSORS IN THE EMBEDDED ELECTRONICS

Sensor	Manufacturer	Type
Accelerometer	STMicroelectronics	LIS331DL
Gyroscope	STMicroelectronics	L3G4200D
Current sensors	Texas Instruments	INA198
Incremental encoders	Faulhaber	PA2-100

the linear speed of the robot, i.e., $\dot{\theta} = (\dot{\theta}_1 + \dot{\theta}_2)/2$, $\dot{\psi} = r(\dot{\theta}_2 - \dot{\theta}_1)/d$, and $\dot{s} = r\dot{\theta}$.

The motion of the system was determined by the help of the Lagrange equations [13], which lead us to the following equations of motion of the mechanical system [14]:

$$M(q)\ddot{q} + V(q, \dot{q}) = \tau_a - \tau_f, \quad (1)$$

where $M(q)$ denotes the 3-by 3 symmetric and positive definite inertia matrix, $V(q, \dot{q})$ denotes the 3-dimensional vector term including the Coriolis and centrifugal force terms and also the potential (gravity) force term.

The Lagrange function and the exact elements of the matrices in (1) are described in the appendix. For the vector of generalized coordinates $q = (\theta_1, \theta_2, \theta_3)^T$ was chosen, since it contains the minimum number of independent coordinates that define the configuration of the system. The generalized external forces in (1) consist of the torques τ_a that are produced by the motors and the effect of friction τ_f that is modeled in the system [14]. The torques τ_a are described by the differential equation (2) where the input voltages and currents of the motors are denoted with $u = (u_1, u_2)^T$ and $I = (I_1, I_2)^T$.

$$\begin{aligned} \dot{i} &= \frac{1}{L} \left(u - k_E k \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \dot{q} - RI \right), \\ \tau_a &= k_M k \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} I. \end{aligned} \quad (2)$$

Regarding the effect of friction τ_f , only viscous frictions were assumed. Namely, viscous friction was modelled at the bearings and between the wheels and the supporting surface:

$$\tau_f = \begin{bmatrix} b + f_v & 0 & -b \\ 0 & b + f_v & -b \\ -b & -b & 2b \end{bmatrix} \dot{q}. \quad (3)$$

Based on (1) the state-space representation of the two-wheel inverted pendulum system is obtained. With the state vector $x = (q, \dot{q}, I)^T$ the state-space equation [14]:

$$\begin{aligned} \dot{x}(t) &= h(x, u), \\ h(x, u) &= \begin{bmatrix} \dot{q} \\ M(q)^{-1} (\tau_a - \tau_f - V(q, \dot{q})) \\ \frac{1}{L} \left(u - k_E k \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \dot{q} - RI \right) \end{bmatrix}, \\ y(t) &= x(t). \end{aligned} \quad (4)$$

Remark: In the simulation environment the state-space equation (4) was implemented, however during the design of the LQG controllers the 6-dimensional version defined by the state vector $x = (q, \dot{q})^T$ was used. This outcome was chosen because the current measurements were that noisy that the states $I = (I_1, I_2)^T$ could not be used in the feedback. The 6-dimensional model is derived by neglecting the inductance L of the motors.

IV. CONTROL TASKS

Two control tasks were investigated in the analysis, namely the balance control and the anti-sway control of the plant. Therefore, around the stable equilibrium point the speed control of the robot had been chosen as control task, where the implemented controller shall minimize the resulting oscillation of the inner body. While around the unstable equilibrium point the position control of the robot had been chosen as control task, where the implemented controller shall stabilize the inner body during the translational motion.

A. LQ control

The linear-quadratic control addresses the issue of achieving a balance between good system response and control effort [1]. It is based on a developed mathematical algorithm, which results the optimal state-feedback gain K . The feedback gain K minimizes the quadratic cost function

$$J(x, u) = \frac{1}{2} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + \frac{1}{2} x_N^T Q x_N, \quad (5)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and input of the system described by its state-space equation, while $Q = Q^T \in \mathbb{R}^{n \times n}$, $Q \geq 0$ and $R \in \mathbb{R}^{m \times m}$, $R > 0$ are weighting matrixes. According to the LQR method, the state feedback matrix is given by $K = (R + B^T P B)^{-1} B^T P A$, where $P = P^T \geq 0$ is the unique solution of the Control Algebraic Riccati Equation (CARE). The optimal state-feedback $u_k = -K x_k$ ensures the asymptotic stability of the closed loop system. The feedback matrix K is calculated by the built-in Matlab function `lqr(A,B,Q,R,Ts)`.

B. Reference tracking

Since the LQ control defined by the objective function (5) drives the system from the initial state x_0 to the state $x_d = 0$, the control structure shall be extended with the reference tracking matrixes:

$$\begin{pmatrix} N_x \\ N_u \end{pmatrix} = \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0_{n \times m} \\ I_m \end{pmatrix}, \quad (6)$$

where 0 and I are the zero and identity matrixes respectively (the sizes are given in the subscript).

C. LQG control

In the development of the optimal LQ control strategy it is assumed the state variables are measurable, and the system is not disturbed by either internal or external noises. However

in practice the opposite situation is quite common, namely, that a part of the state vector is too noisy to use directly in the feedback. The LQG strategy provides optimal control gain to stochastic, noisy systems by minimizing the expected value of the quadratic objective function (5).

Based on the separation principle the LQG control strategy is given by the state-feedback $u_k = -K \hat{x}_k$, where K is the optimal control gain determined by the LQR algorithm, while \hat{x}_k state vector consists of the original states (those states of x_k that were not noisy) and the Kalman-filter based estimation of the noisy states. Let us denote the unmeasurable or noisy states with ξ , than the corresponding noisy linear system can be given by:

$$\begin{aligned} \xi_{k+1} &= \Phi \xi_k + \Gamma \rho_k + v_k, \\ \gamma_k &= P \xi_k + z_k, \end{aligned} \quad (7)$$

where the process and measurement noises are indicated with v and z respectively, and according to the stochastic hypothesis these noises are uncorrelated and their mean value is zero. In this case the Kalman-filter algorithm provides the optimal estimation $\hat{\xi}$ of the state ξ , i.e., $E[\xi_k - \hat{\xi}_k] = 0$ and $E[(\xi_k - \hat{\xi}_k)(\xi_k - \hat{\xi}_k)^T] \rightarrow \inf$. The estimation algorithm can be found in [15].

Therefore, the design steps are the following: i.) Linearization of the mathematical model around an equilibrium point, ii.) Controllability analysis, iii.) Specification of the weighting matrixes, iv.) Calculation of the optimal control gain K , v.) Identification of the noisy states, vi.) Specification of the noise covariance parameters of the filter, vii.) State estimation by Kalman filter and viii.) Application of the state feedback strategy $u_k = -K \hat{x}_k$.

V. CONTROL STRATEGY AROUND THE STABLE EQUILIBRIUM POINT

Around the stable equilibrium point the anti-sway speed control of the robot is elaborated.

A. Elaboration

The linear state space equation is given by the linearization of (4) around the equilibrium $(x_e, u_e) = (0, 0)$:

$$\dot{x} = \underbrace{\left(\frac{\partial h}{\partial x} \right)_{(x_e, u_e)}}_{\tilde{A}_s} x(t) + \underbrace{\left(\frac{\partial h}{\partial u} \right)_{(x_e, u_e)}}_{\tilde{B}_s} u(t), \quad (8)$$

where the subscript s refers to the stable equilibrium point. In order to reduce the complexity of implementation the $\tilde{x} = T x = (s, \theta_3, \dot{s}, \dot{\theta}_3, \psi, \dot{\psi})$ coordinate transformation is applied. The resulting state-space representation is given by:

$$\begin{aligned} \dot{\tilde{x}} &= \begin{bmatrix} 0_{2 \times 2} & I_2 & 0_{2 \times 2} \\ \tilde{A}_{s,21} & \tilde{A}_{s,22} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & \tilde{A}_{s,33} \end{bmatrix} \tilde{x} + \begin{bmatrix} 0_{2 \times 2} \\ \tilde{B}_{s,2} \\ \tilde{B}_{s,3} \end{bmatrix} u, \\ y &= [0_{2 \times 2} \quad \tilde{C}_{s,2} \quad \tilde{C}_{s,3}] \tilde{x}, \end{aligned} \quad (9)$$

where the block matrixes are described in the appendix.

The controllability matrix [1] is given by $M_c = [B \ AB \ \dots \ A^5 B]_{(\tilde{A}_s, \tilde{B}_s)}$ and the evaluation of its rank results $\text{rank } M_c = 4$. Therefore, according to the Kalman rank condition for controllability (KRCC) the system (9) is not controllable, since the dimension of the state vector is $\dim \tilde{x} = 6$. The non-controllable states of \tilde{x} are the position s and the orientation ψ .

Thus, a new coordinate transformation $z = T_{C\bar{c}} \tilde{x}$ is defined, such that $T_{C\bar{c}} = (T_C, T_{\bar{c}})$ is a basis for \mathbb{R}^6 , furthermore the columns of T_C form the basis for the controllable subspace, $\dim T_C = 6 \times 4$ and $\dim T_{\bar{c}} = 6 \times 2$. As a consequence of the definition, the state vector $z = (z_C, z_{\bar{c}})^T$ is clearly divided into two parts, namely $z_C = (\theta_3, \dot{s}, \dot{\theta}_3, \dot{\psi})^T$ denotes the controllable states, while $z_{\bar{c}} = (s, \psi)^T$ contains uncontrollable ones, and the state-space representation becomes

$$\begin{aligned} \dot{z} &= \begin{bmatrix} A_C & A_{C\bar{c}} \\ 0 & A_{\bar{c}} \end{bmatrix} z(t) + \begin{bmatrix} B_C \\ 0 \end{bmatrix} u(t), \\ y &= [C_C \quad C_{\bar{c}}] z(t). \end{aligned} \quad (10)$$

The LQ strategy is elaborated by using the controllable subsystem (A_C, B_C) . The weighting matrixes $Q = \text{diag}(Q_{ii})$ and $R = \text{diag}(R_{jj})$ were defined based on the Bryson's rule, where $Q_{11} = (15 \cdot \pi/180)^{-2}$, $Q_{22} = (0.08)^{-2}$, $Q_{33} = (50 \cdot \pi/180)^{-2}$, $Q_{44} = (50 \cdot \pi/180)^{-2}$ and $R_{11} = R_{22} = 3^{-2}$ were chosen. Solving the CARE the optimal control gain:

$$K^s = \begin{bmatrix} -1.75 & -1.26 & -0.2 & -0.61 \\ -1.75 & -1.26 & -0.2 & +0.61 \end{bmatrix}, \quad (11)$$

while the reference tracking matrixes are

$$N_x^s = \begin{bmatrix} -0.41 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad N_u^s = \begin{bmatrix} 4.28 & -0.37 \\ 4.28 & +0.37 \end{bmatrix}. \quad (12)$$

B. Simulation results

Simulation of the proposed control strategy was done in MATLAB Simulink environment. The simulation results of the closed loop behavior is depicted in Figure 2.

From the top, the first is the linear speed \dot{s} of the robot, the second is the change of yaw $\dot{\psi}$, the third is the resulting oscillation θ_3 of the inner body, while the last one is the applied voltage to the motors. The following reference signals were applied:

- $\dot{s}_d = \{0.4, 0, -0.2, 0\}$ [m/s],
- $\dot{\psi}_d = \{0.5, 0, -1.2, 0\}$ [rad/s].

The simulation results show that the applied control strategy satisfies the requirements, since the controller simultaneously ensures the speed control and the suppression of the inner body oscillations. The dynamics of the plant was sampled at fixed $f_s = 100$ Hz, which equals to the sampling frequency of the applied sensors.

VI. CONTROL STRATEGY AROUND THE UNSTABLE EQUILIBRIUM POINT

Around the unstable equilibrium point the position control of the robot and the stabilization of its inner body is elaborated.

A. Elaboration

The state space equation is given by the linearization of (4) around $x_e = (0, 0, \pi, 0, 0, 0)^T$ and $u_e = 0$:

$$\dot{x} = \underbrace{\left(\frac{\partial h}{\partial x} \right)_{(x_e, u_e)}}_{A_u} x(t) + \underbrace{\left(\frac{\partial h}{\partial u} \right)_{(x_e, u_e)}}_{B_u} u(t), \quad (13)$$

where the subscript u refers to the unstable equilibrium point. Again, the coordinate transformation $\tilde{x} = T x = (s, \theta_3, \dot{s}, \dot{\theta}_3, \psi, \dot{\psi})$ is applied:

$$\dot{\tilde{x}} = \begin{bmatrix} 0_{2 \times 2} & I_2 & 0_{2 \times 2} \\ -\tilde{A}_{s,21} & \tilde{A}_{s,22} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & \tilde{A}_{s,33} \end{bmatrix} \tilde{x} + \begin{bmatrix} 0_{2 \times 2} \\ \tilde{B}_{s,2} \\ \tilde{B}_{s,3} \end{bmatrix} u, \quad (14)$$

$$y = [\tilde{C}_{s,2} \quad 0_{2 \times 2} \quad \tilde{C}_{u,3}] \tilde{x},$$

where the block matrix $\tilde{C}_{u,3} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$.

According to KRCC the system (14) is controllable, since $\text{rank } M_c|_{(\tilde{A}_u, \tilde{B}_u)} = 6 = \dim \tilde{x}$. Similarly to the stable equilibrium point, the weighting matrixes were specified by the help of the Bryson's rule. Therefore, $Q = \text{diag}(Q_{ii})$ and $R = \text{diag}(R_{jj})$, where $Q_{11} = 0.1^{-2}$, $Q_{22} = (10 \cdot \pi/180)^{-2}$, $Q_{33} = (0.15)^{-2}$, $Q_{44} = (150 \cdot \pi/180)^{-2}$, $Q_{55} = (10 \cdot \pi/$

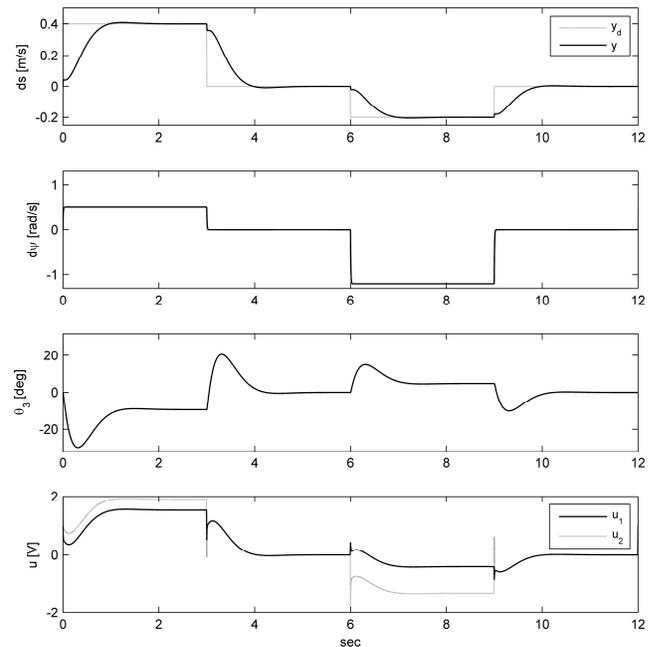


Figure 2. Simulation results. Control performance around the stable equilibrium point.

$180)^{-2}$, $Q_{66} = (45 \cdot \pi/180)^{-2}$, and $R_{11} = R_{22} = 3^{-2}$. Solving the CARE the optimal feedback gain $K^u = (K_1^u, K_2^u)$ takes the form (15), while the reference tracking matrices are given by (16):

$$K_1^u = \begin{bmatrix} -7.46 & -5.47 & -11.1 & -0.28 \\ -7.46 & -5.47 & -11.1 & -0.28 \end{bmatrix}, \quad (15)$$

$$K_2^u = \begin{bmatrix} -4.35 & -0.634 \\ +4.35 & +0.634 \end{bmatrix},$$

$$N_x^u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \quad N_u^u = 0_{2 \times 2}. \quad (16)$$

B. Simulation results

Simulation results are shown in Figure 3. From the top, the first is the linear position s of the robot, the second is the yaw angle ψ , the third is the tilt angle θ_3 of the inner body, while the last one is the applied voltage to the motors. The simulation was started with the following initial conditions: $x_0 = [0.05, 8 \cdot \frac{\pi}{180}, 0, 0, 20 \cdot \frac{\pi}{180}, 0]$. It can be seen, that the calculated optimal feedback gain asymptotically stabilizes the closed loop system, and both the stabilization of the inner body and the position control is ensured. The following reference signals were applied in the simulation: $s_d = \{0, 0.1, -0.05, 0\}$ [m] and $\psi_d = \{0, 1.5, 0, -0.7\}$ [rad].

VII. STATE ESTIMATION WITH KALMAN-FILTER

The Kalman filter is used to estimate the tilt angle θ_3 of the inner body (second element of \tilde{x}). Since the accelerometer measures the projection of gravity vector onto its axes, the angle is given by $\theta_{3,acc} = \text{atan}(\frac{a_y}{a_x})$ [16]. Unfortunately,

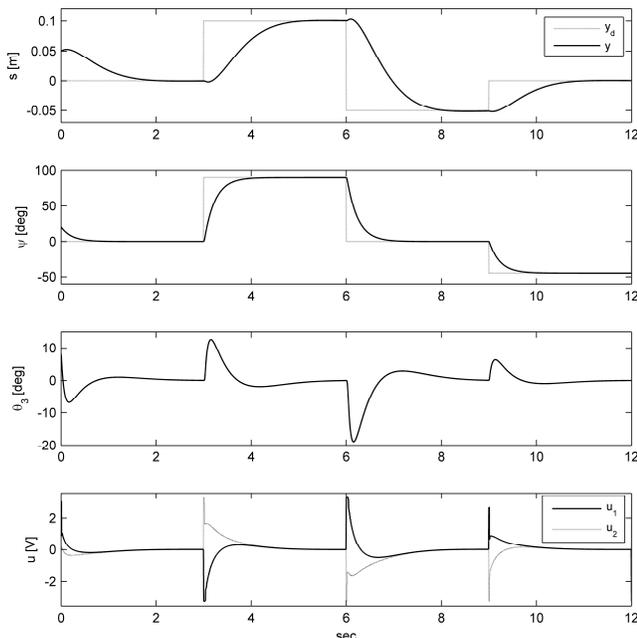


Figure 3. Simulation results. Control performance around the unstable equilibrium point.

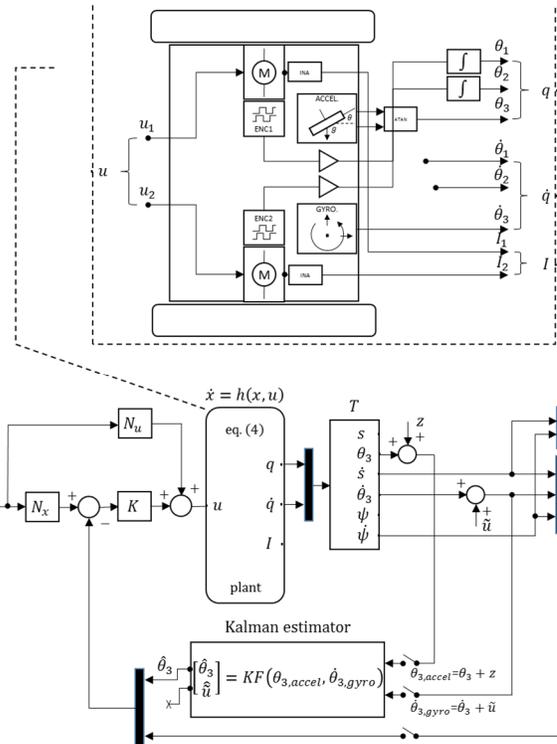


Figure 4. Detailed LQG control structure.

$\theta_{3,acc}$ is very noisy, in the most control methods useless and it can't be considered as an accurate derived quantity at high frequency rates of rotation, because the accelerometer measures both the static acceleration of the gravity and the dynamic acceleration of the robot as well. Thus, it is common to consider the gyroscope and accelerometer as a noisy linear system and use the Kalman filter to estimate the state vector. The corresponding state-space equation is given by:

$$\xi_{k+1} = \begin{bmatrix} 1 & -1/f_s \\ 0 & 1 \end{bmatrix} \xi_k + \begin{bmatrix} 1/f_s \\ 0 \end{bmatrix} \rho_k + v_k \quad (17)$$

$$\gamma_k = [1 \quad 0] \xi_k + z_k,$$

where the state vector $\xi = (\theta_3, \tilde{u})^T$ consists of the inclination angle θ_3 [rad], and the bias of the gyroscope \tilde{u} [rad/s].

Furthermore, the input of the linear system is the angular velocity $\rho = \dot{\theta}_3$ [rad/s] (measured by the gyroscope), while the output of the system is the derived angle $\gamma = \theta_{3,acc}$ [rad] from the pure accelerometer measurements. The covariance matrixes that characterize the measurement and state noises were defined based on offline measurements.

VIII. THE LQG CONTROL STRUCTURES

According to the separation principle the LQG control strategies are elaborated as follows. Around the stable equilibrium point the state feedback $u_k = -K(\hat{\theta}_3, \dot{s}, \dot{\theta}_3, \dot{\psi})$ ensures asymptotic stability of the closed loop system, where $\hat{\theta}_3$ denotes the Kalman-filter based estimation of the tilt angle of the inner body and the optimal control gain K is defined by (11). The detailed control structure is depicted in Figure 4.

While around the unstable equilibrium point, the state feedback $u_k = -K(s, \hat{\theta}_3, \dot{s}, \hat{\theta}_3, \psi, \dot{\psi})$ provides the stabilization of the system, where the optimal control law is defined by (15) (the figure of the control structure is neglected since it's almost the same as Figure 4).

IX. IMPLEMENTATION RESULTS

The control algorithm was coded in C language. MCU2 was programmed to work as an inertial measurement unit (IMU). Its main task is to read the data of sensors (from accelerometer and gyroscope through SPI peripheral), and send a package consisting of $\theta_{3,acc}$, $\hat{\theta}_3$, and $\hat{\theta}_3$ to MCU1 continuously (in every $T_s = 10\text{ ms}$), where $\theta_{3,acc}$ indicates the inclination angle determined based on the pure accelerations measured by the accelerometer, $\hat{\theta}_3$ denotes the angular velocity of the pendulum measured by the gyroscope, while $\hat{\theta}_3$ indicates the Kalman estimation of the inclination angle. MCU1 executes the chosen control algorithm based on the collected measurements. It receives the package $(\theta_{3,acc}, \hat{\theta}_3, \hat{\theta}_3)$ from MCU2 and extends it with the instantaneous position and velocity of the robot (s, \dot{s}) based on the measurements collected from the incremental encoders. Once the measurements are updated the chosen control algorithm updates the duty cycle of the PWM generator. Furthermore, the measurements are sent through the Bluetooth module with the frequency $f_s = 100\text{ Hz}$. A GUI written in MATLAB records the measurements.

The control performance was tested in different environments, i.e., both static and dynamic behavior of the chosen control algorithm. During the test of the dynamic behavior the robot was pushed away from its stable position.

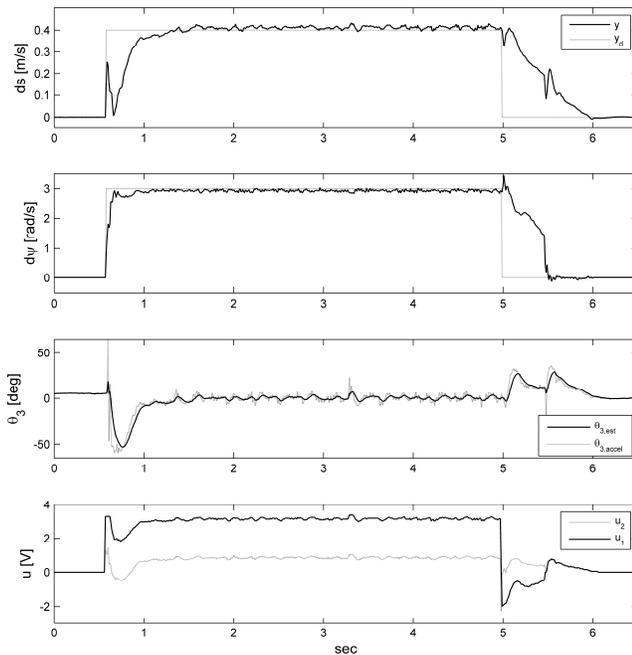


Figure 5. Implementation results. Control performance around the stable equilibrium point. Reference signals: $\dot{s}_d = 0.4\text{ [m/s]}$, $\dot{\psi}_d = 3\text{ [rad/s]}$.

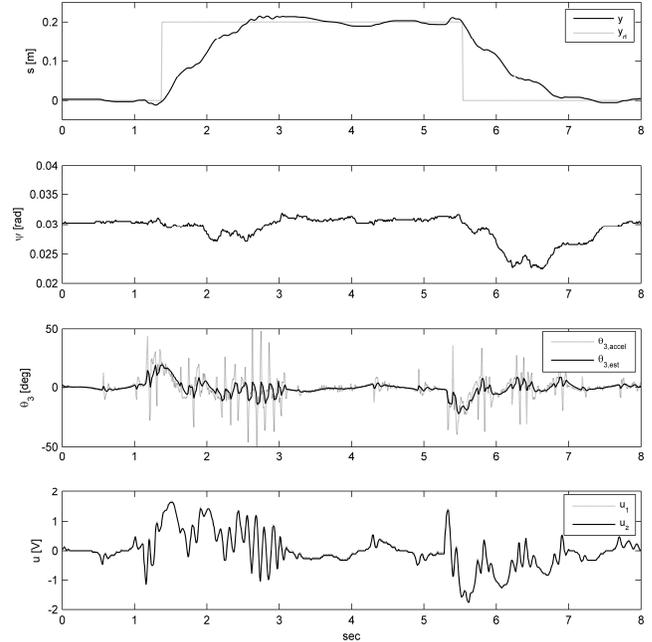


Figure 6. Implementation results. Control performance around the unstable equilibrium point. Reference signal: $s_d = 0.2\text{ [m]}$.

A. Control around the stable equilibrium point

The control performance is depicted in Figure 5. From the top, the first is the linear speed \dot{s} , the second is the change of yaw $\dot{\psi}$, the third is the angle of the inner body θ_3 , while the last one is the applied voltage. It can be seen that the implemented LQG control strategy successfully suppressed the oscillation of the inner body and ensured the speed control of the robot as well.

B. Control around the unstable equilibrium point

The implementation results are depicted in Figure 6, 7 and 8. From the top, the first is the linear position s , the second is the yaw angle ψ , the third is the angle of the pendulum θ_3 , while the last one is the applied voltage. Based on the measurements, it can be concluded that the elaborated LQG control strategy ensured both the stabilization of the inner body around the unstable equilibrium point, and the position control of the robot as well.

In Figure 6 $s_d = 0.2\text{ [m]}$ reference signal was applied and the stabilization took approximately 2.5 seconds. The noisy measurements of the tilt range and the importance of the Kalman filtration can be also observed in the figures, since $\pm 50\text{ [}^\circ\text{]}$ oscillations were not present evidently during the balancing. The static behavior of the balancing was also investigated. It can be seen in Figure 7 that if there is no external perturbation the balancing range is in $\pm 5\text{ [}^\circ\text{]}$, while the position control tracks the reference with $\pm 5\text{ [mm]}$ error.

Finally, the dynamic behavior of the implemented control strategy was also tested on the robot, i.e., the robot was pushed away several times from its equilibrium state (the external perturbations are indicated with arrows). As it can be seen in Figure 8, both the balance and the stabilization of the robot

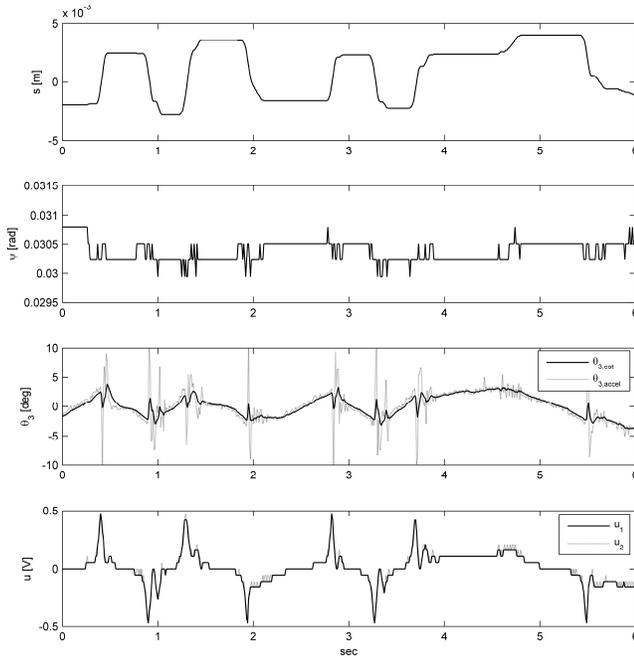


Figure 7. Implementation results. Control performance around the unstable equilibrium point. Static behavior.

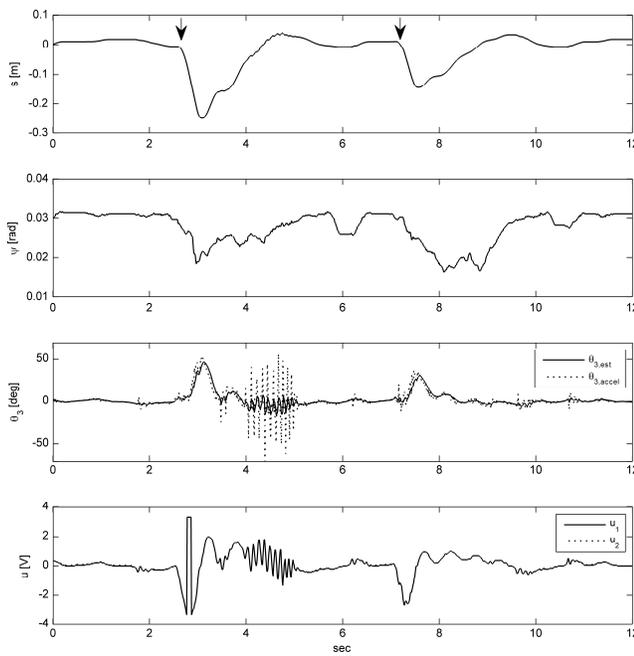


Figure 8. Implementation results. Control performance around the unstable equilibrium point. Dynamic behavior.

was successfully solved. First the robot was pushed about 25 [cm] away while the second time about 15 [cm], the stabilization was achieved approximately in 2.5 seconds.

X. CONCLUSION AND FUTURE WORK

In this paper, LQG control strategies were elaborated to a two-wheeled mobile pendulum system. Namely, around the stable equilibrium state of the plant an anti-sway speed

controller-, while around the unstable equilibrium point a position controller which ensures the stabilization of the inner body were developed and implemented on the real robot. The asymptotic stability of the closed loop system was proved based on simulation and measurement results. According to the implementation results it is concluded that the elaborated LQG control strategies produce satisfying and competitive control performances. These experiments form our initial results in the investigation of the control performances of different modern control methods. Future work will involve the identification of the unknown parameters, the validation of the developed control strategies and the development of modern (such as nonlinear, fuzzy) control algorithms to the system.

ACKNOWLEDGMENT

We acknowledge the financial support of this work by the Hungarian State and the European Union under the TÁMOP-4.2.2.D-15/1/KONV-2015-0002 project entitled "Development of Smart technologies for High-tech industry".

APPENDIX

The Lagrange function of the system:

$$\begin{aligned} \mathcal{L} = & \sum_{i=1}^2 \left(\frac{3}{4} m_w r^2 + \frac{1}{8} m_b r^2 + \frac{l^2 r^2}{2d^2} m_b \sin^2 \theta_3 + \frac{1}{2} J_B \frac{r^2}{d^2} + \frac{1}{2} k^2 J_r \right) \dot{\theta}_i^2 \\ & + \sum_{i=1}^2 \left(\frac{1}{2} m_b l r \cos \theta_3 - k^2 J_r \right) \theta_3 \dot{\theta}_i \\ & + \left(\frac{1}{4} m_b r^2 - \frac{l^2 r^2}{d^2} m_b \sin^2 \theta_3 - J_B \frac{r^2}{d^2} \right) \dot{\theta}_1 \dot{\theta}_2 \\ & + \left(\frac{1}{2} m_b l^2 + \frac{1}{2} J_A + k^2 J_r \right) \dot{\theta}_3^2 - 2m_w g r - m_b g (r - l \cos \theta_3). \end{aligned} \quad (\text{A})$$

Elements of the inertia matrix $M(q) = (m_{ij})_{3 \times 3}$:

$$\begin{aligned} m_{11} &= \frac{3}{2} m_w r^2 + \frac{1}{4} m_b r^2 + k^2 J_r + \frac{l^2 r^2}{d^2} m_b \sin^2 \theta_3 + J_B \frac{r^2}{d^2}, \\ m_{22} &= m_{11}, m_{33} = m_b l^2 + J_A + 2k^2 J_r, \\ m_{12} &= m_{21} = \frac{1}{4} m_b r^2 - \frac{l^2 r^2}{d^2} m_b \sin^2 \theta_3 - J_B \frac{r^2}{d^2}, \\ m_{13} &= m_{23} = m_{31} = m_{32} = \frac{1}{2} m_b l r \cos \theta_3 - k^2 J_r. \end{aligned} \quad (\text{B})$$

The elements of $V(q, \dot{q}) = (v_1, v_2, v_3)^T$:

$$\begin{aligned} v_1 &= 2 \frac{l^2 r^2}{d^2} m_b \sin \theta_3 \cos \theta_3 \dot{\theta}_3 (\dot{\theta}_1 - \dot{\theta}_2) - \frac{1}{2} m_b l r \sin \theta_3 \dot{\theta}_3^2, \\ v_2 &= 2 \frac{l^2 r^2}{d^2} m_b \sin \theta_3 \cos \theta_3 \dot{\theta}_3 (\dot{\theta}_2 - \dot{\theta}_1) - \frac{1}{2} m_b l r \sin \theta_3 \dot{\theta}_3^2, \\ v_3 &= -\frac{l^2 r^2}{d^2} m_b \sin \theta_3 \cos \theta_3 (\dot{\theta}_1 - \dot{\theta}_2)^2 + m_b g l \sin \theta_3. \end{aligned} \quad (\text{C})$$

The block matrices of equation (9):

$$\begin{aligned} \tilde{A}_{s,21} &= \begin{bmatrix} 0 & -0.08 \\ 0 & -136.5 \end{bmatrix}, \tilde{A}_{s,22} = \begin{bmatrix} -25.9 & 0.8 \\ 2338 & -73.6 \end{bmatrix}, \\ \tilde{A}_{s,33} &= \begin{bmatrix} 0 & 1 \\ 0 & -56 \end{bmatrix}, \tilde{B}_{s,2} = \begin{bmatrix} 3 & 3 \\ -279.6 & -279.6 \end{bmatrix}, \\ \tilde{B}_{s,3} &= \begin{bmatrix} 0 & 0 \\ -73.9 & 73.9 \end{bmatrix}, \tilde{C}_{s,2} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \tilde{C}_{s,3} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (\text{D})$$

TABLE II. NOTATION OF THE ROBOT PARAMETERS

Symbol	Name	Value [SI Unit]
r	Wheel radius	$3.15 \cdot 10^{-2}$
m_w	Mass of the wheels	$31.6 \cdot 10^{-3}$
l	Distance between the COG and the wheel axle	$8.36 \cdot 10^{-3}$
m_b	Mass of the inner body	$360.4 \cdot 10^{-3}$
d	Distance between the wheels	$177 \cdot 10^{-3}$
J_A	Moment of inertia of the inner body about the wheel axle \mathcal{A}	$81367 \cdot 10^{-9}$
J_B	Moment of inertia of the inner body about the axis \mathcal{B}	$574620 \cdot 10^{-9}$
k	Gear ratio	64
J_r	Rotor inertia	$0.12 \cdot 10^{-7}$
R	Terminal resistance	2.3
L	Rotor inductance	$26 \cdot 10^{-6}$
k_M	Torque constant	$2.05 \cdot 10^{-3}$
k_E	Back-EMF constant	$2.05 \cdot 10^{-3}$
b	Viscous friction coefficient between body - motor	$2.1 \cdot 10^{-5}$
f_v	Viscous friction coefficient between wheels - ground	$1.8 \cdot 10^{-4}$

REFERENCES

- [1] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson Prentice Hall, 2014, ISBN: 978-0-13349-659-8.
- [2] A. Divelbiss and J. Wen, "Trajectory tracking control of a car-trailer system," *IEEE Transactions on Control Systems Technology*, vol. 5, 1997, pp. 269 - 278, doi: 10.1109/87.572125.
- [3] O. Araar and N. Aouf, "Full linear control of a quadrotor UAV, LQ vs H_∞ ," *UKACC International Conference on Control (CONTROL)*, 2014, pp. 133 - 138, doi: 10.1109/CONTROL.2014.6915128.
- [4] J.-K. Ji and S.-K. Sul, "Kalman filter and LQ based speed controller for torsional vibration suppression in a 2-mass motor drive system," *IEEE Transactions on Industrial Electronics*, vol. 42, 2002, pp. 564 - 571, doi: 10.1109/41.475496.
- [5] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, 2004, pp. 2451 - 2456, doi: 10.1109/IROS.2004.1389776.
- [6] S. Jeong and T. Takahashi, "Wheeled inverted pendulum type assistant robot: inverted mobile, standing, and sitting motions," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, 2007, pp. 1932 - 1937, doi: 10.1109/IROS.2007.4398961.
- [7] S. Zhiyu and L. Daliang, "Balancing control of a unicycle riding," *29th Chinese Control Conference (CCC)*, 2010, pp. 3250 - 3254, ISBN: 978-1-4244-6263-6.
- [8] L. Yi-bo, L. Wan-zhu, and S. Qi, "Improved LQG control for small unmanned helicopter based on active model in uncertain environment," *International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 289 - 292, doi: 10.1109/ICECC.2011.6067810.
- [9] S. Nagaya, T. Morikawa, I. Takami, and G. Chen, "Robust LQ control for parallel wheeled inverted pendulum," *3rd Australian Control Conference (AUCC)*, 2013, pp. 189 - 194, doi: 10.1109/AUCC.2013.6697271.
- [10] Appl-DSP. Video demonstration of the robot. [Online]. Available from: <http://appl-dsp.com/lqg-and-fuzzy-control-of-a-mobile-wheeled-pendulum>
- [11] A. Salerno and J. Angeles, "A New Family of Two-Wheeled Mobile Robots: Modeling and Controllability," *IEEE Transactions on Robotics*, vol. 23, 2007, pp. 169 - 173, doi: 10.1109/TRO.2006.886277.
- [12] B. Cazzolato et al., "Modeling, simulation and control of an electric diwheel," *Australasian Conference on Robotics and Automation (ACRA 2011)*, 2011, pp. 1-10, ISBN: 978-0-9807-4042-4.
- [13] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer-Verlag London, 2000, ISBN: 978-1-85233-221-1.
- [14] Á. Odry, I. Harmati, Z. Király, and P. Odry, "Design, realization and modeling of a two-wheeled mobile pendulum system," *14th International Conference on Instrumentation, Measurement, Circuits and Systems (IMCAS '15)*, 2015, pp. 75-79, ISBN: 978-1-61804-315-3.
- [15] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *Tech. Rep. TR 95-041*, Department of Computer Science, University of North Carolina, USA, 2001.
- [16] STMicroelectronics, "Tilt measurement using a low-g 3-axis accelerometer," *Application note AN3182*, 2010.

Estimation of Nuclear Reactor Vessel Water Level in Severe Accidents Using Cascaded Fuzzy Neural Networks

Dong Yeong Kim, Kwae Hwan Yoo, Geon Pil Choi, Man Gyun Na

Department of Nuclear Engineering

Chosun University

Gwangju, Republic of Korea

e-mail: doo891221@naver.com, yooqh@naver.com, zzangcgp7@naver.com, magyna@chosun.ac.kr

Abstract—The world’s concern about nuclear reactor safety has increased considerably since the Fukushima accident. In case of most severe accidents, the nuclear reactor vessel water level cannot be measured. But, if the cascaded fuzzy neural network (CFNN) is used, under the event of severe accidents it might be possible to estimate the nuclear reactor vessel water level. The cascaded fuzzy neural network model can be used to estimate the nuclear reactor vessel water level value through the process of adding fuzzy neural networks (FNNs) repeatedly. The developed cascaded fuzzy neural network model is sufficiently accurate to be used to estimate the nuclear reactor vessel water level. Therefore, the developed cascaded fuzzy neural network model will be helpful for providing effective information for operators in severe accident situations.

Keywords—Cascaded fuzzy neural network (CFNN); Fuzzy neural network (FNN); Nuclear reactor vessel water level.

I. INTRODUCTION

Recently, the world’s concern about nuclear reactor safety has increased considerably since the Fukushima accident. The cause of these concerns and interest is because the operators do not quickly check the status of the plant in appropriate response to each situation.

The reactor vessel water level is essential information for confirming the cooling capability of the nuclear reactor core, to prevent the reactor core from melting down and to manage severe accidents effectively. In particular, decay heat is continuously generated in the reactor core after reactor shutdown. Therefore, it is important to estimate the reactor vessel water level to make provisions against severe accidents.

Many artificial intelligence techniques have been applied successfully to nuclear engineering areas, such as signal validation [1]–[3], plant diagnostics [4][5], event identification [6]–[9], etc. In this paper, a cascaded fuzzy neural network (CFNN) model is proposed to estimate the reactor vessel water level, which has a direct impact on the important times (time approaching the core exit temperature exceeding 1200°F, core uncover time, reactor vessel failure time, etc.). The CFNN can be used to estimate the nuclear reactor vessel water level through the process of adding fuzzy neural networks (FNNs) repeatedly. To estimate the water level and the loss of coolant accident (LOCA) break size, other measured signals were used. The LOCA break

size is not a measured variable. Instead, it is an estimated variable using the trend data for a short time early in the event preceding a severe accident. The LOCA classification algorithm for determining the LOCA position and LOCA break size estimation algorithm were explained in previous papers [10]–[12]. Because the LOCA break size could be estimated accurately, the LOCA break size was used as an input variable for estimating the reactor vessel water level. The obtained numerical simulation data was obtained and verified by simulating severe accident scenarios for the Optimized Power Reactor 1000 (OPR1000) using MAAP4 code [13].

Section II explains the methodology of CFNN including fuzzy inference system (FIS) and its training. Section III describes its application to estimating the water level in the reactor vessel.

II. CASCADED FUZZY NEURAL NETWORKS

A. Fuzzy inference system

The FIS uses the conditional rules that are comprised of an *if-then* rules of a pair of antecedent and consequent [14]. This study uses the Takagi-Sugeno-type FIS [15], which does not need the defuzzifier in the output terminal because its output is a real value. The Takagi-Sugeno-type FIS consists of three basic components without the defuzzifier block, differently from the Mamdani-type FIS shown in Figure 1.

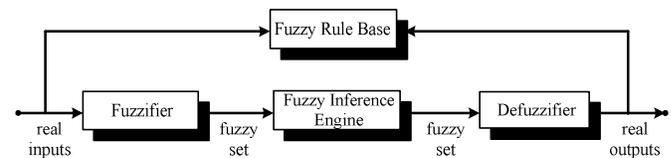


Figure 1. Fuzzy inference system (Mamdani-type FIS)

In the FIS, an arbitrary i^{th} fuzzy rule can be expressed as follows (first-order Takagi-Sugeno-type):

$$\text{If } x_1(k) \text{ is } A_{i1}(k) \text{ AND } \dots \text{ AND } x_m(k) \text{ is } A_{im}(k), \quad (1) \\ \text{then } y^i(k) \text{ is } f^i(x_1(k), \dots, x_m(k))$$

where

x_1, \dots, x_m : FIS input values

m = number of input variables

$A_{i1}(k), \dots, A_{im}(k)$: fuzzy sets of the i^{th} fuzzy rule

y^i : output of the i^{th} fuzzy rule

$$f^i(x_1(k), \dots, x_m(k)) = \sum_{j=1}^m q_{ij} x_j(k) + r_i \quad (2)$$

q_{ij} : weight of the j^{th} fuzzy input variable

r_i : bias of the i^{th} fuzzy rule

Because the function $f^i(x(k))$ is expressed as the first-order polynomial of input variables, FIS is called the first-order Takagi-Sugeno-type FIS in (2). The number of N input and output training data $\mathbf{z}^T(k) = (\mathbf{x}^T(k), y(k))$ (where $\mathbf{x}^T(k) = (x_1(k), x_2(k), \dots, x_m(k))$, $k = 1, 2, \dots, N$) are assumed to be available, and the input and output variables are normalized. In general, there is no special restriction on the shape of the membership functions. In this study, the symmetric Gaussian membership function is used to reduce the number of parameters to be optimized.

$$A_{ij}(x_j(k)) = e^{-(x_j(k) - c_{ij})^2 / 2s_{ij}^2} \quad (3)$$

The FIS output $\hat{y}(k)$ is calculated by weight-averaging the fuzzy rule outputs $y^i(k)$ as follows:

$$\begin{aligned} \hat{y}(k) &= \sum_{i=1}^n \bar{w}^i(k) y^i(k) = \sum_{i=1}^n \bar{w}^i(k) f^i(\mathbf{x}(k)) \\ &= \mathbf{w}^T(k) \mathbf{q} \end{aligned} \quad (4)$$

where

$$\bar{w}^i(k) = \frac{w^i(x(k))}{\sum_{i=1}^n w^i(x(k))} \quad (5)$$

$$w^i(k) = \prod_{j=1}^m A_{ij}(x_j(k)) \quad (6)$$

n : number of fuzzy rules

$$\mathbf{q} = [q_{11} \dots q_{n1} \dots q_{1m} \dots q_{nm} r_1 \dots r_n]^T$$

$$\mathbf{w}(k) = [\bar{w}^1(k) x_1(k) \dots \bar{w}^n(k) x_1(k) \dots$$

$$\bar{w}^1(k) x_m(k) \dots \bar{w}^n(k) x_m(k) \bar{w}^1(k) \dots \bar{w}^n(k)]^T$$

The vector \mathbf{q} is called a consequent parameter vector that has $(m+1)n$ dimensions, and the vector $\mathbf{w}(k)$ consists of input data and membership function values. The estimated output for a total of N input and output data pairs induced from (4) can be expressed as follows:

$$\hat{\mathbf{y}} = \mathbf{W} \mathbf{q} \quad (7)$$

where

$$\hat{\mathbf{y}} = [\hat{y}(1) \hat{y}(2) \dots \hat{y}(N)]^T$$

$$\mathbf{W} = [\mathbf{w}(1) \mathbf{w}(2) \dots \mathbf{w}(N)]^T$$

The matrix \mathbf{W} has $N \times (m+1)n$ dimensions. Figure 2 describes the calculation structure of the FNN model [16].

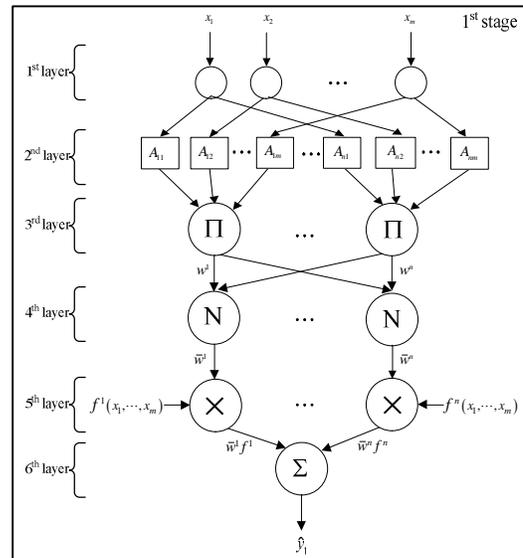


Figure 2. Fuzzy neural network (FNN)

B. FIS training

In this study, the FIS is optimized using the two combined methods of a genetic algorithm and a least squares method. The training data were used to develop the FNN model. The test data were used to verify the developed FNN model, and they are different from the training data set. The following fitness function for the genetic algorithm is proposed to minimize the maximum error and root mean square (RMS) error.

$$F = \exp(-\lambda_1 E_1 - \lambda_2 E_2) \quad (8)$$

where

$$E_1 = \sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} (y(k) - \hat{y}(k))^2}$$

$$E_2 = \max_k (y(k) - \hat{y}(k))$$

λ_1 : weighting value of RMS error

λ_2 : weighting value of maximum error

N_t : number of training data

The variable $y(k)$ is the actual output value, and $\hat{y}(k)$ is its value estimated using the FNN model. If the antecedent parameters are determined using a genetic algorithm through selection, crossover, and mutation, the resulting parameters appear similar to (7) as a first-order combination. Therefore, the consequent parameter \mathbf{q} can be calculated easily using the least squares method. That is, the consequent parameter \mathbf{q} is calculated to minimize an objective function. The objective function consists of the square error between the actual value $y(k)$ and its estimated value $\hat{y}(k)$, and it is expressed as follows:

$$\begin{aligned} J &= \sum_{k=1}^{N_t} (y(k) - \hat{y}(k))^2 = \sum_{k=1}^{N_t} (y(k) - \mathbf{w}^T(k)\mathbf{q})^2 \\ &= \frac{1}{2} (\mathbf{y}_t - \hat{\mathbf{y}}_t)^2 \end{aligned} \quad (9)$$

where

$$\mathbf{y}_t = [y(1) \ y(2) \ \dots \ y(N_t)]^T$$

$$\hat{\mathbf{y}}_t = [\hat{y}(1) \ \hat{y}(2) \ \dots \ \hat{y}(N_t)]^T$$

A solution for minimizing the above objective function can be obtained using the following equation:

$$\mathbf{y}_t = \mathbf{W}_t \mathbf{q} \quad (10)$$

where

$$\mathbf{W}_t = [\mathbf{w}(1) \ \mathbf{w}(2) \ \dots \ \mathbf{w}(N_t)]^T$$

The matrix \mathbf{W}_t has $N_t \times (m+1)n$ dimensions in (10). The parameter vector \mathbf{q} can be solved easily from the pseudo-inverse as follows:

$$\mathbf{q} = (\mathbf{W}_t^T \mathbf{W}_t)^{-1} \mathbf{W}_t^T \mathbf{y}_t \quad (11)$$

The parameter vector \mathbf{q} can be calculated from a series of input and output data pairs and their membership function

values because the matrix \mathbf{W}_t consists of input data and membership function values.

C. Cascaded fuzzy neural networks

The foregoing FNN is composed of the fuzzy logic and neural network theory. Most of the existing FNN models have been proposed to implement different types of single-stage fuzzy reasoning mechanisms. However, single-stage fuzzy reasoning is only the most simple among a human being's various types of reasoning mechanisms. Syllogistic fuzzy reasoning, where the consequence of a rule in one reasoning stage is passed to the next stage as a fact, is essential to effectively build up a large scale system with high level intelligence [17]. Therefore, it is described by applying these techniques in this paper.

The CFNN model contains two or more inference stages where each stage corresponds to a single-stage FNN module. The architecture of the CFNN is shown in Figure 3.

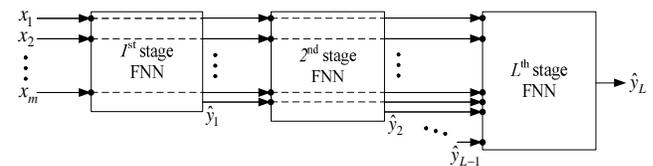


Figure 3. Cascaded fuzzy neural network (CFNN)

The CFNN can be used to estimate the target value through the process of adding FNN repeatedly. In CFNN method, the first stage FNN is the same as the FNN of Figure 2. The second stage FNN uses the initial input variables and the output variable of the first stage FNN as input variable. Therefore, this process is repeated L times to find the optimum value if over-fitting phenomena do not appear.

Similarly to (1), an arbitrary i^{th} rule of the CFNN can be expressed as (12):

$$\begin{aligned} \text{Stage 1} & \left[\begin{array}{l} \text{If } x_1(k) \text{ is } A_{i1}^1(k) \text{ AND } \dots \text{ AND } x_m(k) \text{ is } A_{im}^1(k), \\ \text{then } \hat{y}_1^i(k) \text{ is } f_1^i(x_1(k), \dots, x_m(k)) \end{array} \right] \\ \text{Stage 2} & \left[\begin{array}{l} \text{If } x_1(k) \text{ is } A_{i1}^2(k) \text{ AND } \dots \text{ AND } x_m(k) \text{ is } A_{im}^2(k) \\ \text{AND } \hat{y}_1(k) \text{ is } A_{i(m+1)}^2(k), \\ \text{then } \hat{y}_2^i(k) \text{ is } f_2^i(x_1(k), \dots, x_m(k), \hat{y}_1(k)) \end{array} \right] \\ & \vdots \\ \text{Stage } L & \left[\begin{array}{l} \text{If } x_1(k) \text{ is } A_{i1}^L(k) \text{ AND } \dots \text{ AND } x_m(k) \text{ is } A_{im}^L(k), \\ \text{AND } \hat{y}_1(k) \text{ is } A_{i(m+1)}^L(k) \text{ AND } \dots \text{ AND } \hat{y}_{(L-1)}(k) \text{ is } A_{i(m+L-1)}^L(k), \\ \text{then } \hat{y}_L^i(k) \text{ is } f_L^i(x_1(k), \dots, x_m(k), \hat{y}_1(k), \dots, \hat{y}_{(L-1)}(k)) \end{array} \right] \end{aligned} \quad (12)$$

where L is the stage number of CFNN. The CFNN model is trained sequentially at each FNN module in the same way as explained in subsections II. B and II. C.

III. APPLICATION TO ESTIMATING THE NUCLEAR REACTOR VESSEL WATER LEVEL

The proposed CFNN model was applied to estimating the water level in the reactor vessel. The training and test data of the proposed model was acquired by simulating the severe accident scenarios using the MAAP4 code concerning the OPR1000 nuclear power plant.

The simulation data is divided into the break position and break size of LOCA. The break position was divided into hot-leg LOCA, cold-leg LOCA and SGTR, and the break size was divided into a total of 200 steps.

The LOCA position was identified completely and the LOCA break size was estimated accurately in previous studies [10]-[12], with an approximately 1% error level. Therefore, the LOCA break size signal, which is an input signal to the FNN model, is assumed to be estimated from the algorithms of previous studies. Through the simulations, a total of 600 cases of severe accident scenarios are obtained. This data is composed of 200 pieces of hot-leg LOCA, 200 pieces of cold-leg LOCA and 200 pieces of SGTR.

The test data were different from the data that were used to develop the CFNN model, and consisted of the time that elapsed after reactor shutdown, the estimated LOCA break size, and the containment pressure. At this study, 300 data points in each of the LOCA break positions, namely, hot-leg and cold-leg LOCA, and SGTR, were selected as test data points.

The parameter values that are concerned with the genetic algorithm and the FIS are as follow: the crossover 100%, the mutation probability is 5%, and the population size is 20.

Table I shows the performance results that were obtained with the CFNN model for the break positions of hot-leg, cold-leg and SGTR, respectively.

TABLE I. PERFORMANCE OF THE CFNN MODEL

RMS error(m)	All break sizes		
	Hot-leg LOCA	Cold-leg LOCA	SGTR
2 fuzzy rules	0.1721	0.2130	0.3399
3 fuzzy rules	0.2280	0.1895	0.3351
5 fuzzy rules	0.2255	0.7045	0.3233
7 fuzzy rules	0.1380	13.6493	0.3261

For the test data of the hot-leg LOCA, the RMS errors were approximately 0.17m, 0.23m, 0.23m, and 0.14m for the CFNN model with 2, 3, 5, and 7 fuzzy rules, respectively. And the RMS errors were approximately 0.21m, 0.19m, 0.70m, and 13.65m for the test data of the cold-leg LOCA and 0.34m, 0.34m, 0.32m, and 0.33m for the test data of the SGTR for the CFNN model with 2, 3, 5, and 7 fuzzy rules, respectively. Therefore, the CFNN model with 7 fuzzy rules proved to be the most accurate for estimating the nuclear reactor vessel water level in hot-leg

LOCA and the CFNN model for cold-leg LOCA is 3 fuzzy rules, while the CFNN model with 5 fuzzy rules was shown to be the most accurate for estimating the nuclear reactor vessel water level in SGTR.

The CFNN models have been shown to be capable of accurately estimating the nuclear reactor vessel water level in case of a severe accident.

IV. CONCLUSION

In this study, a CFNN model was developed to estimate the nuclear reactor vessel water level in severe accident. The developed CFNN model is verified based on the simulation data of OPR1000 using MAAP4 code. The simulation results show that the performance of the developed CFNN model is quite accurate with about approximately 2% error. The developed CFNN model will be helpful for providing effective information for operators in severe accident situations.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2012M2B2B1055611).

REFERENCES

- [1] J. W. Hines, D. J. Wrest, and R. E. Uhrig, "Signal validation using an adaptive neural fuzzy inference system," *Nucl. Technol.*, vol. 119, no. 2, pp. 181-193, Aug. 1997.
- [2] M. G. Na, "A neuro-fuzzy inference system for sensor failure detection using wavelet denoising, PCA and SPRT," *J. Korean Nucl. Soc.*, vol. 33, no. 5, pp. 483-497, Oct. 2001.
- [3] J. Garvey, D. Garvey, R. Seibert, and J. W. Hines, "Validation of on-line monitoring techniques to nuclear plant data," *Nucl. Eng. Tech.*, vol. 39, no. 2, pp. 149-158, Apr. 2007.
- [4] E. B. Bartlett and R. E. Uhrig, "Nuclear power plant diagnostics using an artificial neural network," *Nucl. Technol.*, vol. 97, pp. 272-281, March 1992.
- [5] M. Marseguerra and E. Zio, "Fault diagnosis via neural networks: The Boltzmann machine," *Nucl. Sci. Eng.*, vol. 117, no. 3, pp. 194-200, July 1994.
- [6] Y. Gyu No, J. H. Kim, M. G. Na, D. H. Lim, and K.-I. Ahn, "Monitoring Severe Accidents Using AI Techniques," *Nucl. Eng. Technol.*, vol. 44, no. 4, pp. 393-404, May 2012.
- [7] M. G. Na, et al., "Prediction of major transient scenarios for severe accidents of nuclear power plants," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 2, pp. 313-321, April 2004.
- [8] S. W. Cheon and S. H. Chang, "Application of neural networks to a connectionist expert system for transient identification in nuclear power plants," *Nucl. Technol.*, vol. 102, no. 2, pp. 177-191, May 1993.
- [9] Y. Bartal, J. Lin, and R. E. Uhrig, "Nuclear power plant transient diagnostics using artificial neural networks that allow "don't-know" classifications," *Nucl. Technol.*, vol. 110, no. 3, pp. 436-449, June 1995.
- [10] S. H. Lee, Y. G. No, M. G. Na, K.-I. Ahn, and S.-Y. Park, "Diagnostics of Loss of Coolant Accidents Using SVC and GMDH Models," *IEEE Trans. Nucl. Sci.*, vol. 58, no. 1, pp. 267-276, Feb. 2011.

- [11] M. G. Na, W. S. Park, and D. H. Lim, "Detection and Diagnostics of Loss of Coolant Accidents Using Support Vector Machines," IEEE Trans. Nucl. Sci., vol. 55, no. 1, pp. 628-636, Feb. 2008.
- [12] M. G. Na, S. H. Shin, D. W. Jung, S. P. Kim, J. H. Jeong, and B. C. Lee, "Estimation of Break Location and Size for Loss of Coolant Accidents Using Neural Networks," Nucl. Eng. Des., vol. 232, no. 3, pp. 289-300, Aug. 2004.
- [13] R. E. Henry, et al., MAAP4 – Modular Accident Analysis Program for LWR Power Plants, User's Manual, Fauske and Associates, Inc., vol. 1, 2, 3, and 4, 1990.
- [14] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," Int. J. Man-Machine Studies, vol. 7, pp. 1-13, 1975
- [15] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Systems, Man, Cybern., vol. SMC-1, no. 1, pp. 116-132, Jan./Feb. 1985.
- [16] D. Y. Kim, K. H. Yoo, J. H. Kim, M. G. Na, S. Hur, and C-H. Kim, "Prediction of Leak Flow Rate Using Fuzzy Neural Networks in Severe Post-LOCA Circumstances," IEEE Trans. Nucl. Sci., Vol. 61, No. 6, pp. 3644-3652, Dec. 2014.
- [17] J. C. Duan and F .L. Chung, "Cascaded Fuzzy Neural Network Model Based on Syllogistic Fuzzy Reasoning," IEEE Trans. Fuzzy Systems, vol. 9, no. 2, pp.293-306, Apr. 2001.

Prediction of Golden Time Using SVM for Recovering SIS in Severe Post-LOCA Circumstances

Kwae Hwan Yoo, Dong Yeong Kim, Ju Hyun Back, Man Gyun Na

Dept. Nuclear Engineering of Chosun University
Chosun University

Gwangju, Republic of Korea

e-mails: yooqh@naver.com, doo891221@naver.com, magyna@chosun.ac.kr, bjh4210@naver.com

Abstract— After the Fukushima accident, the nuclear power plant (NPP) accident that occurred as a result of the East Japan Great Earthquake, the safety problem of NPPs has emerged as a global concern. As a result, many countries using nuclear energy are conducting research to improve the safety of NPPs. In this study, we predicted the golden time of safety injection system (SIS) recovery for accomplishing the reactor cold shutdown and preventing reactor vessel (RV) failure. The support vector machine (SVM) was used to predict the golden time for the SIS recovery in loss-of-coolant accident (LOCA) circumstances. If the golden time of SIS for accident recovery is predicted, the core will not be exposed through appropriate action. Also, the RV failure will be prevented by the cooling water injection even if the reactor core is exposed. These various golden time data are thought to be very useful to quickly deal with the actual accident.

Keywords—Golden time, Support vector machine, Loss of coolant accident, Core uncover, Reactor vessel failure.

I. INTRODUCTION

After the Fukushima accident, the nuclear power plant (NPP) accident that occurred as a result of the East Japan Great Earthquake, the safety problem of NPPs has emerged as a global concern. As a result, many countries using nuclear energy are conducting research to improve the safety of NPPs. In addition, the interest in severe accidents in nuclear power plants has been increasing. Nuclear power plants are designed in consideration of design basis accidents (DBAs). DBAs such as loss-of-coolant accident (LOCA) in NPP may lead to serious accidents that exceed the DBAs due to failure of safety systems. If the heat removal system is not working properly, the core uncover and the reactor vessel (RV) failure may be possible [1][2]. Several researches acquiring important information under severe accident using artificial intelligence methodologies have been conducted [3]–[5].

In this study, the golden time of SIS recovery for accomplishing the reactor cold shutdown and preventing RV failure according to LOCA break sizes were predicted by using the support vector machine (SVM) model when safety injection system (SIS) was not operating normally. The data was obtained by simulating severe accident scenarios for the Optimized Power Reactor 1000(OPR1000) using MAAP code.

Section II explains the methodology of SVM and its optimization. Section III describes accident scenarios applied in this study. Section IV shows the prediction performance of the SVM model and its results.

II. GOLDEN TIME PREDICTION USING SVM MODEL

The SVM was used to predict the golden time for the SIS recovery in LOCA circumstances. The SVM model can be applied to classification problem and regression analysis.

A. SVM method

The SVM model is an algorithm for learning linear classifiers. SVM is a learning system using a high dimensional feature space. It yields prediction functions that are expanded on a subset of support vectors. Support vector regression (SVR) is the most common application form of SVMs. SVR model is to map nonlinearly the original data \mathbf{x} into higher dimensional feature space and to conduct linear regression. Hence, given a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N \in R^m \times R$ where \mathbf{x}_i is the input vector to an SVR model, y_i is the actual output value, N is the total number of data points used to develop the SVR model, the SVR is based on the following regression function [6]:

$$y = f(\mathbf{x}) = \sum_{i=1}^N w_i \phi_i(\mathbf{x}) + b = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b \quad (1)$$

where

$$\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^T, \quad \boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_N]^T$$

The function ϕ_i is called feature, and parameters \mathbf{w} and b are support vector weight and bias. After the input vector \mathbf{x} is mapped into vector $\boldsymbol{\phi}(\mathbf{x})$ of a high dimensional kernel-induced feature space, the nonlinear regression model is turned into a linear regression model in the feature space. These parameters can be calculated by minimizing the following regularized risk function:

$$R(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \lambda \sum_{i=1}^N |y_i - f(\mathbf{x}_i)|_e \quad (2)$$

where

$$|y_i - f(\mathbf{x}_i)|_\epsilon = \begin{cases} 0 & \text{if } |y_i - f(\mathbf{x}_i)| < \epsilon \\ |y_i - f(\mathbf{x}_i)| - \epsilon & \text{otherwise} \end{cases} \quad (3)$$

The first term of (2) is weight vector norm which characterizes the complexity of the SVR models and the second term is an estimation error. The parameters λ and ϵ are user-defined parameters, and $|y_i - f(\mathbf{x}_i)|_\epsilon$ is called the ϵ -insensitive loss function [7]. The loss equals zero if the predicted value $f(\mathbf{x})$ is within an error level ϵ , and for all other predicted point outside the error level ϵ , the loss is equal to the magnitude of the difference between the predicted value and the error level ϵ (refer to Figure. 1).

Increasing the insensitivity zone means a reduction in the requirement for accuracy of the estimation and a decrease in the number of support vectors (SVs), leading to data compression. In addition, as understood from Figure 2, increasing the insensitivity zone has smoothing effects on modeling on highly noisy polluted data.

The regularization parameter λ of (2) is used to ensure good generalization of the SVR model. An increase in the regularization parameter penalizes larger error, which leads to a decrease in the estimation error. The decrease in the estimation error can also be achieved easily by increasing the weight vector norm of the first term of (2). However, an increase in the weight vector norm does not ensure good generalization of the SVR model. This generalization property is of particular interest to data-based model development because a good model is not a model that performs well on only training data but a model that performs well even on other data that is no training data.

In classical support vector regression, the proper value for the parameter ϵ is difficult to determine beforehand. Minimizing the regularized risk function of (2) is equivalent to minimizing the following constrained risk function:

$$R(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \lambda \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (4)$$

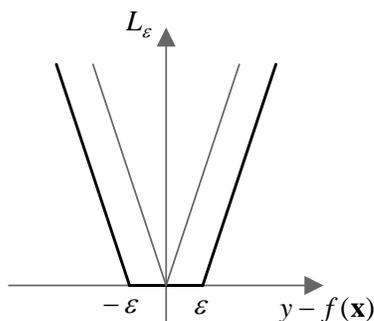


Figure 1. Linear sensitivity loss function.

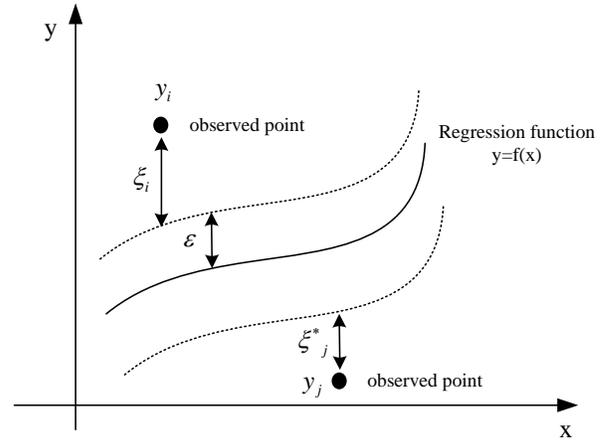


Figure 2. Parameters for the SVR models [8].

subject to the constraints

$$\begin{cases} y_i - \mathbf{w}^T \boldsymbol{\phi}(x) - b \leq \epsilon + \xi_i, & i = 1, 2, \dots, N \\ \mathbf{w}^T \boldsymbol{\phi}(x) + b - y_i \leq \epsilon + \xi_i^*, & i = 1, 2, \dots, N \\ \xi_i, \xi_i^* \geq 0, & i = 1, 2, \dots, N \end{cases} \quad (5)$$

The parameters $\boldsymbol{\xi} = [\xi_1 \ \xi_2 \ \dots \ \xi_N]^T$ and $\boldsymbol{\xi}^* = [\xi_1^* \ \xi_2^* \ \dots \ \xi_N^*]^T$ are the slack variables that represent the upper and lower constraints on the output of the system, and are positive values (refer to Figure. 2).

The constrained optimization problem of (4) can be solved by applying the Lagrange multiplier technique to (4) and (5), and using a standard quadratic programming technique [8], [9]. Finally, the regression function of (1) is derived as

$$\begin{aligned} y = f(\mathbf{x}) &= \sum_{i=1}^N (\alpha_i - \alpha_i^*) \boldsymbol{\phi}^T(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}) + b \\ &= \sum_{i=1}^N \beta_i K(\mathbf{x}, \mathbf{x}_i) + b \end{aligned} \quad (6)$$

where $K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\phi}^T(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x})$ is known as the kernel function and the coefficient β_i is expressed as the Lagrange multiplier α_i and α_i^* . In this study, the SVR model uses the following radial basis kernel function [8]:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2}\right) \quad (7)$$

Many of the coefficients β_i are nonzero values, and the training data points \mathbf{x}_i corresponding to the nonzero values, which are known as SVs, have an estimation error greater than or equal to the insensitivity zone.

B. Optimization of SVM model

It is important to use good data and input variables because the SVR model is a data-based model. Therefore, we are required to select the input variables and optimize the related parameters in the SVR model. In this study, GAs were used to select the input variables and optimize the parameters of the SVR model: the insensitivity zone ε , regularization parameter λ , and radial basis kernel function parameter σ .

GA is a search algorithm based on the mechanics of natural selection and natural genetics. In GA, the term chromosome typically refers to a candidate solution to a problem, generally encoded as a bit string. Because the GA is used to select the input variables and optimize the SVR parameters, the chromosome consists of a part of the parameter optimization and a part of the input variable selection. An allele in a bit string is either 0 or 1. The genotype of an individual in a GA using a bit string is simply the configuration of bits in that individual's chromosome. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosomes, successively replacing one such population with another. The GA requires a fitness function that assigns a score (fitness) to each chromosome in the current population. The fitness of a chromosome depends on how well that chromosome solves the problem at hand [11]–[13].

A fitness function to evaluate the appropriate level is proposed as follows:

$$F = \exp(-\mu_1 E_1 - \mu_2 E_2) \quad (8)$$

where μ_1 and μ_2 are weighting factors, and E_1 and E_2 are the root-mean-square (RMS) error and maximum absolute error, respectively. E_1 and E_2 can be described as follows:

$$E_1 = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2} \quad (9)$$

$$E_2 = \max_k \{|y_k - \hat{y}_k|\} \quad (10)$$

In (9), N is the number of data points, and y_k and \hat{y}_k are the target values and estimated values, respectively. The GA minimizes the weighted sum of the RMS error and the maximum absolute error.

III. ACCIDENT SCENARIOS

It was assumed that there were a variety of situations for SIS failure. Accident scenarios were proposed according to break sizes (270 beak sizes) relative to the double ended guillotine break (DEGB), and High and Low Pressure Safety Injection systems (HPSI, LPSI) actuation status in hot-leg LOCA and cold-leg LOCA. It was assumed that safety

injection tank (SIT) and Containment Spray System (CSS) were normally actuated [10].

The SIS including SIT, HPSI, and LPSI actuates automatically as soon as the safety injection actuation signal (SIAS) is generated based on low pressurizer pressure and high containment pressure. If the SIT that is a passive system actuates normally but the HPSI and LPSI systems do not actuate due to failure, the reactor core will be uncovered and then the RV will rupture.

Through the MAAP simulations, a total 540 data of severe accident scenarios were obtained. This data was composed of 270 pieces of hot-leg LOCA and 270 pieces of cold-leg LOCA. Simulation scenarios were assumed for the four cases. It was assumed for case 1 that the LPSI system was failed and the HPSI system was not operated at first but operated late in hot-leg break. For case 2, it was assumed that the LPSI system was failed and the HPSI system actuation was delayed in hot-leg break. For case 3 it was assumed that the LPSI system was failed and the HPSI system actuation was delayed in cold-leg break. Finally, for case 4 it was assumed that the LPSI system failed and the HPSI system actuation was delayed in cold-leg break. Simulations were conducted according to LOCA break size for each case (Table I). The purpose of this study was to predict the golden time for recovering the SIS to prevent the core uncover and RV failure when SIS actuation is delayed due to problems. The scenarios are similar to accident scenarios in a previous study [2].

IV. DETERMINING THE SIS GOLDEN TIME

A. Prediction performance of the SVM model

Table II summarizes the prediction performance results of the SVM model (HPSI delay). This table shows that the RMS errors for training data are approximately 12.1%, 0.57%, 18.3% and 0.35% for the two LOCA positions, and for the core uncover and RV failure, respectively. The RMS errors for the test data are approximately 10.86%, 1.02%, 21.37% and 0.56%.

Table III summarizes the prediction performance results of the SVM model (LPSI delay). This table shows that the RMS errors for training data are approximately 1.62%, 0.6%, 1.66% and 2.85% for the two LOCA positions, and for the core uncover and RV failure, respectively. The RMS errors for the test data are approximately 1.69%, 0.74%, 1.88% and 2.78%.

TABLE I. SIMULATION CASES

Case	Location	SIT Operation	CSS Operation	HPSI Operation	LPSI Operation
1	Hot-leg	Success	Inj & Rec	Delay Inj & Rec	N/A
2				N/A	Delay Inj & Rec
3	Cold-leg	Success	Inj & Rec	Delay Inj & Rec	N/A
4				N/A	Delay Inj & Rec

TABLE II. PREDICTION PERFORMANCE OF SVM MODEL (HPSI DELAY)

Break position	HPSI delay	Training Data		Test data	
		Maximum Error(%)	RMS Error(%)	Maximum Error(%)	RMS Error(%)
Hot-leg LOCA (Case1)	Core uncover	53.26	12.1	15.16	10.86
	RV failure	3.17	0.57	2.29	1.02
Cold-leg LOCA (case3)	Core uncover	84.88	18.3	34.74	21.37
	RV failure	1.95	0.35	1.48	0.56

TABLE III. PREDICTION PERFORMANCE OF SVM MODEL (LPSI DELAY)

Break position	LPSI delay	Training Data		Test data	
		Maximum Error(%)	RMS Error(%)	Maximum Error(%)	RMS Error(%)
Hot-leg LOCA (Case2)	Core uncover	9.5	1.62	2.39	1.69
	RV failure	1.68	0.6	1.13	0.74
Cold-leg LOCA (case4)	Core uncover	12.95	1.66	89.51	1.88
	RV failure	19.31	2.85	15.72	2.78

B. Results of SVM model

Figures 3-10 show the predicted golden time using the SVM model. And these show the comparison of the SVR model and MAAP data. Figures 3 and 4 show the HPSI golden time prediction of the case 1 for the severe accident scenario of hot-leg LOCA. Figures 5 and 6 show the LPSI golden time prediction of the case 2 for the severe accident scenario of hot-leg LOCA. Figures 7 and 8 show the HPSI golden time prediction of the case 3 for the severe accident scenario of cold-leg LOCA. Figures 9 and 10 show the LPSI golden time prediction of the case 4 for the severe accident scenario of cold-leg LOCA. The results show that using the SVM model, it is possible to accurately predict the golden time.

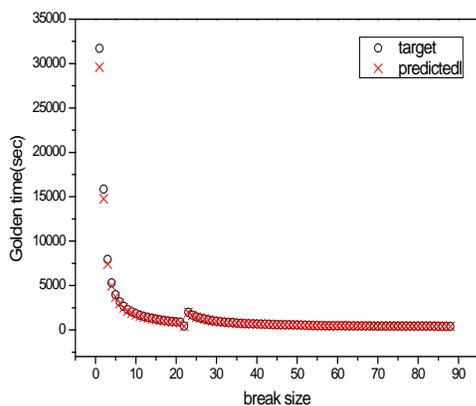


Figure 3. Golden time prediction of case 1 (HPSI delay - core uncover).

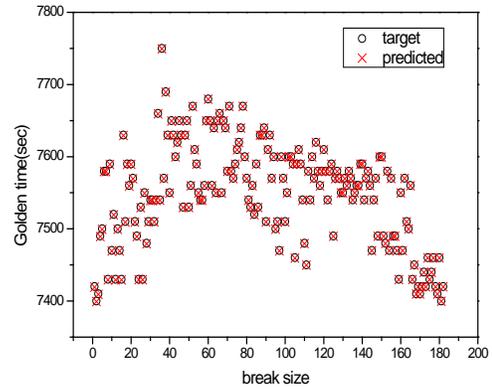


Figure 4. Golden time prediction of case 1 (HPSI delay - RV failure).

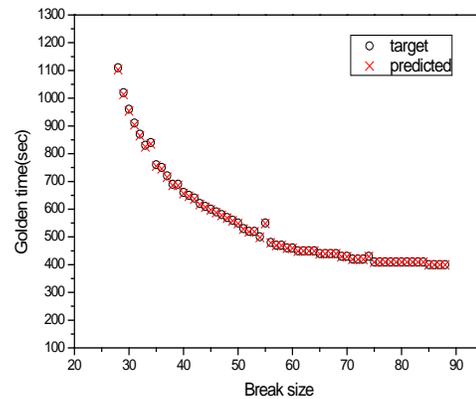


Figure 5. Golden time prediction of case 2 (LPSI delay - core uncover).

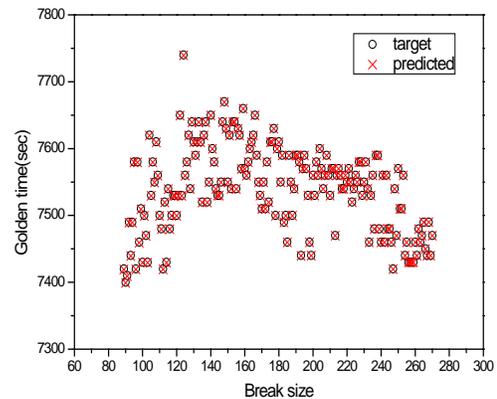


Figure 6. Golden time prediction of case 2 (LPSI delay - RV failure).

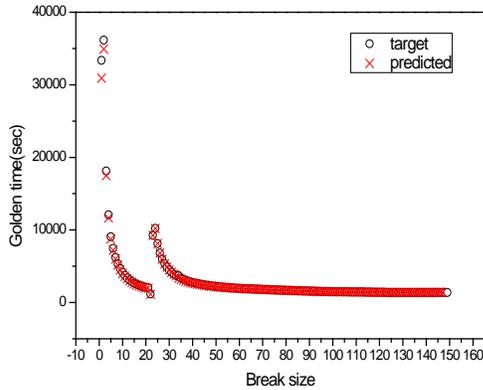


Figure 7. Golden time prediction of case 3 (HPSI delay - core uncover).

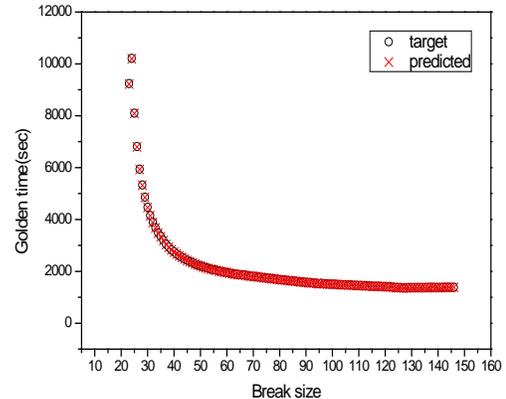


Figure 9. Golden time prediction of case 4 (HPSI delay - core uncover).

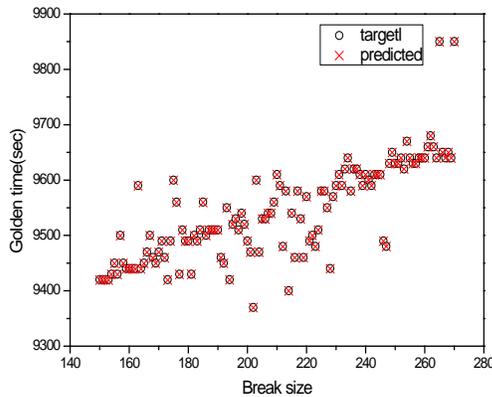


Figure 8. Golden time prediction of case 3 (HPSI delay - RV failure).

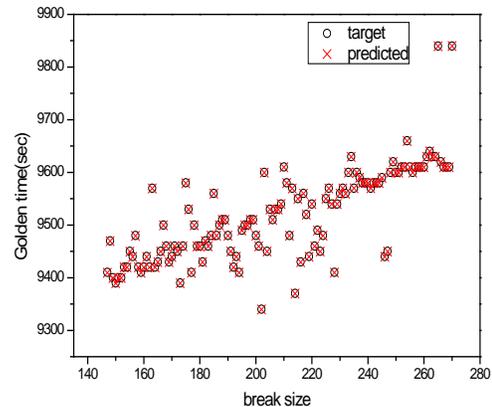


Figure 10. Golden time prediction of case 4 (LPSI delay - RV failure).

(MSIP), (Grant Nos. 2012M2B2B1055611, 2014M2A8A4044966).

V. CONCLUSION

In this study, the golden time according to LOCA break sizes was analyzed by using the MAAP code when SIS was not operating normally. In addition, the golden time prediction model was developed in LOCA circumstances by using SVM model. In summary, the results of this study suggest that the SVM model can accurately predict the golden time. If the golden time of SIS for accident recovery is predicted, the core will not be exposed through appropriate action. Also, the RV failure will be prevented by the cooling water injection even if the reactor core is exposed. These various golden time data are thought to be very useful to quickly deal with the actual accident. Also, it will be possible to more efficiently manage accidents beyond design basis for accident recovery.

ACKNOWLEDGMENT

This work was supported by a Nuclear Research & Development Program of the National Research Foundation of Korea (NRF) grant funded by the Korean government

REFERENCES

- [1] S. J. Han, H. G. Lim, and J. E. Yang, "Thermal Hydraulic Analysis Aggressive Secondary Cooldown in Small Break Loss of Coolant Accident with Total Loss of High Pressure Safety Injection," Proc. of KNS Autumn Mtg, Seoul, Korea, Oct. 5-11, 2003.
- [2] K. H. Yoo, D. Y. Kim, G. P. Choi, J. H. Back, and M. G. Na, "Prediction of Golden Time for Recovering the Safety Injection System in Severe LOCA Circumstances" Proc. of KNS spring Mtg, Jeju, Korea, May 6-8, 2015
- [3] D. Y. Kim, J. H. Kim, K. H. Yoo, and M. G. Na, "Prediction of hydrogen concentration in containment during severe accidents using fuzzy neural network," Nucl. Eng. Technol., vol. 47, no. 2, pp. 139-147, Mar. 2015.
- [4] S. H. Park, D. S. Kim, J. H. Kim, and M. G. Na, "Prediction of the reactor vessel water level using fuzzy neural networks in severe accident circumstances of NPPs," Nucl. Eng. Technol., vol. 46, no. 3, pp. 373-380, Jun. 2014.
- [5] S. H. Park, J. H. Kim, K. H. Yoo, and M. G. Na, "Smart sensing of the RPV water level in NPP severe accidents using a GMDH algorithm," IEEE Trans. Nucl. Sci., vol. 61, no. 2, pp. 931-938, Apr. 2014.

- [6] V. Kecman, Learning and Soft Computing. Cambridge, Massachusetts: MIT press, 2001.
- [7] V. Vapnik, The Nature of Statistical Learning Theory. New York: Springer, 1995.
- [8] M .G. Na, I. J. Hwang, and Y. J. Lee, "Inferential sensing and monitoring for feedwater flowrate in pressurized water reactor," IEEE Trans. Nucl. Sci., vol. 53, no. 4, pp. 2335-2342, Aug. 2006
- [9] D. P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods. New York: Academic Press, 1982.
- [10] D. S. Kim, S. H. Park, J. H. Kim , K. H. Yoo, and M. G. Na, "Determination of the Recovery Time of Unhealthy SISs in LOCA," Proc. of KNS spring Mtg, Gwangju, Korea, May 30-31, 2014.
- [11] M. G. Na, Y. R. Sim, K. H. Park, S. M. Lee, D. W. Jung, and S. H. Shin, "Sensor monitoring using a fuzzy neural network with an automatic structure constructor," IEEE Trans. Nucl. Sci., vol. 50, no. 2, pp. 241-250, Apr. 2003.
- [12] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Boston: Addison Wesley, 1989.
- [13] M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

Automatic Trigger Speed for Vehicle Activated Signs using Adaptive Neuro fuzzy system and Classification Regression Trees

Diala Jomaa, Siril Yella, Mark Dougherty
 Department of Computer Engineering, Dalarna University
 78170 Borlänge, Sweden
 Email: djo@du.se, sye@du.se, mdo@du.se

Abstract—Vehicle activated signs (VAS) are speed warning signs activated by radar when driver speed exceeds a pre-set threshold, i.e., the trigger speed. In order to be able to operate the sign more efficiently, it is proposed that the sign be appropriately triggered by taking into account the prevalent road and traffic conditions. This study presents the use of adaptive neuro-fuzzy inference systems (ANFIS) and classification and regression trees (CART) to predict the trigger speed of the VAS by using a historical speed data. The speed data is first explored and clustered by using a self-organizing map (SOM). Input vectors for simulation composed of time of day, traffic flow and standard deviation of mean vehicle speeds whereas the output vector consists only of vehicle speeds in the 85th percentile. The two models examined in this study were tested with historical speed data collected in Sweden during a period of one week and their performance was compared with Multi-layer perceptron (MLP). The results show that CART is reliable for predicting trigger speed for vehicle activated signs. However, compared to MLP and ANFIS, CART has superior performance than the other algorithms in terms of accuracy and complexity.

Keywords- *vehicle activated signs; adaptive neuro-fuzzy inference systems; classification and regression tree; self-organizing maps; trigger speed.*

I. INTRODUCTION

Vehicle activated signs (VAS) are road warning signs that measure the speed of passing vehicles and when a driver exceeds a particular threshold, display a warning message, e.g., ‘Slow down’ in combination with the current speed limit [1]. The threshold, which triggers the message to the driver, is commonly based on a vehicle’s speed, and accordingly, is called a trigger speed. At present, the trigger speed activating the VAS sign is usually set to a constant value that is relative to the static speed limit for the particular road segment. At the same time, static signs fail to provide the appropriate speed during dynamic traffic conditions [2]. To cope with the real time traffic management and time lags, a self-learning algorithm based on historical traffic speed data is proposed in this study. The proposed algorithm will be able to control the appropriate threshold, i.e., trigger speed for the VAS according to traffic situations. However, a large number of input factors, which impact the current traffic situations, need to be considered. These input factors include time/day, traffic flow, speeds, and type of vehicle. Several statistical approaches and artificial intelligence algorithms have been developed and implemented among road traffic management and control applications. Examples

of statistical methods are the auto-regressive integrated moving average (ARIMA) [3] and several non-parametric regression models. Several artificial intelligence approaches have been properly explored and developed. Among these neural networks (ANN) [4, 5], fuzzy logic [6], and further hybrid neuro-fuzzy intelligent systems [7] have been properly explored and developed. Given this background, the first objective of this study is to analyse the traffic speed data using a Self-organising map (SOM). SOM will further partition the data into separate clusters that have similar traffic patterns without the need of prior determination of the output. The second objective is the comparison of adaptive neuro-fuzzy inference system (ANFIS) with classification and regression trees (CART) for predicting the trigger speed for a VAS within each cluster obtained by the SOM.

The paper is organised as follows. Section II describes the automatic algorithm for triggering VAS. Section III presents the data collection and the experimental results obtained in this study and the paper is concluded in Section IV.

II. AUTOMATIC ALGORITHM FOR TRIGGERING VAS

A. Traffic pattern clustering

In the first step, SOM is initially used to visualize and explore the speed characteristics of the traffic data that has been collected. A SOM is further used to group traffic patterns into clusters that have similar speed characteristics. Based on the SOM algorithm described in the previous section, the SOM network is trained with 4 input factors based on the historical data: speed, time of the day, day of the week and type of vehicle. In this study, the type of the vehicles is mainly based on the length of vehicle detected by the radar. Speed characteristics for cars may be different than speed characteristics for trucks/trailers. Moreover, they may change or may repeat over time of the day and day of the week such as morning and evening hours or rush hours and non-rush hours during the weekday and weekend.

B. Trigger speed prediction

After exploring and grouping the traffic speed data into an appropriate number of clusters, a prediction algorithm is then developed for each cluster, which predicts the 85th percentile speed for each hour on the day. The prediction algorithm will be based on ANFIS and CART methods, which are powerful algorithms for traffic prediction based on a learning process. Time of the day, traffic flow and standard deviation are used as inputs traffic features whereas the 85th percentile is considered as the output of the two algorithms.

III. DATA COLLECTION AND EXPERIMENTAL RESULTS

All reported analyses were conducted with speed data collected at a roadway in Borlänge Sweden restricted with speed limit 40km/hr. A VAS was installed and was equipped with radar and a data logger to record the speed of passing vehicles 100m before the location of the VAS. The data comprised the vehicle speed, length of vehicle and date and time the vehicle passed the VAS.

In order to analyse and find similar traffic patterns, a SOM was further applied to the original speed data. Figure 1 shows a clear partitioning of the speed data respective to the length of vehicles and to the time of the day (night/day). Based on the clustering results, the speed data was grouped into major clusters, cars/vans (cluster 1) and trucks/trucks with trailers (cluster 2). Motorcycles are excluded from this study. Besides, the analysis showed that the day of the week has no effect on driver behaviour.

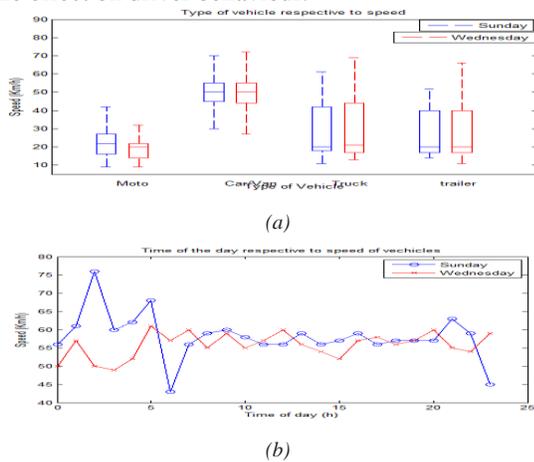


Figure 1. Analysis of speed of vehicles passing on Wednesday and Sunday respective to (a) type of the vehicles and (b) time of the day

Although, in the case of the trigger speed prediction, traffic flow, standard deviation and time of the day are utilized to alter the trigger speed of the VAS; various trigger speeds of the VAS come into service by assigning a different 85th percentile speed to each hour of the day. Following the practical rule of thumb, for each cluster dataset, 1/2 is used for training set and 1/2 used for testing set. Correspondingly, in order to illustrate the accuracy of the predicted trigger speed data set models the real output data set, the root mean squared error (RMSE) and the coefficient of determination (R^2) will be used as a performance index of the ANFIS and the CART. Better performance of the two algorithms requires a low value for RMSE and a high value of R^2 . Note that RMSE and R^2 are the average value of the two sets.

Table 1 summarized the results of the trigger speed prediction for each of the two clusters investigated in this study. The results obtained by the CART and the ANFIS are also compared to a multilayer perceptron (MLP). These results clearly demonstrate the superior predictive performance of the CART when compared to ANFIS and to the static MLP. Since, the RMSE for CART are lower to those of ANFIS within the two clusters but similar to the MLP within cluster 2. Furthermore, the learning duration of

the CART is much lower than the duration of MLP and ANFIS. This also implies that when using a huge data set, the performance of CART to predict the speed would be more useful to overcome faster the complexity of the problem.

TABLE I. PERFORMANCE FOR ANFIS , CART COMPARED TO THE PERFORMANCE OF MLP WITHIN CLUSTER1 AND CLUSTER2

Cluster 1- cars and vans			
	CART	MLP	ANFIS
R-squared	0.60	0.55	0.46
RMSE	0.15	0.19	0.18
Time(s)	0.13	22.54	0.27
Cluster 2- Trucks and trucks with trailer			
	CART	MLP	ANFIS
R-squared	0.59	0.60	0.46
RMSE	0.08	0.08	0.09
Time(s)	0.16	13.81	0.24

IV. CONCLUSION

The results from this study clearly demonstrate that first SOM can group the speed data into two major clusters, the first one is for all cars/vans and the second one trucks/trucks with trailers. Second, the results show that CART is reliable and could be used for predicting the trigger speed for vehicle activated signs in order to construct adaptive decision support systems. However, compared to MLP and CART are capable of predicting trigger speed with a high degree of performance. The performance is measured by a low value of RMSE and a high value of R^2 . In terms of computational complexity, CART is more efficient since regression trees use one pass and offer a fast learning approach when compared to the other learning approaches.

REFERENCES

- [1] L. K. Walter and J. Knowles, “ Effectiveness of Speed Indicator Devices on reducing vehicle speeds in London”, TRL report TRL 314, Transport Research Laboratory, 2008.
- [2] V. Sisiopiku, A. Sullivan, and G. Fadel, “Implementing active traffic management strategies in the U.S.”, Management and safety of transportation, UTCA, 08206, Birmingham, 2009.
- [3] M. M. Hamed, H. R. Al-Masaeid, and Z. M. Bani Said, “Short-term prediction of traffic volume in urban arterials”, Journal of Transportation Engineering, vol. 121 (3), 1995, pp. 249–254.
- [4] J. Park, D. Li, Y. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, “Real time vehicle speed prediction using a neural network traffic model”, The 2011 International Joint Conference on Neural Networks (IJCNN), San Jose, 2011, pp.2991–2996.
- [5] M. Dougherty, “A review of neural networks applied to transport”, Transportation Research Part C: Emerging Technologies, vol 3 (4), 1995, pp.247-260.
- [6] Y. Zhang, Z. Ye, “Short-term traffic flow forecasting using fuzzy logic system methods”, Journal of Intelligent Transportation Systems, 12 (3), 2008, pp. 102–112.
- [7] L. Dimitriou, T. Tsekeris, and A. Stathopoulos, ”Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow”, Transportation Research Part C: Emerging Technologies, vol 16 (5), 2008, pp. 554–573.

Process Chain Optimization using Universal State and Control Features

Melanie Senn

Fraunhofer IWM, Freiburg, Germany
Email: melanie.senn@iwm.fraunhofer.de

Ingo Schwab and Norbert Link

Karlsruhe University of Applied Sciences, Karlsruhe, Germany
Email: ingo.schwab@hs-karlsruhe.de,
norbert.link@hs-karlsruhe.de

Abstract—In order to obtain components with desired properties in production processes, it is necessary to consider all single process steps. The holistic view of a process chain enables the identification and the control of interactions between the single processes. In contrast to the description of a single process, state and control variables might not be consistent along the entire process chain. We propose a universal characterization of state and control features along the process chain that takes into account all relevant information and, at the same time, reduces the complexity in optimization. This allows us to optimize the entire process chain to obtain a desired product at the end under consideration of process noise, even for high dimensional state and control spaces.

Keywords—process chains; optimization; feature extraction.

I. INTRODUCTION

A production process describes the transformation of a component from its initial to its final state (e.g., stresses in the material) which depends on applied process controls (e.g., forces) and unknown process noise (e.g., different friction conditions due to lubrication). The holistic view of a process chain allows to identify and to control interactions between the individual processes to obtain a desired product at the end of the process chain (e.g., compensate geometric imperfections with distortion engineering [1]). The linking of the single processes can be realized by forward and backward information exchange [2]. A statistical learning approach [3] enables the control of a single process [2] using radial basis functions to identify controls. For each time step, past process noise is taken into account by measurements of process quantities. However, future uncertainties are neglected in that approach, but are considered in optimal control as realized for deep drawing including process noise [4].

If the characteristic process state is not accessible during processing, state observers can be used to extract the state information by observable quantities that are measurable during process execution. State observers can be established by statistical learning approaches such as Artificial Neural Networks [5] and Symbolic Regression [6].

While we can use the same state and control variables within one process, the description among different processes is not necessarily consistent. There are properties that are meaningful for a specific process, but would be pointless for the entire process chain (e.g., cup height is a component property in deep drawing which is not applicable to rolling or heat treatment). We can find a common description along the entire process chain by expert knowledge (e.g., stresses in sheet metal forming) and / or use a universal description for state and control variables as proposed in this paper.

We introduce the concept of process chain optimization in Section II and give an example process chain in Section III.

II. PROCESS CHAIN OPTIMIZATION

We propose a universal description for state and control variables from which the characteristic features are automatically extracted for optimization and retransformed for application to the process.

A. Process chain modeling

Each process in the chain is characterized by its state during processing. A single process p_t transforms an initial state \mathbf{x}_t to a final state \mathbf{x}_{t+1} depending on the control variables \mathbf{u}_t and the unknown process noise \mathbf{v}_t as depicted in Figure 1. Each transformation is associated with local costs C_t (e.g., the production effort). The final costs C_{t+1} are added at the end of the process chain (e.g., the deviation of the actual state from a desired state). The Bellman equation [7] describes the optimal control problem for each process. The total costs J_t to be minimized comprise the local costs C_t and the successor costs J_{t+1} . The successor costs consist of the local costs of the remaining processes in the chain plus the final costs at the end. The Bellman equation can be solved by (Approximate) Dynamic Programming [7] (e.g., modeling successor costs and state transitions by nonlinear regression with Artificial Neural Networks from simulation data [4]).

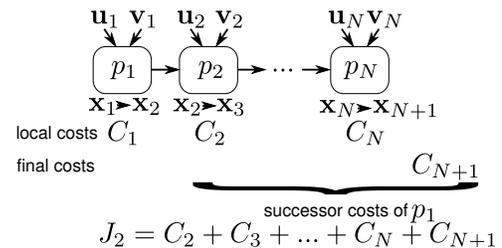


Figure 1. Process chain modeling.

The process data $\mathbf{u}_t, \mathbf{x}_t$ can be obtained by real experiments including uncertainty or by deterministic simulations without random noise. The local and final costs are defined by a human process expert.

B. Feature extraction and unwrapping

In order to obtain a characteristic description of the process chain, all relevant quantities are collected to represent

- 1) universal state variables $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{x}_{N+1}$, (e.g., thickness and depth [8])
- 2) universal control variables $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$.

Figure 2 points out how feature extracting and unwrapping is interconnected with optimization. We can either apply Principal Component Analysis (PCA) to state and control variables separately or use Partial Least Squares Regression (PLS) on

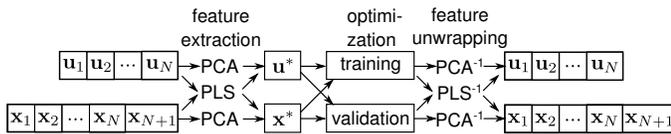


Figure 2. Feature extraction and unwrapping.

both simultaneously. PCA [3] allows dimension reduction from a higher dimensional to a lower dimensional space retaining most of the information (e.g., variance) in the data. PLS [3] enables a dimension reduction in input and output during regression. The features of the universal state and control variables are extracted: the control features \mathbf{u}^* and the state features \mathbf{x}^* . The optimization problem (Bellman equation) is formulated and solved in the feature space. The training step establishes the models for state transitions and successor costs, whereas the validation step tests the established models with previously unseen data. Then the features are unwrapped (inverse PCA / PLS). The unwrapped features reveal the controls and states in its original space for process application and are interpretable.

This procedure allows to reduce the dimensionality in state and control spaces to handle the complexity in optimization (that grows exponentially with increasing spaces). Constant dimensions in state space (e.g., zero cup height in rolling) or control space indicate that these dimensions are not affected by any changes in the current process. They will be removed by dimension reduction.

C. From process data to chain optimization

A real life example for sheet metal forming is to obtain a homogeneous sheet thickness distribution at the end (final costs) under consideration of low production effort in each step (local costs). To realize the proposed concept, we recommend to implement the following procedure:

- record process data $\mathbf{u}_t, \mathbf{x}_t$ from simulations or experiments and optionally induce artificial process noise \mathbf{v}_t if not contained in process data,
- define cost functions (local costs C_t and final costs C_{N+1}) based on \mathbf{u} and \mathbf{x} ,
- extract control features \mathbf{u}^* and state features \mathbf{x}^* along process chain from process data $\mathbf{u}_t, \mathbf{x}_t$,
- create cost mapping functions based on extracted features $C_t(\mathbf{u}^*, \mathbf{x}^*)$ and $C_{N+1}(\mathbf{u}^*, \mathbf{x}^*)$,
- build and validate process models for each step in chain by regression for (1) state transition, (2) costs, and (3) Bellman equation (after its solution),
- unwrap control features \mathbf{u} and state features \mathbf{x} along process chain from control features \mathbf{u}^* and state features \mathbf{x}^* ,
- evaluate process chain optimization.

III. EXAMPLE PROCESS CHAIN

An example process chain in sheet metal forming is given in Figure 3 [9]. It contains the processes rolling (forming a metal sheet from a metal block), annealing (heat treatment) and deep drawing (forming a cup-shaped workpiece from a metal

sheet). The optimization objective is to compensate direction-dependent deformation which results in unwanted earing of the resulting cup. The holistic view of the process chain allows to

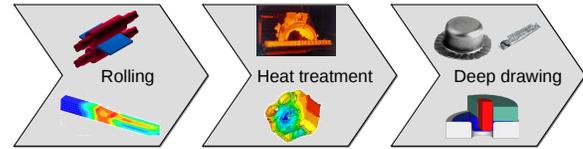


Figure 3. Example process chain "sheet metal forming" [9].

understand and to control the deformation behavior of the sheet metal components. The evolving microstructure (e.g., grain size and orientation of the crystalline structure of metals [10]) as the characteristic state of the metal sheet is controlled along forming, heat treatment and deep drawing operations to finally produce minimal earing. The proposed approach enables the handling of the complex microstructure.

IV. CONCLUSION

We introduced a universal characterization for state and control features in production process chains. This allows a consistent description along the entire process chain and, at the same time, a reduction of complexity in state and control spaces to realize an efficient process chain optimization. Future work deals with application of the proposed concept to different process chains. This allows us to compare it with conventional ways of process chain optimization.

ACKNOWLEDGMENT

We gratefully acknowledge the support by the DFG Research Training Group 1483 "Process chains in manufacturing" and the Fraunhofer TALENTA grant.

REFERENCES

- [1] H.-W. Zoch, "From single production step to entire process chain - the global approach of distortion engineering," *Materialwissenschaft und Werkstofftechnik*, vol. 37, no. 1, 2006, pp. 6–10.
- [2] C. Ament and G. Goch, "A process oriented approach to automated quality control," *CIRP Annals - Manufacturing Technology*, vol. 50, no. 1, 2001, pp. 251–254.
- [3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [4] M. Senn, N. Link, J. Pollak, and J. H. Lee, "Reducing the computational effort of optimal process controllers for continuous state spaces by using incremental learning and post-decision state formulations," *Journal of Process Control*, vol. 24, no. 3, 2014, pp. 133–143.
- [5] M. Senn, K. Jöchen, T. Phan Van, T. Böhlke, and N. Link, "In-depth online monitoring of the sheet metal process state derived from multi-scale simulations," *International Journal of Advanced Manufacturing Technology*, vol. 65, no. 5-8, 2013, pp. 1–12.
- [6] I. Schwab, M. Senn, and N. Link, "Improving expert knowledge in dynamic process monitoring by symbolic regression," in *Sixth International Conference on Evolutionary Computation*, 2012.
- [7] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed. Wiley-Interscience, 2011.
- [8] M. Senn, J. Schäfer, J. Pollak, and N. Link, "A system-oriented approach for the optimal control of process chains under stochastic influences," in *AIP Conference Proceedings*, vol. 1389, 2011, pp. 419–422.
- [9] Research training group 1483. Last accessed: 07/24/2015. [Online]. Available: <http://grako-1483.wbk.kit.edu>
- [10] D. Helm, A. Butz, D. Raabe, and P. Gumbsch, "Microstructure-based description of the deformation of metals: theory and application," *JOM Journal of the The Minerals, Metals & Materials Society (TMS)*, vol. 63, no. 4, 2011, pp. 26–33.

Application of Task-to-Method Transform to Laser Seam Welding

Jürgen Pollak

Institut für Angewandte Forschung
Hochschule Karlsruhe - Technik und Wirtschaft
Karlsruhe, Germany

e-mail: juergen.pollak@hs-karlsruhe.de

Abstract—Intelligent machines are supposed to automatically set process parameters when faced with a task to be processed. The intelligence is often realized by databases which link the task with process parameters. This paper reviews a flexible and portable (to various processes) system to find (optimized) process parameters which force the process outcome to pre-defined quality under given variable conditions. In addition, extensions to the original system are presented and the whole concept is applied to laser seam welding (LSW). Experimental results based on real process executions demonstrate the applicability in industrial environments.

Keywords—Machine Intelligence; Task-to-Method-Mapping; Support Vector Regression; Data Domain Description; Laser Welding

I. INTRODUCTION

Intelligent production machines have to flexibly respond to varying tasks by setting their process parameters in such a way that given task goals are reached under given conditions. For this purpose, the machine needs to represent and use knowledge about the relation between process parameters and process goals under given (but varying) process conditions.

This paper reviews a general concept and an implementation of the automatic extraction and application of such process knowledge represented in experimental outcome data. [1] A process goal is represented by quality measure values to be achieved by the process. In experiments, a process is executed with defined process parameters and under known or controlled conditions. The achieved quality is measured after the experimental process execution. The data (process parameter, condition and quality values) may be sampled from real physical, or simulated numerical experiments. The data form the basis for the estimation of a so-called goal function with the process parameters and the condition quantities as independent variables. The goal function defines quantities which describe the desired end state of the process and represents the process knowledge. Once derived from the data, it is used to find the process parameters which yield a desired result. Finding the appropriate process parameter settings (“process methods”) yielding a given goal is then equivalent to finding those parameter values, where the goal function takes on the desired, given goal values. More than one solution exists, the set of all solutions is called “level set”. To select the best suitable method, it is further proposed to use the level set as a basis to optimise a given cost function which associates cost with the process parameters.

The goal function is constructed by applying non-linear kernel regression to the experimental data. Experiments frequently also deliver process boundaries, beyond which the process will not execute or no result is reached at all. This feasibility boundary is modelled in our approach via a two-class support vector machine. Furthermore, it has to be taken

into account that the goal function can only be applied in areas which are supported by experimental data. This so-called confidence domain is modelled by hulls enclosing the experimental data.

The paper is organized as follows: Section II gives a review of the originally developed Task-to-Method Transformation (T2MT). This section is divided into subsections giving an overview over the general concept, followed by details about the process modelling by goal functions and a classifier to constrain the predictions to regions supported by data. The last subsection of Section II presents the procedure how to find process parameters from given tasks and process models. The next part, in Section III, applies the methods to LSW and describes various extensions to the original system. A short introduction to LSW is given in Section III-A. Section III-B extends the process model by multi-valued goals. Acceptable goal ranges are introduced in Section III-C. The calibration of models to new situations (Section III-D) and using a process model to apply small parameter adjustments (Section III-E) are further enhancements. All these extensions convert the original T2MT into an industrial applicable system. An experimental verification using real process data is given in Section III-F.

II. SUMMARY OF T2MT

This chapter gives a summary of the T2MT, which was introduced in [1]. It was developed as a general applicable system to find process parameters from process models derived from experimental data. T2MT was originally verified by numerical simulations for the process of resistance spot welding.

A. General Concept

Process parameters describe the variable control quantities, which can be set by the process machine in a vector \vec{p} . The process conditions represent all fixed quantities in a vector \vec{c} , which otherwise govern the process and cannot be set by the machine. They are fixed externally and independently from process execution. The goals are quantities characterising the desired end state of the process in a vector \vec{g} . For example, in car seat manufacturing metal sheets are joined by welding seams. The process parameters are in the simplest case laser power, laser focus point and welding speed. Possibly varying conditions are the materials and thicknesses of the two sheets. The goal is the double valued extend of the welding seam, seam width and seam depth, which have to be obtained.

The task is then given by the combination of the goals and the conditions $\vec{t} = [\vec{g}, \vec{c}]$. An intelligent machine has to find at least one method (consisting of process parameters \vec{p}) fulfilling a given task \vec{t} , or it has to state that the task is not feasible. In other words, the machine has to perform a mapping from \vec{t} to \vec{p} . We call this the T2MT.

In experimental process investigations, a variety of process conditions is explored. For each specific condition, a set of methods \vec{p} is applied and the resulting goal values \vec{g} are measured. Each single experiment gives a vector triple $[\vec{g}, \vec{p}, \vec{c}]$ and the available experimental series give a set of such triples. We propose to build an abstraction of the experimental data by the formation of a goal function $\vec{g}(\vec{p}, \vec{c})$. It represents the knowledge contained in the experimental data.

Furthermore, the goal function should only be applied in areas supported by experimental data. This so-called confidence domain is modelled by hulls enclosing the experimental data. A support function $s(\vec{p}, \vec{c}) > 1$ can be defined inside the hulls, $s(\vec{p}, \vec{c}) = 1$ on the hulls and dropping continuously to $s(\vec{p}, \vec{c}) = 0$ within some distance outside. This support function defines some space around the experimental data, which may be accepted as a region for inter- and extrapolations of new (yet unseen) tasks \vec{t} and methods \vec{p} .

The goal function $\vec{g}(\vec{p}, \vec{c})$ and the support function $s(\vec{p}, \vec{c})$ finally form the process knowledge model, extracted from the experimental data.

The goal function is then used to perform the T2MT. The condition vector $\vec{c}_{\vec{t}}$ is a constant, when a specific task \vec{t} is given. In this case, the goal function is only a function over the corresponding subspace of \vec{p} . The level set of parameter vectors \vec{p} defined by $\vec{g}(\vec{p}, \vec{c}) = \vec{g}_{\vec{t}}$ represents the set of methods fulfilling the task. Finding the level set of the goal function is thus the core component of the T2MT. Afterwards, the other model function $s(\vec{p}, \vec{c})$ is applied to the level set to exclude unsupported method solutions.

The resulting restricted solution set forms the search space for the minimisation of a cost function. Based on external knowledge, the cost function assigns cost to the process parameters and process goals.

The solution for the vector-valued goal function $\vec{g}(\vec{p}, \vec{c}) = \vec{g}_{\vec{t}}$ with $\vec{g}, \vec{g}_{\vec{t}} \in \mathbb{R}^M$ can be broken down into the solution for M single-valued goal functions $g_i(\vec{p}, \vec{c}) = g_{\vec{t}, i}$, $i = 1, \dots, M$. Each of them has a level set $\{\vec{p}\}_i$ as a solution. The level set satisfying all equations is given by the intersection of all single sets $\{\vec{p}\}_1 \cap \{\vec{p}\}_2 \cap \dots \cap \{\vec{p}\}_M$. It is therefore sufficient to construct a method for single-valued goal functions.

B. Goal Function Approximation

A central part, when looking for a method \vec{p} solving a given task $\vec{t} = [\vec{g}, \vec{c}]$ under constraints \vec{c} , is a model description of the physical process. The process is modelled by construction of a goal function $\vec{g}(\vec{p}, \vec{c})$ which comprises the whole necessary knowledge about the process. In any experiment, the conditions \vec{c} and the method parameters \vec{p} are set. The process is then executed and the outcome is measured. The outcome quantities are identical with the goal describing quantities \vec{g} , which describe the desired final properties of the process result. Experiments are conducted under many different \vec{c}_i and \vec{p}_i and corresponding \vec{g}_i are measured. This gives an experimental sample of triples $\{\vec{p}_i, \vec{c}_i, \vec{g}_i\}_{i=1}^N$, which is used to create an abstraction in the form of a goal function. Subsequently, this goal function can be inverted to find appropriate parameters for given goals and conditions. For a given task \vec{t} the conditions \vec{c} are fixed and the goal function $\vec{g}(\vec{p})$ depends only on process parameters \vec{p} . In most cases there is no explicit prior model available to form this function. Therefore, the goal function has to be extracted from experimental (real or simulated) data.

To represent the goal function, basically any regression

method can be used. All methods build up the regression function by a weighted superposition of base functions, which itself may need parametrisation. A fitting algorithm is applied to determine the weights and parameters of the base functions so that the superposition approximates the observed data as accurately as possible. For most methods, the number of base functions must be specified in advance (and by association, the complexity of the representable function).

Real production processes may show very complicated non-linear dependencies on process parameters. But piecewise, in the small surrounding of an assumed operating point, the process model function behaves quite smooth. Support Vector Regression (SVR) is an universal method to find the smoothest regression function representing observed data. [2] The regression function is build up by a superposition of more or less localized non-linear functions (depending on kernel choice) pinned at (measured or simulated) data vectors. It is beneficial that SVR picks out only the relevant subset of the whole data set to describe the smooth goal function. These data vectors, which determine the function to represent this experimental knowledge in a generalized way by the goal function $\vec{g}(\vec{p}, \vec{c})$, are called Support Vectors.

The goal function by SVR representation takes the form

$$g(\vec{x}, \vec{\alpha}) = \sum_{i=1}^l \alpha_i K(\vec{x}, \vec{x}_i) - \rho, \quad \vec{\alpha} = (\alpha_1, \dots, \alpha_l), \quad (1)$$

where the parameters α_i and ρ are determined by an quadratic optimization algorithm from the data. [2]

Two kernels, used in calculation for the present paper, were

$$\text{Polynomial } K(\vec{p}, \vec{q}) = (\gamma \langle \vec{p}, \vec{q} \rangle + c)^d \quad \text{and} \quad (2)$$

$$\text{Gaussian RBF } K(\vec{p}, \vec{q}) = \exp(-\gamma \|\vec{p} - \vec{q}\|^2). \quad (3)$$

The free parameters (γ, d) in these kernels are found and fixed by exploring the (γ, d) -space for values of minimum residual fitting error by cross validation. [3]

When doing the numerics, especially with polynomial kernels, one will face numerical issues when using the raw values of the process quantities. They are more significant if some quantities have very small values, others very high values, and if the dynamic ranges are very different. These issues can be circumvented by normalizing the training data to range $[0, 1]$ and de-normalizing the results accordingly.

Processes usually have sharp boundaries in the space of parameters and conditions, beyond which the process collapses or exhibits unacceptable behavior. This region of unfeasible processes could be represented by a special goal value. But this would result in a discontinuity of the goal function (1) and consequently in fitting problems. We propose to represent the feasibility region within the boundary by a separate step function, changing value at the boundary. The feasibility region is then represented by a two-class (feasible / unfeasible) classifier. The model of the feasibility region is formed by the training of a Support Vector Machine [2], which requires training data covering both classes.

C. Data Support Region

Experimental data will usually explore some finite areas in the (\vec{p}, \vec{c}) -space, while the goal function covers the whole space. In order to get a reliable functional approximation of the goal function, it is necessary to restrict the goal function to areas supported by experimental data.

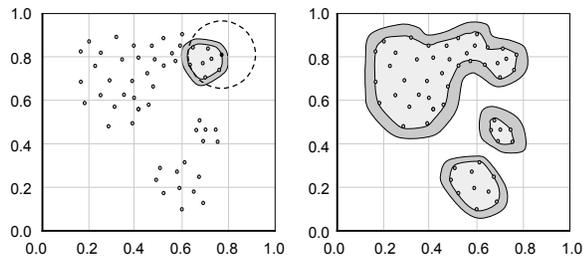


Figure 1. Local (a) and global (b) SVRM boundaries.

The region in input space, defined by the hull enclosing the training data $\{\vec{p}_i, \vec{c}_i, \vec{g}_i\}_{i=1}^N$, will be described by the support function $s(\vec{p}, \vec{c})$. The generalization (interpolation and extrapolation) of the experimental data by the goal function is only valid inside this hull and in a small region around it. We call this whole trusted region the data supported region.

In T2MT, this region is modelled by a one-class support vector machine [4] with parameters calculated from the training data set, inspired by the work [5].

According to [4], the problem to describe the supported region may be solved by a mapping of the training data to a feature space where they are linearly separated from the origin. The separating hyperplane with maximum distance from the origin determines the boundary.

The region of known data (corresponding to the set of known tasks) can be formulated as [1]

$$\frac{1}{\rho} \sum_i^{S_N} c_i K(\vec{x}, \vec{x}_i) \geq 1, \quad (4)$$

where

$$K(\vec{x}, \vec{y}) = \exp(-\gamma |\vec{x} - \vec{y}|^2) \quad (5)$$

and

$$\rho = \sum_j^{S_N} \alpha_j K(\vec{x}_i, \vec{x}_j), \quad (6)$$

with S_N = number of support vectors.

In order to accept a new task some distance away from the already known task (which means to accept the value of our goal function in an extrapolated region), we soften the boundary definition:

$$\frac{1}{\rho} \sum_i^{S_N} c_i K(\vec{x}, \vec{x}_i) \geq p \quad (7)$$

$$p \approx 0.9 \quad (0 < p \leq 1) \quad (8)$$

The setting of parameter p determines the range of how far extrapolations of the goal function are accepted. The value of p depends on the sample density of the training data and on peculiarities of the process under consideration. It has to be found by experiments. In Figure 1, the boundary of the light grey region is defined by $p = 1$. In this area data is available and the goal function interpolates in between. In the dark grey region, defined by $p < 1$, the goal function extrapolates to regions not supported by training data. In this region, the results of the goal function are assumed to be uncertain but the result shall be accepted. Outside the dark region results of the goal function are rejected as untrustable.

A still open question is how to choose the width γ of the Gaussian Kernel function (5). If these functions are highly

localized, the boundary around the training data will be very sharp with poor generalization performance and a large number of support vectors. If the kernel functions on the other hand are too broad, the resulting boundary may be too smooth (only a small part of the training vectors are considered as support vectors) and occlude essential structures. It is therefore crucial to find a value γ producing a good description of the training data.

Cross-validation cannot be used, because there is no false-class in our training set. We follow the procedure presented in [5], which is summarized shortly.

In a first step all training vectors which might lie on the data domain boundary are identified (Figure 1a) by local one-class Support Vector Machine Classifiers (in [5], [6] this is called SVRM - Support Vector Representation Machine). Every training vector is considered with its surrounding vectors inside a sphere of a given radius. It can be assumed that this restricted sub-sample follows a simple distribution, therefore it is justified to choose a local $\gamma_i = d_i$. If the training vector \vec{x}_i lies on the boundary (or very near the boundary) of the sphere, this vector is stored in a list of 'local' boundary vector candidates of the total sample.

The second step is to train several global SVRMs using all training data, each with different γ . For every cycle, the training vectors lying on or very close to the boundary of the global SVRM are selected and stored in lists of 'global' boundary vector candidates.

In a last step (Figure 1b) the global γ is chosen, for which the best match is found between the set of 'global' boundary vector candidates and the set of potential (local) boundary vector candidates. Further details about the algorithm can be found in [1].

D. Parameter Extraction (Level Set)

The goal function $\vec{g}(\vec{p}, \vec{c})$ (1), defined in Section II-B, represents a surface embedded into a high-dimensional space spanned by the process parameters \vec{p} and process conditions \vec{c} . A specific task is then given by the demand to reach a task goal $\vec{g}_T = \vec{g}(\vec{p}, \vec{c}_T)$. This can be viewed as the \vec{p} -dependent intersection of a hyperplane $\vec{g}_T = \text{const}$ with the curved surface $\vec{g}(\vec{p}, \vec{c}_T)$. The level set is the set of solutions

$$\text{level set: } \{ \vec{p} \mid \vec{g}_T = \vec{g}(\vec{p}, \vec{c}_T) \}. \quad (9)$$

In the present case, the level set can be found by meshing the high-dimensional feature space. The mesh is refined by incrementally subdividing cells, which are intersected by \vec{g}_T , until the desired accuracy is reached. The level set is afterwards given by a discrete set of solutions. It may contain solutions outside the region supported by data (Section II-C). Therefore, the level set has to be confined to this region to form the final set of feasible methods.

The final level set is then a list of process parameter vectors. Each of them will produce the result \vec{g}_T as requested by the task:

$$\{ \vec{p}_k \mid \vec{g}_T = \vec{g}(\vec{p}_k, \vec{c}_T) \}, \quad k = 1, 2, \dots \quad (10)$$

In the special case of a quadratic polynomial kernel

$$K(\vec{p}, \vec{q}) = (\gamma \langle \vec{p}, \vec{q} \rangle + c)^2, \quad (11)$$

the solution can be found analytically by direct calculation

$$g_t = g(\vec{x}, \vec{\alpha}), \quad (12)$$

$$x_k = f(g_t, \vec{\alpha}, (x_1, x_2, x_{k-1}, x_{k+1}, \dots, x_n)). \quad (13)$$

Every solution in the found level set is associated with some cost such as energy, wear of tools, production cycle time and so on. To select the most efficient process method from the level set, one should be able to define a cost function depending on process parameters \vec{p} and process conditions \vec{c} .

If the resulting level set is given as a discrete set of only a few hundred or thousand points, and if the computational effort to calculate the cost function is low, it is sufficient to do a complete search.

III. APPLICATION TO LSW

The main goal of this work is to demonstrate the applicability of T2MT to real industrial processes. LSW was chosen as a sample process because the setup of a specific laser welding machine or welding task is a very time consuming procedure.

A software library incorporating all the methods of Section II was developed. To fulfil additional functional requirements which emerged during application and testing, extensions to this original system were developed and implemented. In the following part of the paper, these extensions and the results of the verification procedure are presented.

A. Introduction to LSW

In order to weld work pieces by laser, the work pieces have to be held in fixed positions. For that purpose, a laser welding cell is equipped with complex jigs composed of many pneumatic cylinders, limit switches, proximity switches, mechanical stops and grippers. Such jigs are usually mounted on a turn table which moves the fixed work pieces into a completely enclosed welding cabin. Inside this cabin, one or more robots are equipped with laser welding heads. These are optical devices with fixed or adjustable focal length. An optical fibre guides the laser light from the laser device to the welding head.

To make a seam, the laser light has to be focused on the work piece. The focus point has to be moved along the target line, it can be exactly on the surface of the work piece, some millimetres above, or inside or below the work piece. The corresponding parameter, called defocus, can be used to control the ratio between welding seam width and the penetration depth. Laser power is in the range of up to 6000 W. One of the most important advantages of LSW is the distance of the welding head to the work piece. In the presented examples, this distance (approximately equal to the focal length) is about 60 cm. Another benefit is the huge processing speed, the welding progress can be more than 200 mm/s.

The result of the welding process can be described by weld width and penetration depth (Figure 2). The customer usually wants to specify these values. Additionally, more quality constraints must be satisfied: Undercut, root cavity, excess penetration, excess weld metal (Figure 3).

B. Two Goal Functions

Penetration depth and weld width are two quantities, which define the goal values to be fulfilled by the process. Each quantity is modelled separately by SVR. In order to find the appropriate process parameters, one needs to search for the overlap of the two level sets for each goal value. One method to achieve this, is (1) to determine the level set from only one goal model and then (2) restrict this level set by the evaluation of the second goal model and force it to be equal to the second goal value.

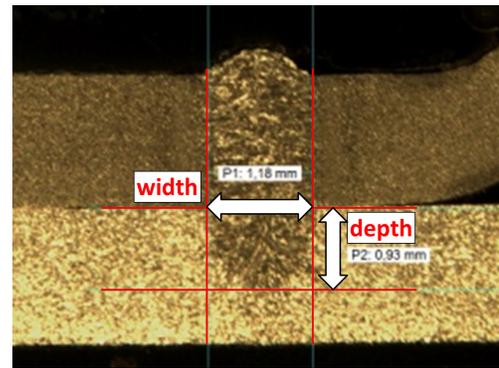


Figure 2. Micrograph showing the main quality quantities in LSW: penetration depth and weld width (with permission from AWL [7]).

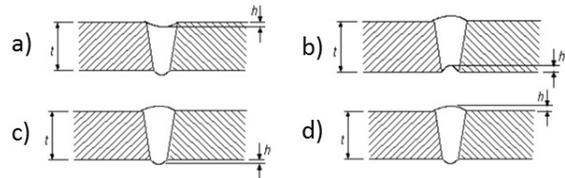


Figure 3. Additional quality measures: a) Undercut, b) Root concavity, c) Excess penetration and d) Excess weld metal.

If the level set of the first goal value is determined by the analytical method (11)-(13), the resulting process parameters will reproduce exactly the first goal value. But there are situations in which no single parameter set out of this level set will produce the second goal with acceptable accuracy. The user has to specify the allowed deviations in the goal values and the feasible resolution of the process parameters. Additionally, to accept small deviations it showed to be advantageous to run the determination of the level set twice, with changed roles of the first and second goal quantities.

Determination of the level set by the subdivision algorithm (Section II-D) does not suffer from this issue, because it is internally already working in a discretized parameter and goal space.

C. Parameter Extraction Allowing Goal Ranges

In LSW, it is not always appropriate to match both goal values of Penetration Depth and Weld Width exactly. A customer may require the penetration depth to be equal to the thickness of the lower sheet. Additionally, he may only set the requirement on the weld width to be greater than a given minimum value or to lie in a given range. For this, the goal range is discretized according to the specified resolution of the goal value and a levelset is determined for each of these discrete levels. After that, the union of all found levelsets is build and repetitions of parameters are deleted.

D. Model Mapping and Model Calibration

A laser welding cell is typically build up at the vendors facility, where also process parameters for good quality products are determined. Test sheets are welded with different process parameter settings, cross sections are cut and polished. Penetration depth, weld width and other parameters, which characterise the welding seam, are measured by micrography (Figure 2 and Figure 3). First products are produced, and if the customer is satisfied with the quality, the welding cell will be dismantled and rebuilt at the customers factory.

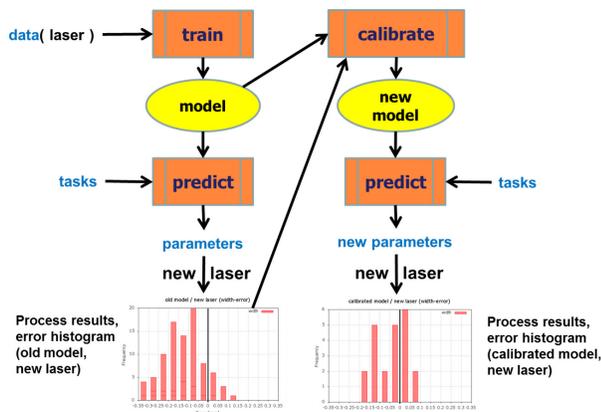


Figure 4. Model mapping or model calibration.

When the first products are produced at the customer, it is not unusual that the results are slightly different to the previously results at the vendor. Something small has changed in the whole setup, which has an influence on the results but the cause might not be obvious. Because the physics of the process has not changed, it is a good assumption that a process model developed by the vendor is still correct. Then only a small affine transformation of the combined process parameter and goal quantity space will shift the process model, so that it now captures the new situation with sufficient accuracy. The new measurements at the customer are used to calibrate the process model (Figure 4).

A new task may require to weld a material combination which is similar to another material combination with an already existing process model. It can be assumed that the physics does not behave very different and therefore, the existing model can be used as a basis for the new welding task. Because the qualitative behaviour is already modelled, only a small number of additional experiments have to be done to capture deviations. The original model can then be transformed into a new model for the new task by the same procedures as in the case of calibration. In this way, an existing process model is mapped to a new process model for a new task.

Both procedures, model calibration and model mapping, are algorithmic equivalents.

E. Parameter Adjustment

The previous chapter dealt with the calibration of a process model to slightly different boundary conditions. The idea was to create an adjusted process model, which again is capable to describe the whole process space.

But sometimes it is enough to just find better process parameters for a given task. Again, under the assumption of similar physical behaviour, an existing process model can be used to calculate gradients in parameter space which yield better goal values. The process model does not have to be very precise in an absolute sense, but it should exhibit the same qualitative behaviour.

F. Experimental Verification

In the I-RAMP³ project (see Section ACKNOWLEDGEMENT), the changed conditions (after dismantling and rebuilding a laser welding cell) were simulated by the exchange of the laser source and the optical fibre connecting the source with the laser head mounted on the robot. Theoretically, there

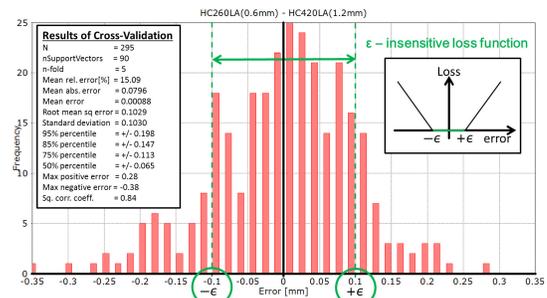


Figure 5. 5-fold cross validation error histogram of process model.

should be no change in the process results, if the laser source and the fibre are exact replacements. But it is nearly sure that these components are a bit different, e.g., the source is build by an other vendor and produces slightly different laser output power.

In the following, the steps performed to demonstrate the usability and applicability of T2MT to LSW are described in details, please refer to Figure 4. All experiments were done by AWL. [7]

1) *Experimental Sampling of Process Space*: Experiments were done on three different material and thickness combinations:

- HC260LA-0.6mm on HC420LA-1.2mm,
- HC420LA-1.2mm on HC380LA-1.5mm, and
- DC04-1.5mm on HC380LA-1.5mm.

The process parameter space was sampled on a regular grid in following ranges:

- laser power: 3500 W to 5500 W, in steps of 500 W,
- focus: -20 mm to +20 mm, in steps of 5 mm and
- weld velocity: 80 mm/s to 220 mm/s, in steps of 10 mm/s.

In the case of DC04 and low laser power, the lower limit of weld velocity was reduced to 30 mm/s.

All in all 1485 experiments were made. Each welding seam was cut, sanded and measured by micrography. The measured quantities were penetration depth, weld width, undercut, root concavity, excessive penetration and excess weld metal (Figure 2 and Figure 3).

2) *Generation of Process Models*: Based on these data, process models for penetration depth and weld width were calculated using SVR (Sections II-B and III-B). The boundary of the space supported by the data was modelled by SVRM (Section II-C). All calculations regarding support vectors are based on the library libsvm [8].

Only data which produced good quality was used to build up the process models. The conditions to specify good quality were set to penetration depth ≥ 0.1 mm, weld width ≥ 0.5 mm, undercut $\leq 0.25 \times$ 'upper sheet thickness', and root cavity $\leq 0.25 \times$ 'lower sheet thickness'.

An example of the error distribution of such a model is shown in Figure 5. The Figure shows the histogram for the first material combination (HC260LA-0.6mm on HC420LA-1.2mm). It is the overlay of 5-fold cross validation. The inlay on the left side shows some statistical quantities, e.g., 75% of the errors are in the range ± 0.113 mm. The inlay on the right side shows the ϵ -insensitive loss function used in the SVR algorithm to weight the errors. In the case shown, ϵ was set to 0.1 mm. Errors in the range of $\pm \epsilon$ are weighted by zero, they have no influence on the optimization algorithm used to

determine the regression coefficients. Errors outside this range are weighted linearly. These two facts are responsible for the very robust behaviour of SVR with respect to outliers.

3) *Parameter Prediction for Sample Tasks (Original Model)*: New welding tasks were specified by selection of a material combination, specification of demanded values of penetration depth and weld width and in some cases also specification of one of the process parameters laser power, laser focus, or weld velocity. The remaining process parameters were determined by calculation of the level set for the given goals of penetration depth and weld width.

In order to select one parameter set out of the level set, a cost function was defined which prefers smaller cycle time (faster speed):

$$\text{cost} = \sqrt{(d - d_0)^2 + (w - w_0)^2 + (v/100)^2},$$

where d = penetration depth, D_0 = demanded penetration depth, w = weld width, w_0 = demanded weld width and v = weld velocity.

4) *Measurements with a new Laser*: Changed production conditions were simulated by exchange of the laser by an other laser made by an other vendor. Also, the fibre connecting the laser source with the laser head in the robot was exchanged.

The parameters predicted in the previous step were used to perform weld processes. Again, all produced welding seams were cut, sanded and measured by micrography. All measured penetration depths and weld widths were found to be smaller than requested.

As a cross-check, some additional measurements were made with process parameters taken from the original experiments, from which the process models were created. Also in these cases, the results were too small.

The deviations produced by the new laser with respect to the original process models on penetration depth was -0.16 ± 0.16 mm, the deviation on weld width was -0.13 ± 0.10 mm.

5) *Calibration of the Process Models*: The process models were calibrated using the data of the previous step, where all results are out of demanded ranges. Only 37 experiments were used to calibrate the process models, whereas the original models was created by 295 (good quality) experiments out of about 490. It can be expected that the number of required calibration experiments can be further reduced by application of intelligent sampling algorithms.

6) *Parameter Prediction for Sample Tasks (Calibrated Model)*: Based on the calibrated model again new tasks were specified and corresponding process parameters are determined in the same manner as in Subsection III-F3.

7) *Verification of the Predictions by new Laser*: Experiments with the new parameters were executed and evaluated. The deviations produced by the new laser with respect to the calibrated process models on penetration depth was $+0.05 \pm 0.11$ mm, the deviation on weld width was -0.05 ± 0.07 mm. It can be stated that these results are a good improvement compared to the original model. This improvement was reached by only few additional experiments with the new laser. It must be kept in mind, that the evaluation of each experiment is very time consuming because it involves cutting, sanding and micrography.

IV. CONCLUSION

In [1], a concept (called T2MT) was presented for the automatic extraction and representation of process knowledge from

experimental data. It was used to derive process parameters to reach a given goal under given process conditions. The concept was demonstrated in that paper by numerical simulations on resistance spot welding.

In Section II of the current paper, a short review of T2MT is given. Section III applies the methods to LSW and describes additional extensions, which converted the T2MT into a system usable in industry.

The whole concept was now demonstrated to be ready to be applied in industrial environments by experimental verification with real data, sampled from the LSW process. The focus was to demonstrate the advantages by finding good process parameters using T2MT with highly reduced time effort. This time-saving aspect becomes more and more impressive if more process models are available. Data should be gathered from the setup of new machines and from processing of new tasks and should be stored in a database. Process models derived from this database are candidates for the calibration to slightly different tasks, they are the starting point for the generation of new models.

It is worthwhile here to mention the flexibility and portability of the T2MT. The whole framework makes no assumption about the underlying processes, it is exclusively driven by experimental data. The T2MT can also be integrated into machines and perform the automatic parameter finding on-line. In this case, the user needs to describe the demanded task in terms of goal values and process boundary conditions, e.g., materials and sheet thicknesses. The process parameters are determined automatically in this task-driven operation.

ACKNOWLEDGMENT

This work was supported by the EU Project I-RAMP³ (Intelligent Network Devices for fast Ramp-up), project homepage <http://www.i-ramp3.eu/>. I-RAMP³ is co-financed by the European Commission DG Research under the 7th Framework Programme.

REFERENCES

- [1] J. Pollak, A. Sarveniazi, and N. Link, "Retrieval of process methods from task descriptions and generalized data representations," *The International Journal of Advanced Manufacturing Technology*, vol. 53, no. 5-8, pp. 829-840, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s00170-010-2874-1> [retrieved: July, 2015]
- [2] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag, 1995.
- [3] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199-222, 2004.
- [4] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, pp. 1443-1471, 2001.
- [5] C. Yuan and D. Casasent, "Support vector machines for class representation and discrimination," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, July 20-24, 2003, pp. 1611-1616, DOI 10.1109/IJCNN.2003.1223940.
- [6] J.-C. Wang and D. Casasent, "Hierarchical K-means Clustering Using New Support Vector Machines for Multi-class Classification," in *International Joint Conference in Neural Networks*, JUL 16-21, 2006, pp. 3457-3464.
- [7] AWL-Techniek B.V., Nobelstraat 37, NL-3846 CE Harderwijk, (postal address: P.O. Box 245, NL-3840 AE Harderwijk), The Netherlands, Web: <http://www.awl.nl> [retrieved: July, 2015].
- [8] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> [retrieved: July, 2015].

Globally Optimized Production by Co-operating Production Agents Based on Bellmans Principle

Norbert Link

Karlsruhe University of Applied Sciences
 Karlsruhe, Germany
 e-mail: norbert.link@hs-karlsruhe.de

Abstract— The production of items is usually separated into a sequence of processing steps from raw materials to the finished product. Each of the processing steps is executed by dedicated machines where the output of one machine is the input of the next machine. The total effort of all processes can be drastically reduced and the resulting quality of the end product be maximized by exploiting the mutual dependencies of the individual process steps. The concepts of task-driven, intelligent production agents are extended to account for this global optimization task, maintaining the autonomous decision of the individual agent about the optimal process parameters. This can be reached by supplying the local production agent with information about the effect of some of its output on the efforts of the subsequent processes and with information about the actual input to be processed. When the process agent knows the efforts related to its own parameters required to transform the input into some output states, the overall effort can be minimized. Stochastic process influences turn the optimization into a Markov decision process where Bellmans equation can be applied to yield on average the best total result at lowest effort. The encountered exponential complexity when solving Bellmans equation via Dynamic Programming is relieved by Approximate Dynamic Programming. By looking upon one single process, as a process chain with discrete, repetitive steps with different process parameter values, the same optimization concept can be applied to control the individual process. Agents using this optimization scheme require special capabilities: output state estimation, state transformation function representation, Bellman optimization and assessment function representation (assigning effort to process output). The concepts, the architecture, the required components and the methods will be presented in this paper.

Keywords—manufacturing; agent systems; process chain; optimization; control; Markov decision process; Bellmans principle.

I. INTRODUCTION

Optimization of production processes can be performed on different levels: (a) supply chain, (b) workflow, (c) machine. The traditional way of looking at optimization assumes perfectly specified input/output relations on all levels. Deviations are considered as failures with eventual fall-back strategies to react on. In this sense only the most efficient supply chain/workflow has to be found and the best parameter values for the machine settings have to be investigated. The first two optimization tasks are of the discrete type and subject to intense research which has found

its way to corresponding software products controlling supply chains and workflows.

On the machine level, a continuous optimization task in the parameter space is encountered, where an optimum point maximizing a certain objective function (of quality, etc.) or minimizing a certain cost function (of energy, wear, time, etc.) has to be found. Usually, these functions are not given as analytic functions of the process parameters which would make optimization an easy task.

One of the highest remaining potentials, but with also the highest challenges (even scientifically), is to account for the mutual influence of the processes in subsequent processes of a chain. The interrelation between processes is due to the fact that parts, with the same specification, may differ in aspects which are relevant for subsequent processing.

Example: Different milling parameters can produce the same geometry but different surface layer properties (hardness, stresses, grain properties, etc.). Later heat treatment may then affect the geometry differently or subsequent surface processing will need different effort.

In order to achieve an overall optimum over a whole process chain, these relations have to be accounted for. This requires that a process gets the information about its actual input. This information will be generated by the preceding process from its process data by means of a quality model. With this information, the parameters have to be set in a way that the cost related with them plus the expected cost of the subsequent processes, related to the output, minus the price of the final product are minimized. There exist scientific approaches to address this challenge, but they are of high computational complexity and need the full set of information, as mentioned above.

An individual process which needs to compensate fluctuations during processing can be considered as temporal sequence of processing steps where after each step the parameter values for the next step are decided. The time steps can be looked upon like processes in a chain, and the approaches for process chains can be applied for the optimal control of a single process as well.

The paper is organized as follows: The basic concept of an agent-based global optimization is developed in Section II, where also the required information and the additional components of self-optimizing agents are identified. Section III discusses the knowledge extraction and representation methods, which are called "Optimizations models" and Section IV presents the concept and methods of a chain-

optimizing closed-loop controller. Conclusions about the presented concept are drawn in Section V.

II. BASIC CONCEPT AND COMPONENTS

The core requirement to control individual processes in a way, that the result is also optimized for the subsequent processes, is to make the local optimization account for the subsequent processes. This can be achieved in the following way: if the process knows about the different efforts, which are caused by its different end states (results), then it has to consider the sum of its own effort to reach a certain end state plus the efforts of this end state for the subsequent processes. To be more precise, it is the effort or cost expectation value of the subsequent processes related to the end state. Now it needs to find the parameter values which yield the lowest total cost. Figure 1 shows the associated information flow and required data with the example of a three-step process chain, where a product is made from steel sheets by heat treatment and deep drawing of parts, which are welded together in the final step.

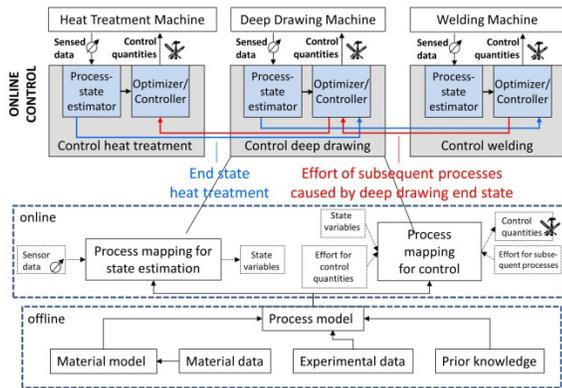


Figure 1. Chain optimization work flow and required components

The information about the „subsequent“ cost is propagated from the process next downstream the process chain (red arrow in Figure 1). To decide upon the control quantity or process parameter values, the information about the initial state is required as well. This is the final state of the preceding process and the information is propagated from there (blue arrows in Figure 1). The process state has therefore to be derived by the preceding process from the data from machine-attached sensors. This transformation from sensor data values to state variable values is represented by the blue box „process state estimator“ in Figure 1, which is using a **quality model** to specify the transformation. The optimizer/controller uses a **state transition model (or process model)** which relates the process state at a future time, to the process state at present under certain values of the control quantities. By virtue of this model, it can find the control quantities minimizing the total cost. This is depicted in the „online“ box in Figure 1, summarizing the activities which must be performed during processing. These activities are controlled by **models** (representing the sensor data – state relation and the state

transition) which are fed into the „process mapping for state estimation“ and in the „process mapping for control“ respectively. These models must be executable in process real time, which means that they must be computationally simple. These models represent the dedicated process knowledge and have to be created in an off-line process. Usually, there are no simple analytical models for production processes, which mean that the models have to be formed from formalized prior expert knowledge and from experimental data (either real or simulated process executions with determinations of the process states). For simulation experiments, material models and data are required.

When all the models are available for the processes of a process chain, the path (sequence of parameter values for the process steps) with minimum total cost can be found. The painful way would be to consider all combinations of all process parameters of all processes and to search the combination with the lowest cost (full search). Fortunately this is not necessary thanks to Bellmans theorem [5] which allows to propagate the cost upstream from the end, and to set up the „subsequent cost function“ for each process in the chain.

In order to enable the downstream optimization, the process machines are extended accordingly with dedicated controls as shown in Figure 1. Such a control is an integrated component of some general production agent (for Instance NETDEV [1]) shell which has to acquire the necessary information and to perform the necessary optimizing control activities.

A process chain is assumed to be a Markov process [8] where each step is transforming the input state of the (semi-finished) product into some output state, and where the final product is the result of a sequence of such otherwise independent transformations.

Describing the

- input state via a vector \vec{x} of state quantities, the
- output state by vector \vec{y} of state quantities and the
- process as a general transformation operator $\vec{\rightarrow}^T$,

we can write a process step as

$$\vec{x}_{i-1} \xrightarrow{T_i} \vec{y}_i,$$

where a process chain is then written as

$$\vec{x}_0 \xrightarrow{T_1} \vec{y}_1 = \vec{x}_1 \xrightarrow{T_2} \vec{y}_2 = \vec{x}_2 \dots = \vec{x}_{N-1} \xrightarrow{T_N} \vec{y}_N.$$

If the control is supposed to perform an optimizing process $\xrightarrow{T_{i,opt}}$, it will adjust the process parameters (or control quantities) \vec{u}_i in a way $\vec{u}_{i,opt}$, that the transformation will produce the lowest effort (cost) related to the parameters $J_{i,loc}(\vec{u}_i)$ necessary to transform \vec{x}_{i-1} into \vec{y}_i and lowest effort (cost) for all subsequent downstream processes related to the produced output \vec{y}_i , namely $J_{i,sub}(\vec{y}_i)$.

In order to do so properly, it requires the following information:

- input state \vec{x}_{i-1}

- process model describing the transformation $\vec{y}_i = \vec{T}(\vec{x}_{i-1}, \vec{u}_i)$, e.g., in functional form $\vec{y}_i = \vec{T}(\vec{x}_{i-1}, \vec{u}_i)$
- cost associated with the process parameters $J_{i,loc}(\vec{u}_i)$
- cost of subsequent processes associated with output state $J_{i,sub}(\vec{y}_i)$

The input state information is supplied by the previous process which uses the sensor and process data to estimate its output state. Each NETDEV control component has therefore to generate this estimate for the next subsequent process. The estimation is transforming the observable sensor and process data \vec{s}_i into an estimated output state $\vec{y}_{i,est}$.

This “measurement” transformation $\vec{y}_{i,est} = \vec{M}(\vec{x}_{i-1}, \vec{s}_i)$ is represented by a second process model, the “inverse measurement model”, which could again be described in functional form $\vec{y}_{i,est} = \vec{M}(\vec{x}_{i-1}, \vec{s}_i)$.

The optimizing process $\vec{u}_{i,opt} = \text{argmin}_{\vec{u}_i} \{J_{i,loc}(\vec{u}_i) + J_{i,sub}(\vec{y}_i) | \vec{y}_i = \vec{T}(\vec{x}_{i-1}, \vec{u}_i)\}$ consists of minimization of the total cost (local plus subsequent) with respect to the process parameters.

$$\vec{u}_{i,opt} = \text{argmin}_{\vec{u}_i} \{J_{i,loc}(\vec{u}_i) + J_{i,sub}(\vec{y}_i) | \vec{y}_i = \vec{T}(\vec{x}_{i-1}, \vec{u}_i)\} \quad (1)$$

The optimization therefore requires the supply of the respective information which is generated via dedicated models. This is depicted in Figure 2.

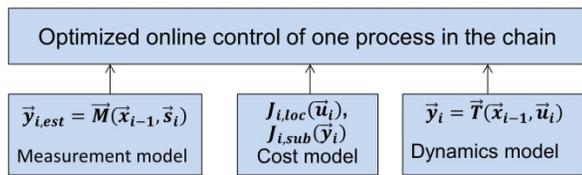


Figure 2. Information, models required for process-chain optimizing controls

During on-line optimization operation, these models are fed with the actual data (sensor, control quantities or process parameters) from which they derive the actual state, the expected output state and the related cost information, which is used by the optimization algorithm (solving Bellmans equation) in order to derive the optimal process parameters.

III. METHODS FOR DERIVING THE OPTIMIZATION MODELS

As discussed so far the models are transformation functions of different kind:

1. The inverse measurement model which relates a process state to sensed process data and recorded process parameters,
2. The process dynamics model which relates a next state to the present state under the action of dedicated process parameter values,

3. The cost local cost model which relates effort (cost) to dedicated values of the process parameters and
4. The subsequent cost model which relates expected cost to dedicated final process states.

The requirements on the transformation models are:

- a. Sufficient precision to enable the optimization algorithms to yield results close to the theoretical optimum and to let them converge there robustly and quickly.
- b. Low computational complexity to allow information delivery at times defined by the process real time requirements.

Requirement b prohibits quite often the direct usage of numerical process simulation or the direct solution of first principal equations during processing time, since in almost all cases both approaches are computationally extremely complex or analytically too difficult for a real process.

This is the motivation, why the usage of numerically highly efficient regression models which are formed from process data is proposed. The split of the overall solution in the two steps of model creation and model application allows specifying low-complexity, real-time suited on-line models, which are efficiently executed during processing. The latter are representatives of a restricted class of the process (defined by the process capabilities of the device), encapsulating the respective process knowledge. The functional form of these on-line models is determined in a much more time-consuming off-line step in advance. Real-time properties can be achieved this way for an optimizing control by representing the transformations in a simple, explicit form which is derived from the existing process knowledge.

A generic method to set up these on-line models is to derive them from process data which are collected to form a representative sample of the input and output quantities of the respective transformations. A regression analysis is then applied to the sample data yielding an estimation of the desired transformation function. For this purpose, an Ansatz function with a set of parameters is defined where the parameters are fitted by means of the sample data set via minimization of a deviation cost function. The Ansatz function can be selected as a set of non-linear base functions (such as polynomial, logistic or Gaussian) using standard methods of (preferentially robust) optimization (M-estimators or RANSAC) or as being composed of a set of pre-defined mathematical symbols where the formula structure and the parameters are optimized via genetic or evolutionary algorithms [7], forming the so-called symbolic regression.

The data sampling can be either performed via laboratory experiments or by numerical experiments if such simulation models are available for the process under consideration. Also, a combination of both can be used, where the majority of samples are created in the computer

via simulation which is again calibrated by laboratory experiments to ensure correct simulation.

The model formation, as discussed so far, complements the required methods. The according extension of the diagram of Figure 2 is shown in Figure 3.

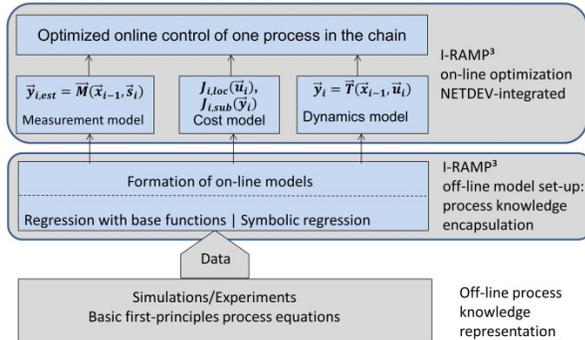


Figure 3. Generic model creation based on representative process data samples

The above discussed model generating components extend the production agent concepts, but not their internal structure.

IV. CONCEPT OF A CHAIN OPTIMIZING CLOSED-LOOP CONTROLLER

We have introduced methods to derive necessary models in a generic way by representing them as transformation functions which are derived via regression from data generated either by numerical or real experiments or from first principles equations. We can generate the necessary input for optimization this way.

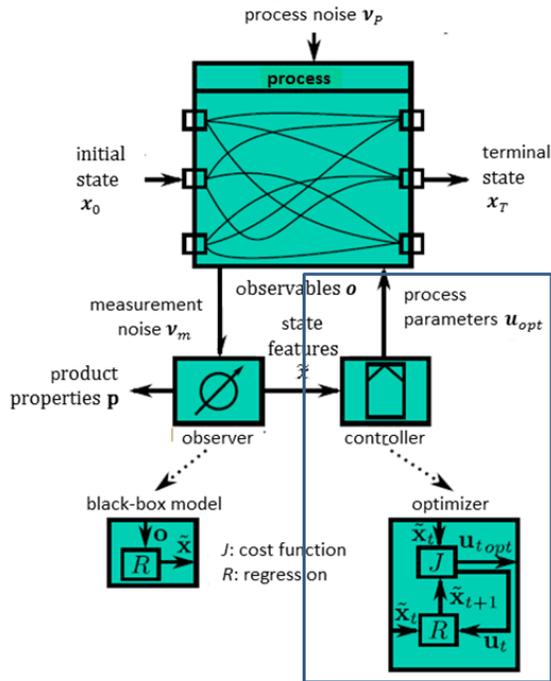


Figure 4. Closed-loop control instantiated by an optimizer

Still a method to find the optimum of the cost function (1) is missing. Figure 4 gives an overview of the local agent context of the closed-loop process control which is instantiated as an optimizer of (1).

The discrete time sequence of a single process or a chain of subsequent processes was introduced in the generic chain optimization concept as a Markov process which is optimized using the Bellman principle [6].

Figure 5 shows a time discrete Markov process where the output x_i of one process i step (green box) is the input of the next process step $i+1$. The output x_{i+1} is the result of the process i (under parameter values u_i). The local cost J_{Di} is associated with the deployed parameter values u_i . From the subsequent process a cost function \tilde{J}_{i+1} is supplied which associates cost with output x_{i+1} . This cost function is the expectation value of the cost, which output x_{i+1} is likely to produce with all subsequent processes minus the price (negative cost) of the expected final state. Since the output x_{i+1} is a result of the input and the process parameter values u_i , \tilde{J}_{i+1} is depending indirectly on u_i as well.

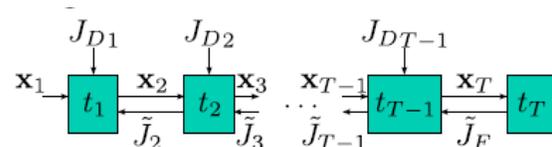


Figure 5. Markov process and Bellman principle

The best parameter values are therefore the ones minimizing the sum of \tilde{J}_{i+1} and J_{Di} which is expressed in

$$u_{t \text{ opt}}(x_t) = \underset{u_t \in U_t}{\operatorname{argmin}} \{ J_{Dt}(x_t, u_t, x_{t+1}) + \langle \tilde{J}_{t+1}(x_{t+1}) \rangle \} \quad (2)$$

Where $\langle \rangle$ denotes the statistical expectation value.

The stochastic influence of disturbances on the process is reflected by the fact, that an output state x_{t+1} is reached from an input x_t state under given process parameter values u_t only with a certain probability $P(x_{t+1}|x_t, u_t)$. Then

$$\langle \tilde{J}_{t+1}(x_{t+1}) \rangle = \sum_{x'_{t+1} \in X_{t+1}} P(x'_{t+1}|x_t, u_t) J_{t+1}(x'_{t+1}) \quad (3)$$

With

$$\sum_{x'_{t+1} \in X_{t+1}} P(x'_{t+1}|x_t, u_t) = 1$$

The expectation values can be calculated in a back-propagation way from the final process where the cost function of the final state is given by the price achieved with the end quality. Given the probabilities of arriving at an end state (with given cost) from a certain initial state, with all possible parameters (and associated cost), will result in an expectation value of the cost associated with the initial state. This way the expected cost can be calculated for all input

states which are the output states of the preceding process. The procedure can now be repeated for the process before the last process and so on, until the initial process is reached and the output states have assigned expected cost values for all processes in the chain. The procedure described so far, yields only such cost values for discrete states. These can be stored in a look-up table which can be used to yield a cost value for a state under consideration via assigning the cost value of the Nearest Neighbor (NN) or via interpolation with the nearest neighbors [9]. A complete function for $\langle J_{t+1}(x_{t+1}) \rangle$ can be achieved for all processes if regression methods are employed as discussed in the preceding section.

This is the backward Dynamical Programming (DP) algorithm of which the pseudo-code is given in Figure 6.

```

Initialize  $J_T(x_T) \quad \forall x_T \in X_T$ 
for  $t = T - 1 : 1$ 
  for all  $x_t \in X_t$ 
    Initialize  $J_{t, opt}(x_t)$ 
    for all  $u_t \in U_t$ 
       $J_{tmp} = C_{Dt}(x_t, u_t, x_{t+1})$ 
      +  $\sum_{x'_{t+1} \in X_{t+1}} P(x'_{t+1} | x_t, u_t) \tilde{J}_{t+1}(x'_{t+1})$ 
      if  $(J_{tmp} < J_{t, opt}(x_t))$ 
         $J_{t, opt}(x_t) = J_{tmp}$ 
      end if
    end for
  end for
end for
    
```

Figure 6. Backward DP algorithm for a finite time horizon

The following models are involved in the optimization

- State transition model $x_{t+1} = f_t(x_t, u_t, v_{pt})$ in the deterministic case with added noise v_{pt} or $P(x_{t+1} | x_t, u_t)$ in the stochastic case, from data via regression
- Cost model $J_t(x_t)$
- Control law $u_t = \pi_t(x_t)$
- Noise model from assumptions or experimental measurements
 - Normally distributed (μ, Σ)
 - Additively superposed with state

Cost considered

- J_{Dt} : Local cost (due to process efforts), derived from vendor knowledge
- J_F : Final (negative) cost (price achieved with end quality), derived from sales knowledge
- \tilde{J}_{t+1} or $\langle J_{t+1}(x_{t+1}) \rangle$: Cost of subsequent process steps due to current state, calculated via backward calculation and regression or explored during processing

The proposed method suffers from the curse of dimensionality [6], when longer process chains and high dimensional state spaces are involved which prevents real-time control. The state space dimensionality can be

drastically reduced by means of dimension reduction methods such as ‘‘Principal Function Approximators’’ (a kind of non-linear partial least squares method) [4] which already yields an optimally reduced state representation, derived from the observed process values. This approach belongs to the class of Approximate Dynamic Programming (ADP) [9]. If only a few process steps are involved, backward DP approach can be applied as was shown with a two stage process in [3].

Other approaches, also allowing the treatment of more complex processes, such as ‘‘Approximate forward Dynamic Programming’’ [2][11] have been investigated in [10].

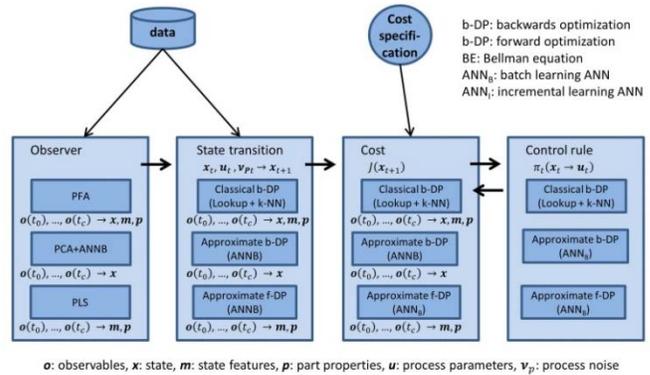


Figure 7. Generic model of a downstream chain optimization control

Combining the observer methods with the different DP approaches finally forms the generic model of downstream process chain optimization (Figure 7).

V. CONCLUSION

A concept has been presented, allowing a process-dedicated production agent to set its process parameter values in a way that a process chain, which it is actually part of, will be (on average) globally optimized. Central optimizer components for each dedicated process chain are no longer required. This makes production more flexible and efficient, since the agent concept is maintained and the efforts are distributed optimally among the process chain members. In order to enable the agents for global optimization, they have to be supplied with extra information about the actual input they have to work on and about the efforts they produce by their outputs with the subsequent processes. This cost function has to be set up by a higher-level component from all downstream process information and is in the responsibility of the process chain agent. A generic method for this purpose has been presented, completing the set of methods contained in the extra components of chain-optimizing methods.

ACKNOWLEDGMENT

The author wants to thank especially Melanie Senn for evaluating, developing and promoting the ADP methods during her work in our research group. The research leading to these results has received funding from the [European

Union] Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no. [314329]."

REFERENCES

- [1] G. Gonçalves, J. Reis, R. Pinto, M. Alves, and J. Correia, "A step forward on Intelligent Factories: A Smart Sensor-oriented approach," *Emerging Technology and Factory Automation (ETFA)*, 2014 IEEE (pp. 1-8), 2014.
- [2] J. H. Lee, "Model predictive control and dynamic programming," *11th International Conference on Control, Automation and Systems*, pp. 1807–1809, 2011.
- [3] M. Senn, J. Schäfer, J. Pollak and N. Link, "A system-oriented approach for the optimal control of process chains under stochastic influences," *AIP Conference Proceedings* 1389, pp. 419–422, 2011.
- [4] M. Senn and N. Link, "A universal model for hidden state observation in adaptive process controls," *International Journal on Advances in Intelligent Systems* 4 (3–4), pp. 245–255, 2012.
- [5] R. E. Bellman, "Dynamic Programming," Courier Dover Publications, Mineola, New York, USA, 2003.
- [6] W. B. Powell, "Approximate Dynamic Programming: Solving the Curses of Dimensionality," 2nd ed., Wiley, Hoboken, New Jersey, USA, 2011.
- [7] M. Davarynejad, J. van Ast, J. L. M. Vrancken and J. van den Berg, "Evolutionary value function approximation," *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 151–155, 2011.
- [8] N. E. Pratikakis, M. J. Realff and J. H. Lee, "Strategic capacity decision-making in a stochastic manufacturing environment using real-time approximate dynamic programming," *Naval Research Logistics* 57 (3), pp. 211–224, 2010.
- [9] J. H. Lee and W. Wong, "Approximate dynamic programming approach for process control," *Journal of Process Control* 20, pp. 1038–1048, 2010.
- [10] M. Senn, N. Link, J. Pollak, and J. H. Lee, "Reducing the computational effort of optimal process controllers for continuous state spaces by using incremental learning and post-decision state formulations," *Journal of Process Control*, 24(3), pp.133-143, 2014.

SMARTLAM - A Modular, Flexible, Scalable, and Reconfigurable System for Manufacturing of Microsystems

Steffen Scholz*, Tobias Mueller*, Matthias Plasch[†], Hannes Limbeck[†], Tobias Iseringhausen[‡], Markus Dickerhof*, Andreas Schmidt*[§], and Christian Woegerer[†]

* Institute for Applied Computer Science

Karlsruhe Institute of Technology

Karlsruhe, Germany

Email: {steffen.scholz, tobias.mueller2, markus.dickerhof, andreas.schmidt}@kit.edu

[†] Machine Vision Profactor GmbH

Steyr, Austria

Email: {matthias.plasch, hannes.limbeck, christian.woegerer}@profactor.at

[‡] Fraunhofer Institute for Manufacturing Engineering and Automation (IPA)

Stuttgart, Germany

Email: tobias.iseringhausen@ipa.fraunhofer.de

[§] Department of Computer Science and Business Information Systems,

Karlsruhe University of Applied Sciences

Karlsruhe, Germany

Abstract—Digital manufacturing technologies are gaining more and more importance as key enabling technologies in future manufacturing, in particular when flexible and scalable production of small and medium lot sizes of customized parts is demanded. This paper describes a new approach to design manufacturing of complex three-dimensional components building on a combination of digital manufacturing technologies such as laminated objects manufacturing, laser, and e-printing technologies. The manufacturing platform is based on a highly flexible and modular approach, which enables manufacturing of different small-sized batches without tool or mask making within a short time. A number of manufacturing modules can be combined by defined hardware and software interfaces. The control system is designed to integrate all processes as well as the base platform with features far beyond ordinary Programmable Logic Controller (PLC) systems.

Keywords—Additive manufacturing; microsystems; flexible scalable system

I. INTRODUCTION

The development of the markets in micro system technology shows that many new and innovative developments in the field of hybrid microsystems do not achieve a satisfactory success. This can be attributed to the high complexity of micro-technical products and processes, a lack of interdisciplinary knowledge in process development, limited flexibility of the applied manufacturing and assembly systems and the high investment risk due to uncertain forecasts of growth. SMARTLAM (Smart production of Microsystems based on laminated polymer films) addresses these issues by introducing a flexible, modular manufacturing approach similar to the Agile Assembly Architecture by Rizzi et al [1]. It is based on a large collection of mechanically, computationally and algorithmically distributed robotic modules, which enables the user to build a production system out of modules in a Lego block manner. The concept of SMARTLAM also features a highly flexible transport system, which allows a free and multiple access to process stations.

II. SMARTLAM CONCEPTS

Digital manufacturing platforms [2] need to meet the challenge of staying competitive with traditional manufacturing systems [3]. Important key factors are extra costs, which emerge from complex hardware and software interfaces, as well as potentially rising idle processing times of single functional modules. A major goal of the SMARTLAM project is to improve competitiveness by designing products in an optimized manner, tailored to the used manufacturing technologies.

In SMARTLAM, each iteration loop can be seen as a workflow consisting of four phases:

- Plan (product specification and design)
- Develop (design support tool)
- Make (process chain selection and process chain configuration)
- Analyze (inspection and process optimization loop)

Figure 1 depicts a scheme of the workflow supported by a 3D-I manufacturing approach, leading from a given product specification to production and product optimization. The corresponding order planning approach is realized through a seamlessly integrated toolchain, which can be sorted along the layer model introduced by the International Standards for Automation (ISA). ISA S95 is an international standard for developing an automated interface between enterprise and control systems as an inherent part of the SMARTLAM project. Four main tools have been realized:

- A capability database, providing information on manufacturing competences (MinaBase),
- a feature-based design tool that enables product design within the systems capabilities,
- a process planning system, which constructs a manufacturing process chain,
- and a configuration tool that enables to set up manufacturing simulation and execution.

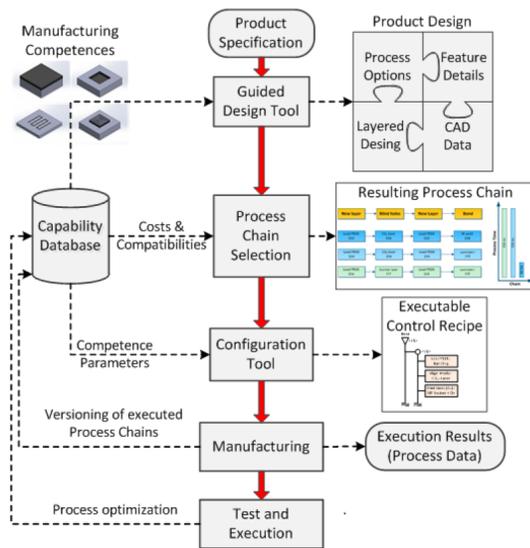


Figure 1. Schematic overview of the 3D-I manufacturing approach

According to Gleadall et al. [4], the feature-based design tool constrains the design engineer in developing a product design that is suitable to be manufactured using integrated SMARTLAM technologies. Therefore, the SMARTLAM 3D-I manufacturing approach supports the development of a microsystem product by using a product concept as the input, which provides a certain degree of flexibility in order to be adaptable to given manufacturing constraints. The manufacturing chain is then developed starting from specifying the technical design of the product design through to configuring and monitoring the actual fabrication process. Also, resulting from its continuously modular design, the system is easily extendable/scalable to cover new manufacturing technologies. Adding a new production technology to the SMARTLAM system requires the following steps to be taken:

- Add a data set of the new manufacturing competence to the database, to make it available to the guided design tool, the process planning system, and the configuration tool.
- Create a component model of the added process module and make it available to the control system. The main controller extracts relevant information, e.g. a protocol and connection description, to establish a link to the new process module.

As the control system's architecture is based on the industrial standard IEC 61499 [3], it is suitable to realize coordination of functional independent modules in order to achieve a given result [5], [6]. Moreover, the openness and interoperability attempts of IEC 61499 reduce development effort that is required for the integration of heterogeneous system components. SMARTLAM supports **seamless data integration**, as its aim is to create an integrated approach to supporting product developers within the scope of the platform in making profound decisions more rapidly. At the core of this is the requirement for a methodology to describe the capabilities and parameters of participating manufacturing technologies and materials. To achieve this, existing knowledge management tools - the Process Capability Database

MinaBase - will be adapted and enhanced for storage and retrieval of SMARTLAM-related process parameters based on a SMARTLAM-specific process knowledge ontology. The customers use a design tool, which is developed in the SMARTLAM project, to design their product for manufacture by a SMARTLAM system. The design tool outputs the design to a process chain selection tool, which predicts the key performance indicators (KPIs) for the customer to review. The process capability database plays a central role in this product development process. In detail, the modules for the current application case include:

- Aerosoljet printing module: printing electronics functionality (circuits, interconnects, sensors)
- Laser structuring module: laser milling and surface modification for structuring or changing polymer surface properties
- Laser welding module: bonding of polymer sheets
- Lamination module: positioning and stacking of polymer films
- Inspection module: inspection of top layer of functional elements in-line and at end of the process
- Assembly module: integration of discrete components into sheet stacks

Based on this approach, different modules for the manufacturing of microsystems are currently being developed within SMARTLAM, which will be connected to the manufacturing databases and back-end processes and finally results in a highly sophisticated and flexible digital fabrication cell. Currently, all modules are being prepared for the assembly of the final production cell. A central control unit to operate the overall manufacturing process including all fabrication, assembly, inspection and transportation steps is being set up. Finally, the overall concept of the D-I-approach has been proved on a laboratory scale by producing a demonstrator specimen.

ACKNOWLEDGMENT

SMARTLAM is funded by the European Commission under FP7 Cooperation Program Grant No. 314580.

REFERENCES

- [1] A. Rizzi, J. Gowdy, and R. Hollis, "Agile assembly architecture: an agent based approach to modular precision assembly systems," in Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, USA, April 20-25, 1997, pp. 1511-1516.
- [2] G. Chryssolouris, D. Mavrikios, N. Papakostas, G. Michalos, and K. Georgoulas, "Digital manufacturing: history, perspectives, and outlook," Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture, vol. 223, no. 5, 2009, pp. 451-462.
- [3] "International Standard IEC 61499-1: Function Blocks - Part 1: Architecture, 1st Ed. International Electrotechnical Commission," Geneva, 2005.
- [4] Gleadall et al., "A design framework for micro devices manufactured by a modular multi-process platform," in Proceedings of the 9th International Workshop of Microfactories, Honolulu, USA, 2014, pp. 16-222.
- [5] D. Ivanova, I. Batchkova, S. Panjaitan, F. Wagner, and G. Frey, "Combining IEC 61499 and ISA S88 for Batch Control," in Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), 2009, pp. 187-192.
- [6] G. Ebenhofer et al., "A system integration approach for service-oriented robotics," in Proceedings of 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation, ETFA 2013, Cagliari, Italy, September 10-13, 2013, 2013, pp. 1-8.

Process State Observation Using Artificial Neural Networks and Symbolic Regression

Susanne Fischer

Intelligent Systems Research Group
Karlsruhe University of Applied Sciences
Karlsruhe, Germany

Email: susanne.fischer@hs-karlsruhe.de

Abstract—Process state observation is important for efficient automated online control in manufacturing. In this paper, we propose a new concept for online observation of the process state. First, we obtain state variables of physically-based numerical simulations of a process. After that, we reduce the state variables to a few characteristic features using artificial neural networks. As a result, we have a process state representation in a lower dimensional feature space. Using this representation, we apply symbolic regression to find a mathematical model that describes the state in the feature space. By applying these methods to a cup deep drawing process, we can describe 99% of the state variables' variation using only 7 features instead of 400. For these 7 features, we can find mathematical descriptions that represent a reduced process model which is used for process state observation.

Keywords—Observation; control; reduction; regression; manufacturing.

I. INTRODUCTION

Process control needs information about the current process state to adjust controllable process parameters. However, if the process state is not directly - or only with large effort - available during the process, a state observer is required to gain and observe process state information from available measurements (observables). For this reason, a process model is necessary to predict the immeasurable process state.

Given that the process model is used for process observation, it is evident that the process model has to be online-capable. For this purpose, black-box models or gray-box models can be applied. Black-box models characterize the process input-output relation. In addition to the input-output relation, gray-box models determine the resulting model structure.

One of the most used methods for implementing a reduced model is the proper orthogonal decomposition (POD), independently proposed by Kosambi [1], Karhunen [2], and Loève [3]. It is also known by Karhunen-Loève theorem (KLT) and principal component analysis (PCA). Rozza [4] covers a wide range of applications in engineering. Using POD, the reduced model $u_N(x, t)$ is represented as a linear combination of basis functions $\phi_j(x)$:

$$u_N(x, t) = \sum_{j=1}^N a_j(t) \phi_j(x), \quad (1)$$

where $a_j(t)$ are time-dependent model coefficients and N is the reduced dimension. It is possible to apply POD to nonlinear problems, but the result leads only to a linear reduction [5].

Established methods for black-box modeling are Kriging and artificial neural networks (ANN). Kriging is based on the interpolation of the simulation results [6]. ANNs are trained

to emulate the system by mapping the process input to the process output [7].

We intend to predict the process state and process dynamic, automated and online-capable. When implementing an online-capable model, one concern is the complexity of the process. Physical-based numerical process simulations predict the process state very accurately, but they are also computationally expensive and hence not feasible for online control. However, the resulting process state of the numerical simulation can be used to create a reduced process model.

We obtain process state variables for each time step and for different process parameters and use artificial neural networks to apply a nonlinear dimensionality reduction to reduce the obtained state variables to a few new state features. To achieve a gray-box model, we apply symbolic regression to describe the time evolution of the state in the reduced space. The determined process model is considered for process state observation by running the process model in parallel to the process.

In Section II, we introduce the several components of the state observation (Figure 1), including the dimensionality reduction algorithm. The result of this algorithm is compared

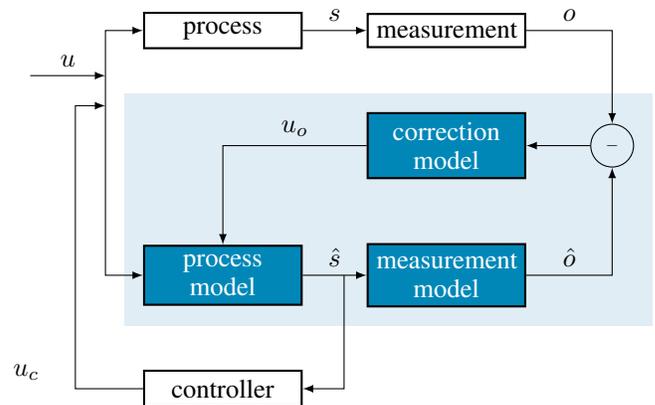


Figure 1. Components of the observer (blue) in parallel to the process.

to the results of the PCA and the nonlinear principal component analysis (NLPCA), introduced by [8]. The feasibility of this process state observation and dimensionality reduction is evaluated with data of a deep drawing finite element model in Section III and discussed in Section IV.

II. STATE OBSERVATION

Standard observers for linear systems are the Luenberger observer [9] and the Kalman filter [10]. The concept for state observation proposed here is based on the Luenberger observer. We adopt the idea to run the observer in parallel to the process (Figure 1) and feed back the error between true observables o and predicted observables \hat{o} to adjust the process model. An additional output is the predicted state \hat{s} to feed the controller with the required information.

The observer components are: (1) the process model that predicts the state for the controller; (2) the measurement model that maps the predicted state to observables; and (3) the correction model. The inner construction of the individual components is explained in the following as well as the required dimensionality reduction.

A. Dimensionality Reduction

The goal of the dimensionality reduction is to reduce the number of state variables of physical-based numerical process simulations to a few characteristic features. The result is a representation of the state in feature space.

PCA is a widely used linear dimensionality reduction method. It starts from the premise that the maximal variance in the data corresponds to the highest information gain and hence small variances constitute negligible information or even noise. Thus, the feature space is described by the directions of the highest variances.

NLPCA, a nonlinear dimensionality reduction method, is based on an autoencoder — a bottleneck neural network where the input has the same meaning as the training target. Here, input and training target are the state variables. The activation of the bottleneck layer represents the extracted features. The NLPCA uses a hierarchical error function to achieve ordered features.

In [11], we described a nonlinear dimensionality reduction method based on artificial neural networks. This approach extracts ordered features that represents the process state in feature space. It has several individual bottleneck neural networks (BNN) that are arranged sequentially (seqBNN), with one bottleneck node each (Figure 2). The input X are the state variables that get reduced through the mapping layer to the bottleneck node. The activation of the bottleneck node represents the extracted feature. We have one network for each feature, so that we can examine each feature separately.

The training target of the first network are the state variables $Y_1 = X$. To construct the second feature, we keep the input and subtract the predicted output \hat{Y}_1 from the true target Y_1 and set it as the training target of the second network. Thus, the second feature is a reduction of the state variables in such a way that the feature reconstructs the variation of state variables that could not be defined by the first feature. This continues until we find n features that describe the state with only a negligible residual.

Thus, the network is trained to reduce the state variables to new features which in turn can reconstruct the state variables. The sum of the predicted output of each network corresponds to the reconstructed state variables. The state variables for the offline training are known through numerical simulation using ABAQUS (finite element analysis software).

B. Process Model

Symbolic regression [12][13], a field of genetic programming, can be used to identify process models. In contrast

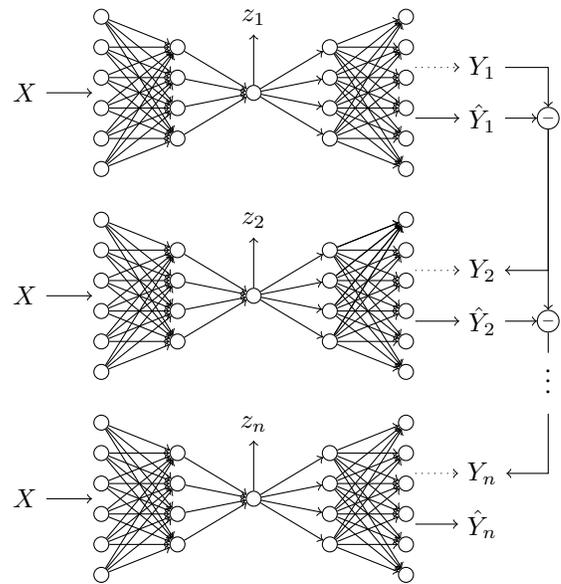


Figure 2. Architecture of the sequential BNN used for the dimensionality reduction.

to conventional regression methods, symbolic regression estimates not only the parameters, but, in addition, the structure of the required mathematical models. It has the advantage that expert knowledge about the structure of the model can be embedded and in turn the resulting model can be interpreted by experts.

Symbolic regression arranges the mathematical model in a binary tree (the so-called individuals) where constants, variables, and/or operators are children of operators (Figure 3). Using an entire population of individuals and genetic

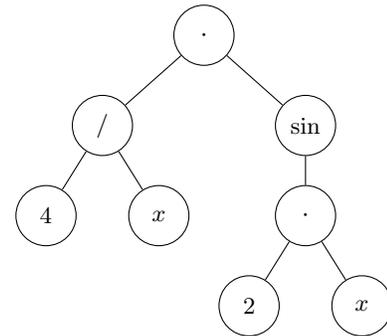


Figure 3. Symbolic regression individual for $\frac{4}{x} \cdot \sin(2x)$.

operations (such as mutation, crossover, and selection), the individuals are changed until an appropriate mathematical model is found that describes the given data points best.

In our case, we are trying to find a model that describes the trajectory of the state in the extracted feature space. The reduced state s depends on time t , the control parameter u_c , and noise ν : $\vec{s}(t, u_c, \nu)$. In this paper, we do not consider the influence of the noise. Consequently, the trajectory of the state \vec{s} can be described by:

$$\vec{s}(t, u_c) = \sum_{i=1}^n z_i(t, u_c) \vec{e}_i, \tag{2}$$

where $z_i(t, u_c)$ is the i -th state feature depending on time t and process parameter u_c , \vec{e}_i the unit vector, and n the dimension.

C. Measurement Model and Correction Model

The measurement model maps the extracted features to the predicted observables \hat{o} (Figure 1). Subsequently, the predicted observables are compared to the true observables o .

Senn [14] shows that a three-layer artificial neural network (input, hidden, and output layer) is sufficient to predict state variables from observables with a negligible error. We assume that the inverse mapping, with the state variables as input and the observables as output of the neural network, provides similarly satisfying results.

The predicted reduced state variables of the process model and true observables, known from the numerical simulations, are used as the input and output, respectively. Thus, the numbers of input nodes and output nodes are predefined. To determine the number of nodes in the hidden layer, we use the following equation, based on [15]:

$$H = \frac{OS - O}{\alpha(I + O)}, \quad (3)$$

where I is the number of input neurons, O the number of output neurons, S the number of samples, and α a problem-dependent adjustable coefficient.

We applied a sigmoid activation function for the hidden layer to model the nonlinear relation between state and observables. The network is trained using the Levenberg-Marquardt backpropagation algorithm [16] to update the weights of the connection between the layers.

The input of the correction model is the difference between the true and predicted observables. The correction model converts this difference into information that the process model can use to update itself. Thereby, the observer can react to its own inaccuracies or noise.

III. RESULTS

In this Section, we apply the presented method to a two-dimensional deep drawing model. The simulation of deep drawing is computed with ABAQUS. The dimensionality reduction is applied to the data that we extract from the simulations with different process parameters. We compare our dimensionality reduction method to the PCA, as a linear method, and NLPCA, as a nonlinear method. Using the results of the dimensionality reduction, we define a reduced process model using symbolic regression.

A. Application - Deep Drawing

As an example process, we used a two-dimensional finite element model of cup deep drawing. Deep drawing is a notable method regarding sheet forming. A metal sheet is clamped between a die and a blank holder (Figure 4). The force of the blank holder is adjustable. During the process the punch presses the metal sheet into an opening of the die.

The numerical simulation of the two-dimensional deep drawing model provides us with observables (such as displacements and reaction forces) and state variables (such as von Mises stress or strain). The simulation consists of 129 time steps and we perform 200 simulations, each with a different blank holder force. The blank holder force is time-invariant and varies between 70 and 100 kN . As a result, we have 9 observables and 400 state variables at 129 time steps for 200 different blank holder forces.

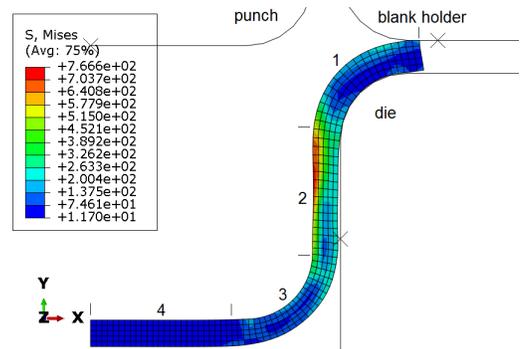


Figure 4. Two-dimensional deep drawing model: drawn metal sheet (v. Mises stress color-coded) clamped between blank holder and die.

We arrange this data in two matrices, one for the observables and one for the state variables. The size of each matrix X is described by $ST \times V$, where S is the number of samples (numbers of simulations with different blank holder force), T the number of time steps and V the number of observables or state variables, respectively.

We normalize the data of X using standard deviation:

$$X = \frac{X - \text{mean}(X)}{\text{std}(X)}, \quad (4)$$

where $\text{mean}(X)$ is the arithmetic mean, and $\text{std}(X)$ is the standard deviation.

B. Dimensionality Reduction

We reduced the state variables (von Mises stress) using our seqBNN approach in Section II-A. The reconstructed state variables are defined by

$$\hat{Y} = \hat{Y}_1 + \hat{Y}_2 + \dots + \hat{Y}_n. \quad (5)$$

These results are compared to the results of PCA and NLPCA. The reconstruction error, the mean square error (MSE) of the true state variables and the predicted state variables (both normalized by (4)), of the three methods is shown in Figure 5.

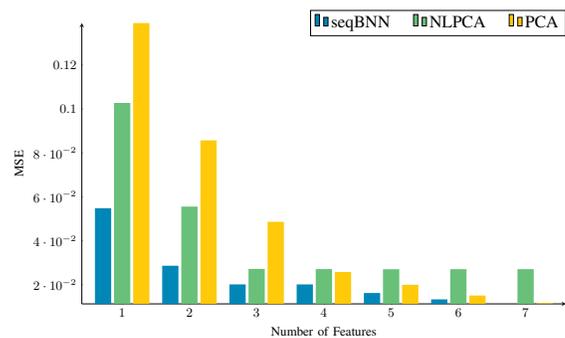


Figure 5. Comparison of reconstruction error (MSE) per feature of seqBNN, PCA and NLPCA.

The results of the methods based on artificial neural networks, seqBNN and NLPCA, depend on the configuration of the network. The network configuration for the seqBNN result in Figure 5 is:

- 8 nodes in mapping and demapping layer each
- 2000 training epochs

- 160 training samples with 129 time steps each
- 40 test samples with 129 time steps each

and for the NLPCA result:

- 16 nodes in mapping and demapping layer each
- 7 nodes in the bottleneck layer
- 2000 training epochs
- 160 training samples with 129 time steps each
- 40 test samples with 129 time steps each

For both methods different configurations, especially with regard to the number of nodes in mapping and demapping layer, yield different results for the higher features. Running different configurations has shown that there is no significant influence on the first two features, as long as the configuration is reasonable.

It is obvious that with increasing number of features the reconstruction error decreases. The error should converge toward zero as the number of features approaches the number of state variables. Furthermore, the diagram shows that feature 4 of the seqBNN could not explain relevant information that would reduce the reconstruction error significantly. The same is true for the feature 5, 6, and 7 of the NPLCA. By comparison, the seqBNN can reconstruct the data with only two features as good as the PCA with four features or the NLPCA with three features.

Figure 6 shows the coefficient of determination R^2 :

$$R^2 = 1 - \frac{\text{SSE}}{\sum_{i=1}^M (Y_i - \text{mean}(Y))^2}, \quad (6)$$

where SSE is the sum of the squared error between true and predicted values, and M the number of samples and time steps. R^2 measures how much of the total variation in the data is

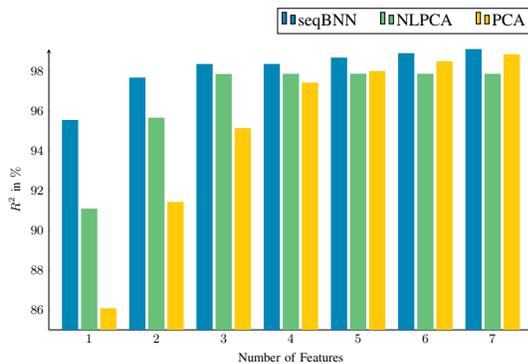


Figure 6. Comparison of R^2 per feature (in %) of the seqBNN, PCA and NLPCA.

described by the variation in the reconstructed state variables. With only one feature for seqBNN we can explain 95.55% of the true state variables variation (in the original domain - without preprocessing), and seven features can explain 99.11% of the example process.

Figure 7 shows the path of the reduced state within the feature space of only the first three features of the seqBNN with a MSE of 0.0201 and R^2 of 98.36%.

C. Process Model

Based on the extracted features, we intend to create a process model using the symbolic regression software Eureqa[®]

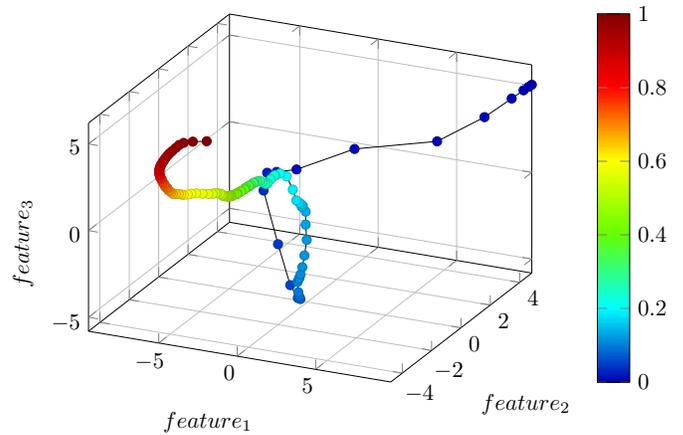


Figure 7. State in three-dimensional feature space. The time is color-coded.

[12]. The quality of the symbolic regression result is measured by the complexity of the mathematical model and the regression error. The complexity of each operation is defined by the user, depending on the prior knowledge, e.g., if it is almost certain that the result contains no \exp -function, then the user can set the complexity of it to a higher number than the complexity of already known operations or exclude it entirely. Furthermore, it is common to set the complexity of basic arithmetic operations to a lesser value than, e.g., trigonometric operations. In our case, we assume no prior knowledge and run the symbolic regression with each operation, constant, and variable set to complexity 1.

For the following results the von Mises stress in each integration point of the finite element model is used as state variables. Figure 8 shows one result for the first feature (created in Section III-B with seqBNN). The resulting equation

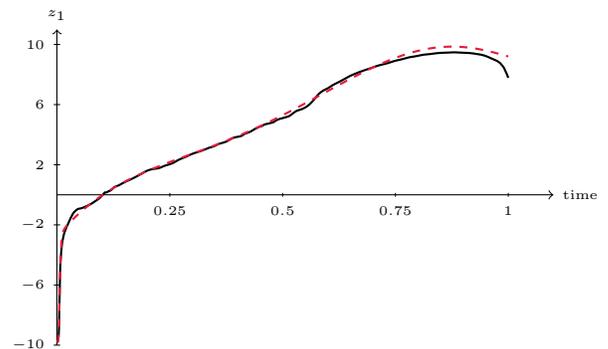


Figure 8. First feature over time (black line) of one specific process parameter u_c and $z_1(t, u_c)$ (red dotted line) using symbolic regression.

with a complexity of 37 and MSE of 0.0327 is:

$$z_1(t, u_c) = -54.29t \cdot 8.60^t + 1.91 \cdot (8.60^t)^2 - \frac{12.80}{1.82 + (1823.35 \cdot 10^8)t^5} + 257.45t^3 + 81.93t - 4.69. \quad (7)$$

An other result with a lesser complexity of 21 but a higher

error of 0.0979 is defined by

$$z_1(t, u_c) = -8.41 \cdot (4.87 \cdot 10^{-40})^{(132.16t^2)} - 3.68t^{10.53} + 14.04t - 1.48. \quad (8)$$

In these results, the process parameter u_c has no influence on z_1 . We assume this is caused by the small range of the variation of u_c (see Section III-A). The obvious variation in the data is caused by the deformation over time. The linear-elastic part at the beginning of the process is obvious (Hooke's law) and so is the beginning of the plastic deformation.

If we consider the normal strain in x -direction in each integration point of the finite element model as state variables, then the extracted z_1 represents the highest variation in this quantity. Thus, we can assume that the derivative of z_1 shows similarities to the rate of change of the deformation. Hence, we are also looking for differential equations:

$$\frac{dz_i}{dt} = f(z_i(t, u_c), t, u_c). \quad (9)$$

One symbolic regression result for the derivative of the first feature of the strain values is explained by:

$$\frac{dz_1}{dt} = 7.53tz_1 + \frac{5.78 + 0.79z_1}{e^{t+z_1}} - 0.01te^{z_1}, \quad (10)$$

with a complexity of 24.

D. Measurement Model and Correction Model

The three-layer artificial neural network is trained with 140 samples and tested with 60 samples with 129 time steps each. We run 2000 training epochs with the extracted state features as input and observables as output.

We experimentally determined the coefficient α in (3), starting with a minimum of 1. Table I shows the MSE and R^2 for different numbers of nodes in the hidden layer. With 35 nodes in the hidden layer, we can explain 99.97% of the variation of the observables.

TABLE I. MSE AND R^2 OF THE TEST SET FOR DIFFERENT NUMBERS OF NODES IN THE HIDDEN LAYER.

	9	12	18	26	35
MSE	0.0064	0.0043	0.0014	0.0006	0.0004
R^2	0.9972	0.9973	0.9989	0.9996	0.9997

Afterwards, the predicted observables are compared to the true observables and the correction model updates the process model with new information. Note that building the process model, i. e., running the symbolic regression algorithm, is not online capable. Thus, for the presented online observation the correction of the process model is only possible using an additional summand.

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduced a new concept for observing state variables of a manufacturing process with the help of a process model, measurement model and correction model.

For this purpose, we reduced 400 state variables of physical-based process simulations to 7 new characteristic features that still describe 99% of the variation of the state variables. This nonlinear dimensionality reduction approach shows promising results for describing the process state in a low-dimensional space.

Before the network training, the NLPCA needs a fixed number of features that have to be extracted. This may result in too few or too many features in order to achieve a desired residual. With our method, however, it is sufficient to define the desired residual before the training and if that has not yet been reached, another feature will be extracted by an additionally network.

We assume that each feature represents a particular area (or part of it) of the metal sheet. This assumption is based on visual comparison of the extracted features with the von Mises stress curve for elements of a particular area, using the finite element model. For example, the stress curves of the elements in the area 1 in Figure 4 are similar to feature 1 and area 2 is close to feature 2. The proof of this assumption is subject of future work.

We proposed symbolic regression to find a process model in the extracted low-dimensional space. First results reveal that a function of time and process parameter can be found for each feature. We also showed first results for learning differential equations, but nevertheless expert interpretation is subject of future work.

The three-layer artificial neural network (measurement model) proved to be sufficient to map the reduced state variables to observables. The variation in the predicted observables describes 99% of the total variation of the true observables. Note that each step:

- dimensionality reduction
- learning of the process model
- regression: predicted state \rightarrow predicted observables

introduces a minor error between predicted results and true results. It has to be ensured that the eliminated information is negligible or noise.

Future work will include the implementation of an optimal process chain controller using the presented process observation for each discrete process in the process chain.

ACKNOWLEDGEMENTS

The funding of this project within the Research Training Group 1483 "Process Chains in Manufacturing" by the German Research Foundation (DFG) is gratefully acknowledged.

REFERENCES

- [1] D. Kosambi, "Statistics in function space." Journal of Indian Mathematical Society, vol. 7, 1943, pp. 76-88.
- [2] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, ser. Annales Academiae scientiarum Fennicae: Mathematica - Physica. Universitat Helsinki, 1947, On linear methods in probability and statistics.
- [3] M. Loève, "Fonctions alatoires de second ordre," in Processus stochastiques et mouvement Brownien, P. Levy, Ed. Gauthier-Villars, 1948, Random functions of the second order.
- [4] A. Quarteroni and G. Rozza, Eds., Reduced Order Methods for Modeling and Computational Reduction, ser. MS&A, Modeling, Simulation and Applications. Springer, 2013.
- [5] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview," Nonlinear Dynamics, vol. 41, no. 1-3, 2005, pp. 147-169.
- [6] M. Strano, "A technique for FEM optimization under reliability constraint of process variables in sheet metal forming," International Journal of Material Forming, vol. 1, no. 1, 2008, pp. 13-20.
- [7] O. Nelles, Nonlinear System Identification, 1st ed. Springer-Verlag Berlin Heidelberg, 2001.
- [8] M. Scholz and R. Vigário, "Nonlinear pca: a new hierarchical approach." in ESANN, 2002, pp. 439-444.

- [9] D. G. Luenberger, "Observing the state of a linear system," IEEE transactions on military electronics, vol. 8, no. 2, 1964, pp. 74–80.
- [10] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Fluids Engineering, vol. 82, no. 1, 1960, pp. 35–45.
- [11] S. Fischer, O. Hensgen, M. Elshaabiny, and N. Link, "Generating low-dimensional nonlinear process representations by ordered features," in 15th IFAC Symposium on Information Control in Manufacturing, 2015.
- [12] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," Science, vol. 324, no. 5923, 2009, pp. 81–85.
- [13] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.
- [14] M. Senn and N. Link, "Hidden state observation for adaptive process controls," in Proceedings of the Second International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE 2010). IARIA, 2010, pp. 52–57.
- [15] W. C. Carpenter and M. E. Hoffman, "Selecting the architecture of a class of back-propagation neural networks used as approximators," Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, vol. 11, 1 1997, pp. 33–44.
- [16] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, Nov 1994, pp. 989–993.

Efficient Implementation Of Network-enabled Devices Into Industrial Environment

Implementation criterias and practical business processes for integrating intelligent network-enabled devices

Martin Kasperczyk

Fraunhofer-Gesellschaft zur Förderung der angewandten
Forschung e. V., Fraunhofer IPA
Stuttgart, Germany
e-mail: martin.kasperczyk@ipa.fraunhofer.de

Eileen Ridders

Steinbeis-Europa-Zentrum
der Steinbeis Innovation gGmbH
Karlsruhe, Germany
e-mail: ridders@steinbeis-europa.de

Abstract - This paper introduces, at early-stage of research, a concept for implementing network-enabled devices (NETDEV) into industrial environment for the phase of ramp-up production. Based on a drafted standard procedure, beside technical aspects, economic ones shall be considered before projecting realization, to ensure cost-effectiveness.

Keywords - Network-enabled device; Efficient Implementation; Evaluation procedure; Ramp-up phase; Economic viability

I. INTRODUCTION

There is no denying that the manufacturing industry is facing a transition towards a completely interrelated knowledge-based business [1]. Intelligent machines, which are able to communicate and are aware of their own status, are becoming indispensable in order to assure and enlarge the competitive advantage and to help the process towards a flexible and sustainable production. This paper describes the business process and respective criteria for on-hand incorporation of NETDEVs into the production line.

A NETDEV is an acronym for Network-Enabled-Devices. These can be considered as logical entities, which encapsulate a device, as equipment or part of equipment, a single sensor or network of sensors or a combination of these components. In general, NETDEVs are able to incorporate build-in intelligence based on a set of models. These models enable the production device to expedite ramp-up process, find optimal process parameters, and support for maintenance and quality assessment. The concept and the NETDEV itself has been developed within the EU-funded research project I-RAMP³ [4], whose goal is the creation of novel solutions for enhancing competitiveness of European industry sector by creating a concept for fast and optimized ramp-up and operation of production lines.

For the majority of companies creating an added value with operating machines, the incorporation of network-enabled devices is projected to be advantageous. However, prior to the implementation into active production lines, it has to be considered that implementations like these require an inter-divisional composition of all business processes, considering evaluation, planning, and change

management. For every company, the de facto person in charge of production output and production costs, e.g., Chief-Technical-Officer (CTO), has to analyze the current state-of-the-art of the manufacturing facility and, based on this, to fathom if the advantages of an implementation prevail.

In Section 2, an overview for a business process for implementing NETDEVs is being provided. Section 3 describes the target definition, as first phase, in more detail, followed by a conclusion in Section 4.

II. GENERAL APPROACH

The process of implementation should go along with existing planning standards, such as the planning process according to VDI 5200-1 “Factory planning” [2] (Figure 1) or other approaches aiming to execute green-field installations, expansions, or re-structuring of existing production facilities. As reference for a planning structure, this instruction was chosen as it is considered as a common planning standard to decision makers within the industrial environment.



Figure 1. Factory planning process according VDI 5200-1 [2]

To ideally gain SMART (“Specific, Measurable, Accepted, Realistic, and Timely”) targets the first phase aims to set the goals for the future project. In the use case of implementing NETDEVs into an existing production line, decisions have to be made on which equipment is technically applicable with the upgrade using NETDEVs. Also, its economic viability has to be reviewed before a final definition on the scope can be determined. This paper is focusing on the process of this phase, which will be examined in detail in the following sections.

After fixing the final definition of the scope, structured requirement analysis, considering framework conditions, use cases and scenarios, roles will follow in a second step. The design of the NETDEV architecture would follow according

to standardized, but customized to the requirements of NETDEV implementation, development methodologies.

In the phase of implementation, acceptance procedures and standardized methods for change management could be applicable. This would go along with appropriate training of personnel, manual working instructions, and documentations, which are based on the results of the requirement analysis and following phases.

III. PHASE I: TARGET DEFINITION

Starting with target definition, it becomes clear that a differentiation has to be made which production machines should be equipped with additional communication and intelligence capabilities, such as the NETDEV.

This analysis is necessary due to the fact that, despite its advantageous aspects, the installation requires human resources for initial installation, integration and customization of the NETDEV for all considered production equipment. Further on, maintenance efforts also would cause additional costs during the future life-cycle of the equipment. Potential realization risks may cause unscheduled downtimes of the machine and would possibly lead to idle periods at process equipment within the value chain.

As a pre-condition, prior to the implementation, an analysis of technical features of the production equipment is indispensable. Giving an example, connecting the equipment, which would have no communication requirements, would also not underlie any flexibility aspects, such as for exchange of tools or changing process recipes, and would not consider any deviating manual working instructions for operators. Applying a NETDEV for this use case should be questioned. Coming along with the fact that the production equipment is an easily exchangeable, multifold redundant and not cost intensive tool, some analysis would be appropriate to evaluate its economically viability. Putting it into a nutshell: a ramp-up exists, in various natures and scopes, for a wide range of production equipment and components. And every ramp-up time causes unused capacities and by that potentially negative impact on life-cycle-costs.

An additional aspect could be its projected life-cycle. Given that installing a NETDEV with equipment which will last for several years makes sense, does not automatically imply that it is profitable to integrate a tool which is definitely worn after a couple of days. (Using a NETDEV to prolong this period of time would be an additional argument for installing.)

Besides technical conditions, additional framework requirements could have an impact on the decision to potentially install a NETDEV. Based on the fact that a certain knowledge and education is necessary in order to handle higher complexity could possibly hamper the installation in individual cases.

These arguments make clear that it is not useful to equip every production machine with a NETDEV and by that to provide additional intelligence and communication abilities to it. Even if arguments would support an installation, it is

important to identify at what level, and respectively which parts of the equipment should be integrated.

It is worth mentioning that NETDEV implementations, triggered by equipment suppliers, in detail would underlie different rules. However, in order to be successful on the market it is required to satisfy customers' requirements.

Despite potential disadvantages and risks, by nature, a range of positive aspects also have an impact on the definition of the scope of integration of NETDEVs.

Additional intelligence and communication skills provide a range of advantages. Due to the bulk of known benefits, in the context of this paper only an extract is provided.

An expedite integration of the production equipment with controlling, Manufacturing-Execution-System (MES), sensors and actors, and other peripherals enables a fast integration when ramping-up, either at initial installation or after a product exchange or an unscheduled downtime.

An instant access to data and information can be achieved. Also additional analyzes by applying "Big-Data", respectively Data-Mining methodologies can be executed. Knowledge can be used to identifying existing or potential problems, and by that may allow identifying potentials for further optimization on the production system.

All of these advantages, by nature, could be executed without any intelligent network-enabled system. However, the efforts to detect, analyze, evaluate, and provide suggestions for optimizations could be significantly higher than with applying a NETDEV or other net-enabled devices providing similar features.

IV. CONCLUSION

The remarkable progress in the field of intelligent production inspired the authors to do research on how new technologies, like the NETDEV, could efficiently be transferred into industrial applications. By applying standardized implementation processes, combined with existing technical and economic planning methodologies, ramp-up phases can be shortened [3]. Implementing a more complex system, like a NETDEV well planned approaches become even more relevant in order to gain significant synergies applying NETDEVs.

ACKNOWLEDGMENT

The authors would like to address special thanks, supporting the progress on this topic to Flavio González (Fraunhofer IPA), Norbert Link (Hochschule Karlsruhe - Technik und Wirtschaft), and Patricia Wolny (Steinbeis-Europa-Zentrum) for their creative ideas and supportive contribution.

REFERENCES

- [1] Bauernhansl, ten Hompel, Vogel-Heuser; *Industrie 4.0 in Produktion, Automatisierung und Logistik*, p.14, 2014 (in German)
- [2] VDI 5200, Part 1, February 2011
- [3] Bruns; *Organisation des Anlaufmanagements*; p.225; 2010 (in German)
- [4] www.i-ramp3.eu [last accessed September 2015]

Optimizing Product Paths in a Production Grid

Leo van Moergestel, Erik Puik, Daniël Telgen
 Department of Computer science
 HU Utrecht University of Applied Sciences
 Utrecht, the Netherlands
 Email: leo.vanmoergestel@hu.nl

John-Jules Meyer
 Intelligent systems group
 Utrecht University
 Utrecht, the Netherlands
 Email: J.J.C.Meyer@uu.nl

Abstract—In a production environment where different products are being made in parallel, the path planning for every product can be different. The model proposed in this paper is based on a production environment where the production machines are placed in a grid. A software entity, called product agent, is responsible for the manufacturing of a single product. The product agent will plan a path along the production machines needed for that specific product. In this paper, an optimization is proposed that will reduce the amount of transport between the production machines. The effect of two factors that influence the possibilities for reductions is shown in a simulation, using the proposed optimization scheme. These two factors are the redundancy of production steps in the grid and the number of steps where the order of execution is irrelevant.

Keywords—Multiagent-based manufacturing; production path planning.

I. INTRODUCTION

Today the requirements for manufacturing are rapidly changing due to newly arrived technologies like 3D-printing and end-user involvement using Internet technology [1]. At the Utrecht University of Applied Sciences, research is done on agile manufacturing. The aim is to achieve low-cost production of small quantities or even single user-specified products. This means that hardware, as well as software should be developed to make this possible.

The hardware that has been developed are cheap reconfigurable devices, called equiplets. Equiplets consist of a basic platform on which specific front-ends can be attached. When a front-end is attached to an equiplet, it will be capable to perform one or several specific production steps.

The software that is used in the production environment, is based on multiagent technology [2]. An agent is an autonomous software entity, having responsibilities and playing a role in the whole manufacturing software infrastructure. Two specific agent roles are the basis of the manufacturing system. The role and responsibility to have a single product made is assigned to a product agent. The role to control an equiplet and to offer production steps is assigned to an equiplet agent [3].

This paper will focus on the product agents and specifically the planning part of its role in the manufacturing. Section II will show an overview of the roles and responsibilities of the product agent. This will also reveal the manufacturing concept used in our research. In Section III the definition of terms used in the paper are introduced and explained. After the

introduction of the terms, the path planning approach is the topic of Section IV. To test this approach, the implementation has been tested in a simulated environment. The results of these simulations are given in Section V on results and discussion. In Section VI, related work is discussed among other work that is related to this new manufacturing paradigm. A conclusion and bibliography will end the paper.

II. AGENT-BASED MANUFACTURING

In the previous section, the concepts of product agent and equiplet agent have been introduced. The manufacturing concept will now be discussed. The equiplets are placed in a grid topology for reasons that will be explained at the end of this section. Each equiplet offers one or more production steps and by combining a certain set of production steps, a product can be made. The set of production steps that can be performed, depends on the type of front-end that is attached to the equiplet. This way every equiplet acts as a reconfigurable manufacturing system (RMS) [4].

Agent technology opens the possibilities to let this grid of equiplets operate and manufacture different kinds of products in parallel, provided that the required production steps are available [3]. Every product requires a given set of production steps and the equiplets in the grid should have these steps available to make it possible to manufacture a specific product. Every product has its own software entity or product agent that is responsible for the manufacturing of a single product. By letting this product agent interact with the equiplet agents the actual manufacturing will take place. In the grid, more than one product agent can be active at any moment, so different products can be made in parallel.

For a product to be made, a sequence of production steps has to be done. More complex products need a tree of sequences, where every sequence ends in a half-product or part, needed for the end product. As a software representative of the equiplet, the equiplet agent advertises its capabilities as production steps on a blackboard that is available in a multiagent system where also the product agents live. A product agent is responsible for the manufacturing of a single product and knows what to do, the equiplet agents know how to do it. A product agent selects a set of equiplets based on the production steps it needs and tries to match these steps with the steps advertised by the equiplets. This selection of equiplets is called the planning and scheduling phase. The planning and scheduling of a product is an atomic action, done

by the product agent in cooperation with the equiplet agents and takes several steps [5]. The planning and scheduling is atomic to prevent problems that arise if more product agents want to schedule steps on equiplets at the same time. If one agent is planning and scheduling, other newly arriving product agents have to wait until the agents finishes the allocation of equiplets. Let us assume that a single sequence of steps is needed.

- 1) From the list of production steps, the product agent builds a set of equiplets offering these steps;
- 2) The product agent will ask the equiplets involved about the feasibility and duration of the steps;
- 3) Next the product agent will generate a path along equiplets;
- 4) The product agent will schedule the product path using first-fit (take the first opportunity in time for a production step) and a scheduling scheme known as earliest deadline first (EDF) [5];
- 5) If the schedule fails, the product agent reports this to the user and proposes a later production time if possible.

For more complex products, consisting of a tree of sequences, the product agent spawns child agents that are each responsible for a sequence. The parent agent is in control of its children and acts as a supervisor. It is also responsible for the last single sequence of the product. In Figure 1, the first two halfproducts are made using stepsequences $\langle \sigma_1, \sigma_2 \rangle$ and $\langle \sigma_3, \sigma_4 \rangle$. These sequences are taken care of by child agents, while the parent agent will complete the product by performing the step sequence $\langle \sigma_4, \sigma_7, \sigma_2, \sigma_1 \rangle$. It means that every single product agent, child or parent, has only a single sequence of steps to perform by itself.

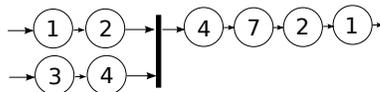


Figure 1. Manufacturing of a product consisting of two half-products

Some important features of the manufacturing model are:

- Every product agent is responsible for only one product to be made;
- The requests for products arrive at random;
- Every product will have its own path along the equiplets during manufacturing;
- The product agent will guide the product along the equiplets.

In the final implementation, a webinterface helps the end-user to design his/her specific product [1]. At the moment all features are selected, a product agent will be created. Because every product can have a different walk along the equiplets, the equiplets are in a grid arrangement that turns out to be more efficient than a line arrangement as used in batch processing [6].

III. STEP PATH AND PRODUCT PATH

This section will define the concepts step path and product path. In Subsection III-A, step path classes will be introduced and in Subsection III-B special cases of step paths are discussed.

Consider a situation where a product is built by 11 production steps. Let us assume that we have 3 equiplets A, B and C. Equiplet agent A offers production step set $E_A = \{1, 2, 3, 4, 8\}$, $E_B = \{5, 6, 7\}$ and $E_C = \{9, 10, 11\}$. The product agent representing our 11-steps product will choose equiplet A first to perform steps 1, 2, 3 and 4. Next equiplet B is used to perform steps 5, 6 and 7. Then, we need again equiplet A for step 8 and finally equiplet C for the last three steps 9, 10 and 11. This so called *step path* is visualized in Figure 2.

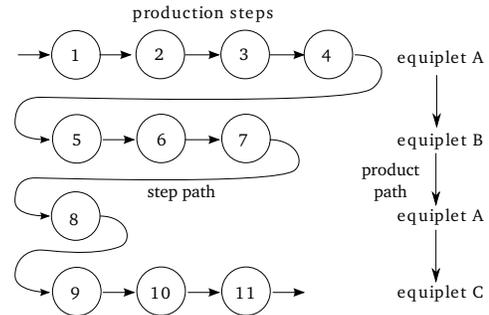


Figure 2. Step path and product path

Definition 1 (Step path). A *step path* is a path along a sequence of production steps that a product agent has to follow to complete a product.

In the example that is visualised in Figure 2 where the step path is shown, another path emerges. This is the path along the equiplet involved. In case of the example, it is a path from equiplet A to equiplet B, from equiplet B to A and finally from equiplet A to equiplet C. This type of path will be referred to as *product path*.

Definition 2 (Product path). A *product path* is a path along a sequence of equiplets that a product agent has to follow to complete a product.

A. Step path classes

In the previous example of our 11-step product (Figure 2) the production steps are in line so our path is a single thread. Figure 3 shows the two possibilities that are considered in this paper: a single line and a tree structure where two half-fabricates are combined.

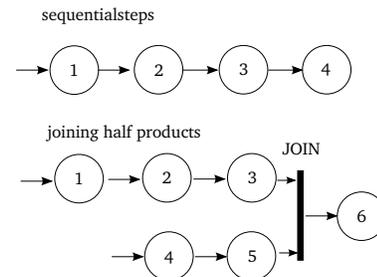


Figure 3. Two combinations of sequential production steps

When these product paths as shown in Figure 3 are written in sets (using: $\{\dots\}$) and tuples (using: $\langle \dots \rangle$) this results in:

- Single path, with tuple notation for a fixed order of steps: $\langle \sigma_1, \sigma_2, \sigma_3, \sigma_4 \rangle$
- Joining half products: $\langle \{ \langle \sigma_1, \sigma_2, \sigma_3 \rangle, \langle \sigma_4, \sigma_5 \rangle \}, \sigma_6 \rangle$

B. Special cases of step paths

In some situations, the order of steps is irrelevant. This results in several possibilities for the step paths. Only one path of these possibilities should be chosen and the number of possibilities is $n!$ in case we have n steps with irrelevant order. This situation can be seen in Figure 4.

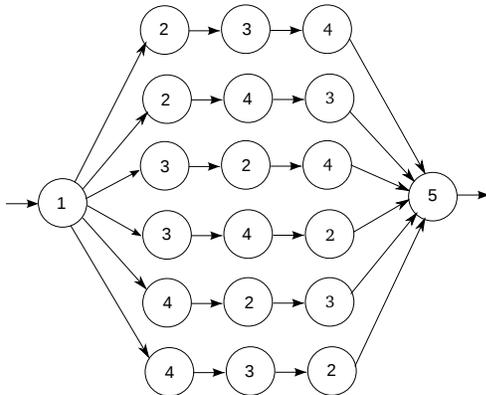


Figure 4. A set of steps with irrelevant order

In formula, this means that the set-notation is used for the steps with irrelevant order:

$$\langle \sigma_1, \{\sigma_2, \sigma_3, \sigma_4\}, \sigma_5 \rangle \quad (1)$$

Parallelism can be achieved if the product has a tree structure as in Figure 5. On the left side of this figure, four incoming arrows, each denoting the start of a production path can be seen. Each path will construct a subpart for the final product and because these paths are independent, these subparts can be made in parallel. At every join in the figure these sub-parts are combined to be input to the next step or steps.

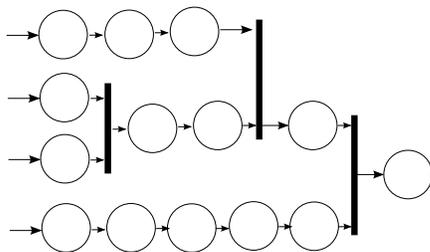


Figure 5. A tree of steps

In the situation of a tree structure a collection of product agents for a single product will be used. The start situation will be one agent, but this agent will spawn child-agents for the separate tuples. The parent agent is in control of its children and acts as a supervisor. It is also responsible for the last single sequence of the product. In Figure 5, we start at the righthand side and walk backwards to the beginning of the production on the left. At every join child-agents will be created. The parent

will wait for its children to complete their subpart. This will be done for every join and will be repeated until the start of the tree structure. When all agents succeed in planning and scheduling, the production will start. At every join the child agents are absorbed by the waiting agent, taking over the collected information and continuing the path until the end is reached as a single agent. This situation arises many times, because most products consist of subparts (Figure 6). The product agent at the root of the tree will finally collect all information from its children. The effect of this decomposition

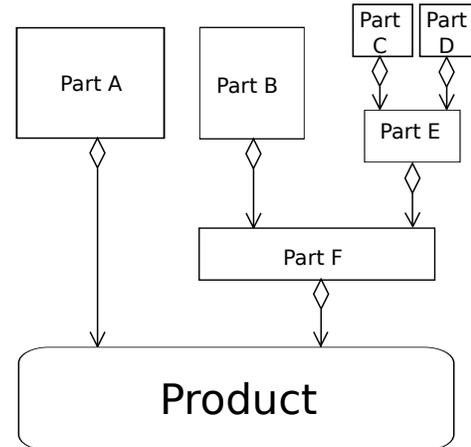


Figure 6. Product consisting of subparts

of complex products is that every product agent only has to deal with a single tuple of production steps. The relationship between these product agents is the fact that they are working on the same product.

IV. PATH PLANNING

A product agent should plan a path along the equiplets. This path will depend on the product steps to be done and the equiplets involved. In this section, matrix-based representations will be presented that will be used in the optimization system that is the main subject of this paper.

A. Step matrix

Consider a grid G of N equiplets, together offering M production steps, this grid can be described by a matrix. This matrix is called *step matrix*. The step matrix G_{step} shows the mapping of equiplets to production steps.

$$G_{step} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix} \quad (2)$$

In this matrix, $a_{ij} = 1$ if equiplet E_i offers step σ_j , otherwise $a_{ij} = 0$.

B. Optimization

A step path is a sequence of production steps. For instance, consider a product to be built with three production steps, this product has step path:

$$\langle \sigma_5, \sigma_2, \sigma_4 \rangle \quad (3)$$

Let us assume a simple grid with four equiplets E_1 , E_2 , E_3 and E_4 , each offering a set of steps. The steps offered by an equiplet are denoted between parentheses as in $E_1(\sigma_1, \sigma_4)$. This grid can be described by this set of equiplets:

$$\{E_1(\sigma_1, \sigma_4), E_2(\sigma_5), E_3(\sigma_2, \sigma_5), E_4(\sigma_3)\} \quad (4)$$

This situation can also be described by the step matrix G_{step} .

$$\begin{array}{ccccc} & E_1 & E_2 & E_3 & E_4 \\ \sigma_1 & 1 & 0 & 0 & 0 \\ \sigma_2 & 0 & 0 & 1 & 0 \\ \sigma_3 & 0 & 0 & 0 & 1 \\ \sigma_4 & 1 & 0 & 0 & 0 \\ \sigma_5 & 0 & 1 & 1 & 0 \end{array} \quad (5)$$

A product agent will make a selection of these equiplets based on the production step or steps that must be performed to construct the product. Next, the product agent will ask the equiplet if the steps offered are feasible given the parameters for the steps. The positive response from the equiplet agent contains an estimated time to complete a given step. This information about the duration of a step will be used in the scheduling phase. When a negative response is received by the product agent it will discard the equiplet. Several solutions to map the steps to equiplets may exist. A sufficient solution for the given situation with a minimum of transitions is:

$$\langle E_3(\sigma_5), E_3(\sigma_2), E_1(\sigma_4) \rangle \quad (6)$$

To find an efficient solution, we try to minimise the transitions or hops between different equiplets. This is done by using a so called production matrix G_p . This production matrix can be derived from the step matrix by selecting the rows of the production steps in the same order as in the tuple that describes the step path $\langle \sigma_5, \sigma_2, \sigma_4 \rangle$.

$$\begin{array}{ccccc} & E_1 & E_2 & E_3 & E_4 \\ \sigma_5 & 0 & 1 & 1 & 0 \\ \sigma_2 & 0 & 0 & 1 & 0 \\ \sigma_4 & 1 & 0 & 0 & 0 \end{array} \quad (7)$$

The production matrix can be reduced by eliminating the columns that contain only zeros. This means that the equiplet on top of this column is not involved in the production of this specific product. In this case the column under E_4 will be removed. This results in a matrix (8) where for every σ_i in this step path a row of a production matrix is created:

$$\begin{array}{ccccc} & E_1 & E_2 & E_3 & \\ \sigma_5 & 0 & 1 & 1 & \\ \sigma_2 & 0 & 0 & 1 & \\ \sigma_4 & 1 & 0 & 0 & \end{array} \quad (8)$$

The rows have the same order as the sequence of steps. Matrix element α_{ij} gives the relation between equiplet E_j and production step σ_x at row i . If the step σ_x at row i is supported by equiplet E_j then $\alpha_{ij} = 1$. Not supported steps result in $\alpha_{ij} = 0$.

Optimization should result in a new matrix, that will be called the path matrix, where α_{ij} has a slightly different meaning and can be different from 1 or 0, giving the product agent a clue for its choice. The product agent will choose the equiplet corresponding with the highest value of α_{ij} . The optimization is minimizing the transitions for a product from equiplet to equiplet. The optimizing algorithm will search for

columns j with sequences of $\alpha_{ij} = 1$ and increment the values in a given sequence by the length of the sequence minus one. This will be done for all columns starting with $\alpha_{ij} = 1$. The matrix of the example has a column under E_3 with a length of 2, with the result that the values of this sequence will be incremented by 1. The production matrix transforms to the path matrix (9):

$$\begin{array}{ccccc} & E_1 & E_2 & E_3 & \\ \sigma_5 & 0 & 1 & 2 & \\ \sigma_2 & 0 & 0 & 2 & \\ \sigma_4 & 1 & 0 & 0 & \end{array} \quad (9)$$

Based on this matrix, the product agent will choose equiplet E_3 for steps σ_5 and σ_2 . The path matrix can be cleaned up by changing values that will not be used in the path to zero.

$$\begin{array}{ccccc} & E_1 & E_2 & E_3 & \\ \sigma_5 & 0 & 0 & 2 & \\ \sigma_2 & 0 & 0 & 2 & \\ \sigma_4 & 1 & 0 & 0 & \end{array} \quad (10)$$

The optimization algorithm works stepwise. First the best starting point is searched for. This will reveal the best equiplet(s) to start with. Let us assume that we have n steps in the step path. This results in a production matrix of n rows. Suppose that the algorithm reveals a maximum set of k steps to be completed by one equiplet as a start. This means that after completing this sequence of k steps, $n - k$ rows, representing $n - k$ steps, should still be done. We reached this point of $n - k$ steps to be done, with the minimum of movements of the product between equiplets. The algorithm is applied to the remaining part (the $n - k$ rows) of the production matrix, without taking into account the previous k rows. We reach a new situation where the number of rows is again reduced. This is repeated until the number of remaining rows is 0. Because of the fact that after every iteration we reach a situation with the minimum of movements of the product between equiplets, the final situation, where the number of rows to be done is 0, will also be reached with the minimum of movements.

C. Region with irrelevant order of steps

Now, consider the situation where there exists a region in the production matrix where the order of steps is irrelevant. This region will be referred to as a region with irrelevant step order. If this irrelevant step order region concerns the whole production matrix, there are no borders with a region where the order is fixed as discussed before. We will discuss this situation first and next a situation where the irrelevant step order region is embedded in two regions with fixed order.

When there are no borders with other regions, the used approach is the following: generate a vector v from the matrix where we sum all separate columns. This means for element v_j of vector v , assuming a matrix with N rows:

$$v_j = \sum_{i=1}^{i=N} \alpha_{ij} \quad (11)$$

From this vector the highest value will be chosen as a start. The irrelevant region will decrease by v_j and a new vector will be generated for the remaining smaller region until all steps needed are taken into account.

When there are borders, a slightly different approach will be used. At the border at the top of the irrelevant step order region, there should be a sequence of at least one step resulting from the fixed step order region. In this case a search will be done to find the best match with this already available sequence from the previous region. The same approach holds for the region at the bottom. A special case in this situation could be a sequence that has the size of the special region. Such a sequence will be called a tunnel and special care should be taken. If there are no matches at the upper or lower border, first matching sequences should be investigated. Matching sequences will not introduce a hop and if these matching sequences at top of border do not cover the whole special region, the tunnel can eventually be used introducing two hops, but if the matching sequences on top and bottom together cover the region only one hop is needed.

Two caveats should be mentioned here. If the boundary with a fixed region has more than one maximum (that should be equal of course), these possibilities should be investigated for the best fit. This means we have to look for the maximum in the fixed region that can be extended to the longest sequence by adding a member of the vector v . The number of maxima (not the maxima itself) will give a clue about where to start, at the top or the bottom border. This will be shown in an example.

Another caveat has to do with overlapping sequences in irrelevant step order region. Consider the situation for a irrelevant region depicted in the matrix (12):

$$\begin{array}{ccccc}
 & E_1 & E_2 & E_3 & \\
 \sigma_a & 1 & 0 & 0 & \\
 \sigma_b & 1 & 1 & 0 & \\
 \sigma_c & 0 & 1 & 1 & \\
 \sigma_d & 0 & 0 & 1 &
 \end{array} \tag{12}$$

Generating the vector v will result in (2, 2, 2). However, the choice to be made depends on the next vector that would result from this choice. If the middle maximum is chosen, the resulting vector v is (1, 0, 1) resulting in a total of two transitions or hops. If the selection was for the first maximum the resulting v would be (0, 1, 2), while choosing the last maximum v would be (2, 1, 0). Both of the latter situations result in only one hop.

As an extra example of the approach discussed so far, consider the situation shown in Figure 7. At the top are two

.
0	3	0	0	3	0	0	0
1	1	0	0	1	1	0	0
0	1	0	0	0	1	0	0
1	1	0	0	0	1	0	0
0	0	0	0	1	1	0	0
0	2	0	0	0	0	0	0
.

Figure 7. Border situations

maxima having a value of 3, resulting from the evaluation of the previous fixed order region. The bottom has only one maximum, also resulting from the evaluation of the following fixed order region. The vector for the irrelevant order region is: (2, 3, 0, 0, 2, 4, 0, 0). Because the bottom border has the least number of maxima, we start there to fit with the vector values.

If we started at the top, we could choose the first fit, but then we would lose the fitting possibility at the bottom. If the fitting at the bottom is performed, only one row is left to be handled, having vector (0, 0, 0, 0, 1, 1, 0, 0). This vector fits with one of the maxima at the top border resulting in matrix (13), having only one hop.

$$\begin{array}{cccccccc}
 . & . & . & . & . & . & . & . \\
 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 . & . & . & . & . & . & . & .
 \end{array} \tag{13}$$

V. TEST RESULTS DISCUSSION

To test the optimising approaches discussed in the previous section, test sets have been generated. All these sets consist of 8 matrices. Thus, 8 equiplets are assumed and the production requires 32 steps. First the effect of redundancy is investigated. This is done by using test sets where at every row, there are 1, 2, 3 or 4 choices for equiplets to perform a certain step. The results of using the optimization approach are shown in Figure 8. In this figure, a slight decrease in the number of hops is shown. This is expected due to the fact that higher redundancy gives rise to longer sequences of steps on the same equiplet, thus reducing the number of hops.

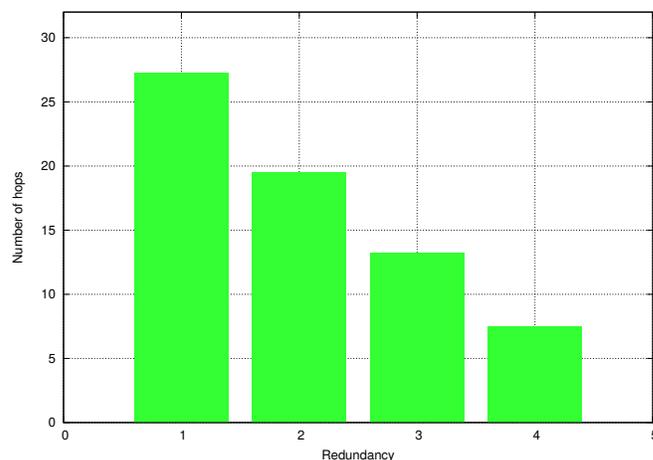


Figure 8. The effect of redundancy

Next, the effect of the size of the region where the order is irrelevant is investigated. This is done by using the same test sets that have been used to see what the effect of redundancy is. The size of the region where the order of steps is irrelevant, changes from 0 (no special region) until 32 (the whole matrix is a special region) in steps of 4. The special region is always placed in the middle of the matrix. The results for the test sets where the redundancy is only 1, are shown in Figure 9. In the subsequent Figures 10, 11 and 12 the results are shown for test sets having a redundancy of 2, 3 and 4. In all figures a decrease of the number of hops can be seen. This is also a result that is expected, because a region where the order of

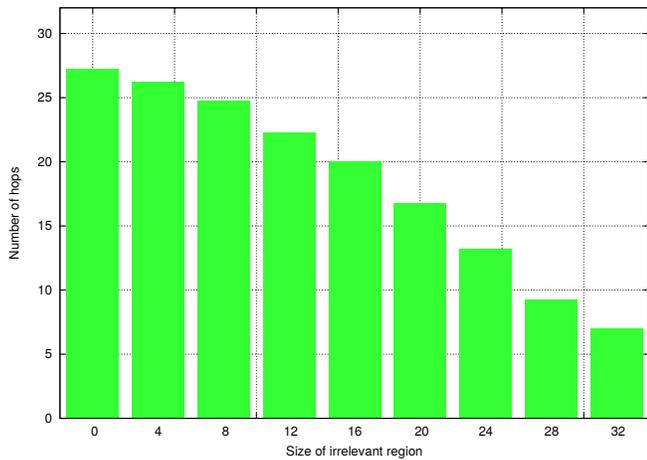


Figure 9. The effect of the size of the irrelevant step region

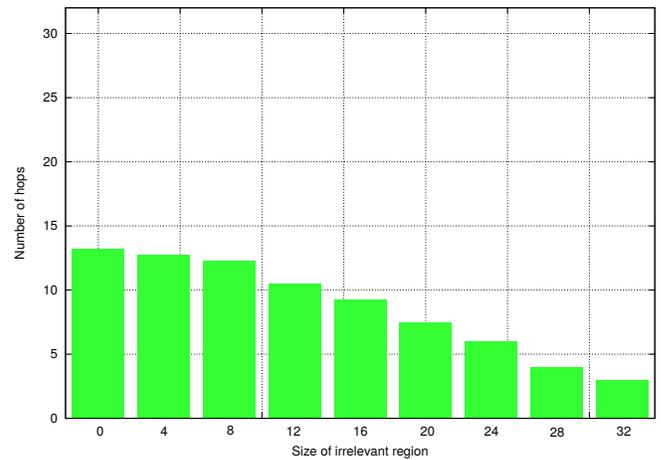


Figure 11. The effect of the size of the irrelevant step region

steps is irrelevant opens more possibilities to generate longer sequences of steps on the same equiplot.

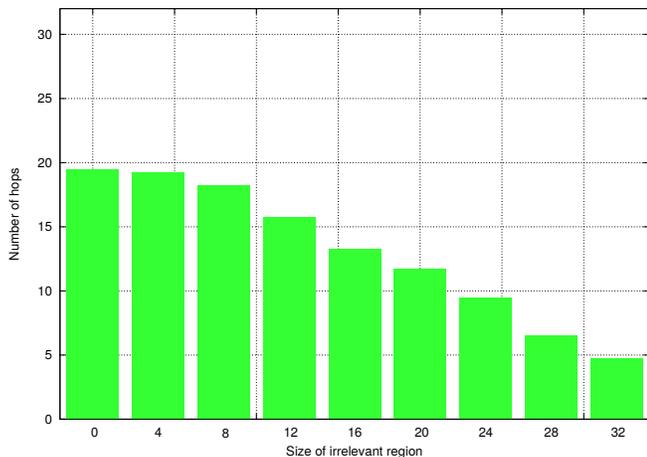


Figure 10. The effect of the size of the irrelevant step region

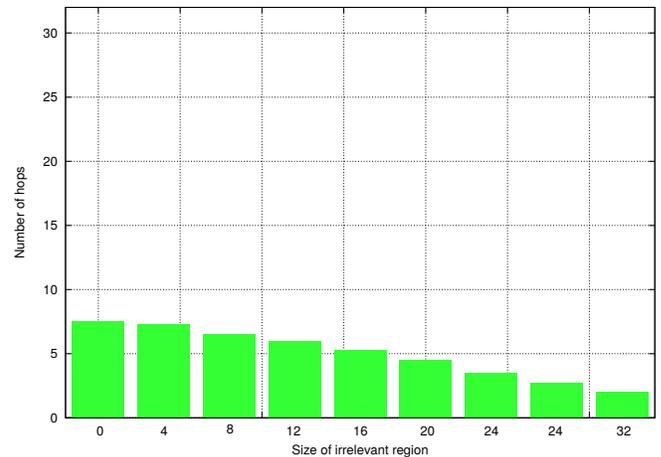


Figure 12. The effect of the size of the irrelevant step region

The test sets were used to test the optimising approach presented in the previous section and this approach turns out to work in the given situations. Important is to emphasize that reduction in transition or hops is an important optimization for the grid production paradigm. In [6], a transport system for the grid is described. This transport system is based on the use of automated guided vehicles (AVG). It turned out that the transport system becomes the bottleneck in a manufacturing grid if the production steps are relatively short, which is in our system mostly the case. The optimization proposed here, will reduce the amount of AVG traffic and therefore alleviates the problem described in [6].

The optimization used and described in this paper was developed with the grid-based agile manufacturing environment in mind. The same approach can also be used in situations where workers are available, who can do one or more specific tasks needed in a certain project and where cooperation is required to reach the final goal of the project.

VI. RELATED WORK

In this section, an overview will be given on agent-based manufacturing. Especially the planning part will be given attention. The Important work in field of agent-based manufacturing has already been done. Paolucci and Sacile [7] give an extensive overview of what has been done. Their work focuses on simulation as well as production scheduling and control [8]. The main purpose to use agents in [7] is agile production and making complex production tasks possible by using a multiagent system. Agents are also proposed to deliver a flexible and scalable alternative for manufacturing execution systems (MES) for small production companies. The roles of the agents in this overview are quite diverse. In simulations agents play the role of active entities in the production. In production scheduling and control agents support or replace human operators. Agent technology is used in parts or sub-systems of the manufacturing process. The planning is mostly based on the type of planning that is used in MES. This type of planning is normally based on batch production. We based the manufacturing process as a whole on agent technology. In our case, a co-design of hardware and software was the basis. The planning will be done on a single product basis and not

on batch production.

Bussmann and Jennings [9][10] used an approach that compares in some aspects to our approach. The system they describe introduced three types of agents, a workpiece agent, a machine agent and a switch agent. Some characteristics of their solutions are:

- The production system is a production line that is built for a certain product. This design is based on redundant production machinery and focuses on production availability and a minimum of downtime in the production process. Our system is a grid and is capable to produce many different products in parallel;
- The roles of the agents in this approach are different from our approach. The workpiece agent sends an invitation to bid for its current task to all machine agents. The machine agents issue bids to the workpiece agent. The workpiece agent chooses the best bid or tries again. This is what is known as the contract net protocol. In our system the negotiating is between the product agents, thus not disrupting the machine agents;
- They use a special infrastructure for the logistic subsystem, controlled by so called switch agents. Even though the practical implementation is akin to their solution, in our solution the service offered by the logistic subsystems can be considered as production steps offered by an equiplot and should be based on a more flexible transport mechanism.

So there are however important differences to our approach. The solution presented by Bussmann and Jennings has the characteristics of a production pipeline and is very useful as such, however it is not meant to be an agile multiparallel production system as presented here. Their system uses redundancy to overcome the the problem that arises in pipeline-based production when one of the production systems fails or becomes unavailable. The planning is based on batch processing.

Other authors focus on using agent technology as a solution to a specific problem in a production environment. The work of Xiang and Lee [11] presents a scheduling multiagent-based solution using swarm intelligence. This work uses negotiating between job-agents and machine-agents for equal distribution of tasks among machines. The implementation and a simulation of the performance is discussed. We did not focus on a specific part of the production but we developed a complete production paradigm based on agent technology in combination with a production grid. This model is based on two types of agents and focuses on agile multiparallel production. The role of the product agent is much more important than in the other agent-based solutions discussed here. In our model, the product agent can also play an important role in the life-cycle of the product [12]. The design and implementation of the production platforms and the idea to build a production grid can be found in Puik [13].

VII. CONCLUSION

In this paper, a path planning optimization approach has been proposed and tested. The optimization turned out to work as expected and results in a reduce of traffic among the production machines. The optimization might be useful in other situations as well, especially in situations of production

systems where the transport becomes a bottleneck. In future research, other step classes can be included like the situation where the order of sequences (tuples) of steps is irrelevant.

REFERENCES

- [1] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Implementation of manufacturing as a service: A pull-driven agent-based manufacturing grid," Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications (ICTERI 2015), Lviv, Ukraine, 2015, pp. 172–187.
- [2] M. Wooldridge, *An Introduction to MultiAgent Systems*, Second Edition. Sussex, UK: Wiley, 2009.
- [3] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Decentralized autonomous-agent-based infrastructure for agile multiparallel manufacturing," Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2011) Kobe, Japan, 2011, pp. 281–288.
- [4] Z. M. Bi, S. Y. T. Lang, W. Shen, and L. Wang, "Reconfigurable manufacturing systems: the state of the art," *International Journal of Production Research*, vol. 46, no. 4, 2008, pp. 599–620.
- [5] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Production scheduling in an agile agent-based production grid," Proceedings of the Intelligent Agent Technology (IAT 2012), Macau, 2012, pp. 293–298.
- [6] L. v. Moergestel et al., "A simulation model for transport in a grid-based manufacturing system," The Third International Conference on Intelligent Systems and Applications (Intelli 2014), Seville, Spain, 2014, pp. 1–7.
- [7] M. Paolucci and R. Sacile, *Agent-based manufacturing and control systems : new agile manufacturing solutions for achieving peak performance*. Boca Raton, Fla.: CRC Press, 2005.
- [8] E. Montaldo, R. Sacile, M. Coccoli, M. Paolucci, and A. Boccalatte, "Agent-based enhanced workflow in manufacturing information systems: the makeit approach," *J. Computing Inf. Technol.*, vol. 10, no. 4, 2002, pp. 303–316.
- [9] S. Bussmann, N. Jennings, and M. Wooldridge, *Multiagent Systems for Manufacturing Control*. Berlin Heidelberg: Springer-Verlag, 2004.
- [10] N. Jennings and S. Bussmann, "Agent-based control system," *IEEE Control Systems Magazine*, vol. 23, no. 3, 2003, pp. 61–74.
- [11] W. Xiang and H. Lee, "Ant colony intelligence in multi-agent dynamic manufacturing scheduling," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 4, 2008, pp. 335–348.
- [12] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Embedded autonomous agents in products supporting repair and recycling," Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2013) Mexico City, 2013, pp. 67–74.
- [13] E. Puik and L. v. Moergestel, "Agile multi-parallel micro manufacturing using a grid of equiplots," Proceedings of the International Precision Assembly Seminar (IPAS 2010), 2010, pp. 271–282.

Self-organising Smart Components in Advanced Manufacturing Systems

Rui Pinto, João Reis, Ricardo Silva, Vitor Sousa, Gil Gonçalves

Institute for Systems and Robotics
Faculty of Engineering of University of Porto
Porto, Portugal

e-mail: { rpinto, jpreis, rps, vdsousa, gil }@fe.up.pt

Abstract—Virtualization of shop-floor components as a way to foster easy access to machine information, collaboration among shop-floor components and task execution on demand are a few key aspects of the latest trends related to Intelligent Manufacturing. This concept is being explored in an ongoing European commission funded project called Intelligent Reconfigurable Machines for Smart Plug&Produce Production (I-RAMP³). The goal is to shorten the ramp-up phase time by making manufacturing systems self-aware and self-diagnosable, increasing the reliability and responsiveness of production systems, and ultimately improving the European industry competitiveness. To achieve this goal, a device virtualization was developed for industrial equipment, such as machines and sensors, called NETWORK-enabled DEVICES (NETDEVs). As a technological background, PlugThings Framework was used for easy sensor integration, together with Universal Plug and Play (UPnP) Architecture for device virtualization, enabling standardized communication, dynamic sensor location, collaboration and diagnostics. The main purpose of the present paper is to describe how the collaboration between a virtualized sensors network was implemented, and pinpoint all the advantages that come out of this.

Keywords—Wireless Sensor Networks; Intelligent Systems; Manufacturing Systems; Sensor Diagnostics and Validation; Sensor Location System.

I. INTRODUCTION

I-RAMP³ [20] is an ongoing European Project funded by the Seventh Framework Program of the European Commission. This collaborative project involves both academic and industrial partners from Germany, Portugal, Netherlands, Hungary, France, and Greece. Therefore, the vision is to improve the European Industry competitiveness by developing technologies for smart manufacturing systems. To achieve it, the goal is to reduce the ramp-up phase of the shop floor equipment and manage efficiently the scheduled and unscheduled maintenance phases, increasing at the same time the efficiency of manufacturing. By virtualizing all shop-floor equipment into an agent-like system, standardized communication skills and a layer of intelligence for collaboration between complex machines, and sensors & actuators are introduced, improving also the plug'n'produce concept towards flexible smart factories. In this context, each agent is represented as a NETDEV, where three variations were considered: Sensor & Actuator (S&A) NETDEV; Device NETDEV; Process Analyzer NETDEV.

The S&A NETDEV is the entity responsible to encapsulate sensors and actuators deployed on the shop-floor, with the intent of monitoring the machines' conditions and the surrounding environment. The Device NETDEV represents

the shop-floor machines, such as a Robotic Arm or a Linear Axis. The Process Analyzer NETDEV, in contrast to the previous entities, does not encapsulate a physical entity, being instead a virtual instance responsible to monitor machines' status and diagnose the sensor networks' condition. NETDEVs have a standardized way to communicate with each other using Device Integration Language (DIL), which is a proprietary task-driven language created in I-RAMP³, in order to ease the quick delivery and reception of process information between all the virtualized shop-floor equipment. The transparency of discovering devices in the network and data exchange between them, using publish-subscribe services, is possible due to UPnP [21] as a base technology.

Sensor data is extremely important to monitor machines at the shop-floor level and its environmental surrounding conditions for condition-based monitoring, machine diagnosis and process adaptation to new requirements. The I-RAMP³ technology allows Wireless Sensor Networks (WSNs) to become more flexible and agile, acquiring new capabilities that can enhance shop-floor operations, such as sensor group collaboration, which aims for providing to the machine aggregated information instead of quantitative data that normally comes in form of raw format. Additionally, it allows for dynamic sensor node location, used on sensor collaborations, to detect if sensor nodes are physically near to each other and to the machine, for the correct interpretation of data, and adaptation of its behavior accordingly.

In the past years, WSNs have become more explored and applied in several domains because of the latest advances on WSN communication protocols (such as ZigBee [22] and others) and more reliable and long-lasting hardware. This is mainly due to its feasibility of installation. WSNs are used when it is difficult to use wired solutions, either because of harsh location or a high number of sensors used. Also, they are used due to ease of maintenance and reduced costs of cabling [13]. Some of the advantages of WSN listed by Chen et al. [14] include its large coverage area, fast communication via Radio Frequency (RF), self-organisation throughout a direct communication between entities and ubiquitous information. As Ruiz-Garcia et al. [15] pinpoint, some of the WSN advantages can be seen in concrete structures or in the transportation sector, where a controlled environment needs to be monitored in real-time. Additionally, Evans [16] presents enablers and challenges, along with some contextual applicability of WSN in a manufacturing environment.

Specifically for the industrial domain, Ramamurthy et al. [13] developed a Smart Sensor Platform that applies the plug'n'play concept by means of hardware interface, payload,

communication between sensors and actuators, and ultimately allows for software update using ‘over-the-air’ programming (OTAP). Cao [17] explored a distributed approach to put closer sensors and actuators in a collaborative environment using WSNs. Chen et al. [14] push this approach forward considering the same approach, but taking into account all the industrial domain restrictions like real-time, functional safety, security, energy efficiency, and so forth. All these industrial restrictions and an overview about the industrial domain was explored and presented by Neumann [18]. In the recent past, Chen et al. [19] tackled the Optimal Controller Location (OCL) in the context of industrial environment.

The paper is composed of five more sections where all the details about the present work are specified. In Section II, an overall description about the I-RAMP³ project is done, specifying the entities used and the communication processes. Section III talks about the sensor collaboration functionality where the communication protocols and sensor failing handling is presented. Section IV depicts the WSN location system used to locate the sensors on the shop-floor. In Section V, a discussion is made based on company personnel perspective of the system and all the functionalities developed, and, finally, in Section VI some conclusions are drawn and future prospects are presented.

II. INDUSTRIAL APPLICATION OF WIRELESS SENSOR NETWORK

Innovative concepts are being explored in I-RAMP³ related to WSNs and their use in the industrial domain, implementing a higher level of complexity using entity virtualization. With the NETDEV concept, sensors and actuators will be equipped with standardized communication capabilities and intelligent functionalities such as self-awareness, self-diagnosis and self-organization, aiming for a smart sensor approach. Moreover, the system should be flexible enough to allow the integration of sensors from various manufactures, minimizing the efforts needed by automating this process. The PlugThings Framework [1] is used to integrate sensors on the system and encapsulate them into S&A NETDEVs. It is composed of 4 main modules: Universal Gateway (UG), PlugThings Server, PlugThings Database and PlugThings App.

As can be seen in Figure 1, each sensor node of the network communicates directly to the gateway node, where the received measurements are processed on the UG, converting raw data into readable form. These data is compiled into Extensible Markup Language (XML) based format files that are part of the Sensor & Actuator Abstraction Language (SAAL), which is used to communicate with Sensor & Actuator Abstraction Middleware (SAAM), where all the intelligence related to the sensors is implemented. When the SAAM detects that a new sensor node was connected to the network, the corresponding S&A NETDEV is created, making it transparent to all the entities on the network what tasks it can perform. Since a sensor node can have multiple sensors integrated, the corresponding S&A NETDEV will be able to perform different tasks related with the different sensor types of the sensor node. Basically, S&A NETDEVs will have one

functionality (execution task) to provide sensor information to other entities per integrated physical sensor in the sensor node. S&A NETDEVs can easily communicate with other NETDEVs on the network using DIL, such as Device NETDEVs that correspond to complex machines on the shop floor level, and the Process Analyzer NETDEV, which corresponds to a virtual instance that monitors sensor behavior while in a group collaboration.

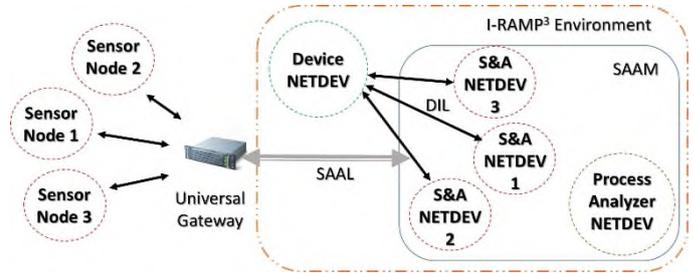


Figure 1. I-RAMP³ Environment

A. Communication between NETDEVs

DIL is a proprietary language used exclusively to communicate between NETDEVs and it is composed of four main XML files: NETDEV Self-Description (NSD) describes the device capabilities in the form of a range of tasks that the NETDEV can perform, such as goals, conditions, process parameters values and also the physical location of the corresponding sensor node; Task Description Document (TDD) specifies information about a task to be requested, specifying the goals, conditions, process parameters and the period of the task execution or number of task repetitions; Quality Result Document (QRD) describes the result after one task repetition, specifying the quality that has to be achieved; Task Fulfillment Document (TFD) is used as an acknowledge document to the task under execution.

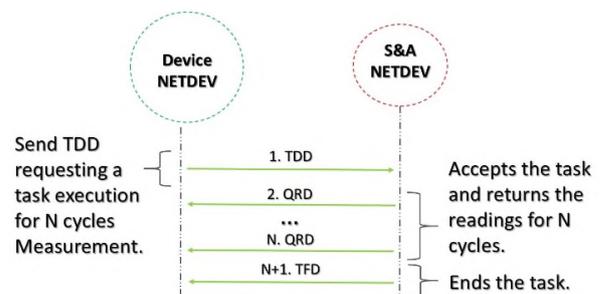


Figure 2. DIL Communication

As represented in Figure 2, DIL is used every time a NETDEV needs other NETDEVs to perform a task. Normally, Device NETDEVs request the S&A NETDEVs task executions for measurement of certain environmental variables, during a given number of cycles, by sending a TDD. If the S&A NETDEV is capable of performing such requested task, it will give a positive feedback via TFD, and answering with QRDs containing the measurement results, during the

number of cycles requested. If the S&A NETDEV is not capable of performing the task or it is already busy performing a task for other NETDEV, it will deny the task sending a denial TFD back to the Device NETDEV.

B. S&A NETDEV Task Execution

At this stage, S&A NETDEVs can execute two different tasks, both usually requested by a Device NETDEV: *Measurement* and *Group Formation*.

A *Measurement* task is used when the Device NETDEV needs the measurements of a single sensor node. Therefore, it should specify the desired type of sensor to receive the corresponding sensory data, the frequency of the readings, sensor accuracy, coverage radius of the sensor in spatial units (if applicable) and the number of cycles to execute the task.

A *Group Formation* task is requested when the Device NETDEV aims to collect several measurements at different locations, which means having multiple sensors executing the same task at the same time. In this specific task, the S&A NETDEV that receives the task is responsible to choose possible S&A NETDEVs candidates to join the group - based on the task parameterization and the sensor location - allowing for a more distributed approach in terms of collaboration, rather than a peer-to-peer-like solution, implying a communication with all the S&A NETDEVs from a group instead of only one. In terms of parameterization, beside the desired type of sensor to receive the specific data, frequency of measurements, sensor accuracy and the number of cycles to perform the task, the *Group Formation* parameterization must also specify the number of sensors intended in the group.

With this collaboration task, there are two main benefits from the task requester perspective. Assuming a Device NETDEV wants to collect and analyze data from multiple S&A NETDEVs, first, it avoids communicating with several S&A NETDEVs at the same time to collect data, since the responsibility to form a group is on the S&A NETDEV, and second, the S&A NETDEVs can process all sensor data and provide a statistical description, passing the data analysis complexity to the group side. This means that the requester does not need to know any statistical technique to process the data from multiple sensor entities on the network.

III. SENSOR & ACTUATOR NETDEV COLLABORATION

A. S&A NETDEV Group Formation

S&A NETDEV *Group Formation* is a methodology used to improve the communication performance and reduce complexity between Device NETDEVs and S&A NETDEVs while executing tasks with a sensor collaboration nature. On the shop floor level there can be thousands of sensors, and therefore, the flow of information can be very high when requesting tasks. The group formation methodology is a more distributed approach that allows S&A NETDEVs to provide a more aggregated information when the task requested from a Device NETDEV requires measurements from more than one sensor node. Instead of establishing communication with every S&A NETDEV required, the Device NETDEV will

have a single point of communication with one S&A NETDEV, which is responsible to form and manage a S&A NETDEVs group.

The main premise for the *Group Formation* is that every S&A NETDEV is capable of forming a group. When a Device NETDEV requests a S&A NETDEV to form a group with a certain number of sensors, this S&A NETDEV is responsible to search in the network (communicating via DIL) for available S&A NETDEVs that are capable of performing the same task and the corresponding sensor nodes are physically located in the same production area. If the requested number of S&A NETDEVs has joined the group, the S&A NETDEV responsible to form it becomes the group leader, called Super S&A NETDEV, and the group is formed. Internally in the group, each S&A NETDEV will collect measurements during the requested number of cycles and the Super S&A NETDEV is responsible, not only to gather all sensor data, but also process them to a more meaningful value, to be sent afterwards to the Device NETDEV. When task execution has ended, the Super S&A NETDEV will terminate the communication with the Device NETDEV and release the S&A NETDEVs from the group, which become available to execute other task requests.

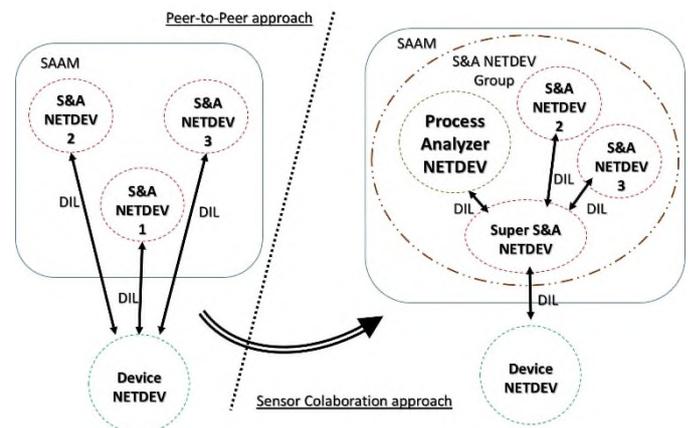


Figure 3. Group Formation Schema

An additional NETDEV entity represented in Figure 3 is the Process Analyzer NETDEV, which is created by the Super S&A NETDEV when the group is created. As previously mentioned, this entity is virtual, not representing any device on the shop-floor, and is responsible to apply the Spatial Correlation technique [11][12] to assess the condition of the group based on the sensor data generated. This entity collects the sensor data from each element of the group and identifies the most devious dataset by comparing the data sets from all group members. If the deviation is greater than a predefined threshold, then the sensor node is classified as probably malfunctioning, so the Process Analyzer reports to the Super S&A NETDEV, via DIL, the existing of a malfunctioning group member at that time so it can make a decision about the faulty sensor node(s) and maintain the group as consistent and reliable as possible.

B. S&A NETDEV Group Formation Fail

Having one single point of communication to interact with all S&A NETDEVs for a task execution is a good way to reduce complexity and increase the performance of communication. On the other hand, relying only on one single point of communication increases the vulnerability, in case the task execution fails on that point. Hence, there are two failing scenarios on a group: 1) The Super S&A NETDEV fails or 2) One or more S&A NETDEVs from the group fail.

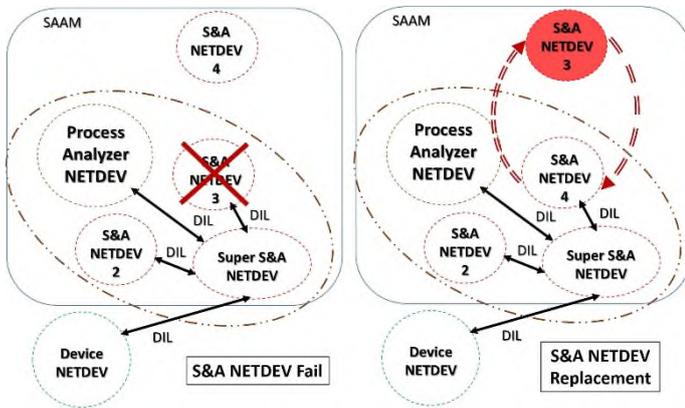


Figure 4. Group Formation - S&A NETDEV Failure

If the Super S&A NETDEV fails, the single point of communication supporting the interaction between the Device NETDEV and S&A NETDEVs from the group is lost. There will be no more conditions to continue with the task execution, so the task stops and the group is disaggregated. In the termination process, the Super S&A NETDEV is responsible to change the process state of the remaining group members, so they can stop executing the measurement tasks for the group, becoming available to perform new tasks upon request from other NETDEVs.

If a S&A NETDEV from the group is failing, the Super S&A NETDEV is still working correctly, so the group isn't in danger of collapsing and the communication with the Device NETDEV is not affected. In this case, the Super S&A NETDEV is responsible to replace the failing S&A NETDEV for a new one able to join in. While the replacement process occurs, the collected data from the group will be less accurate, because the results sent to the Device NETDEV don't contemplate all the requested NETDEVs, due to a temporary deficit of one S&A NETDEV. Figure 4 depicts the process when the S&A NETDEV 3 fails and is replaced by S&A NETDEV 4.

IV. WSN LOCATION SYSTEM

WSNs applied on industry are used to monitor different production cells on the shop-floor, consisting of spatially distributed sensor nodes, which are equipped with several sensors to monitor the environmental conditions surrounding the cells where they are located. If a machine, located in one of the production cells needs information about, e.g., the luminosity conditions surrounding the cell to execute a given

task, the machine may require from available sensor nodes placed in that location, valuable information for process parameterization.

In the I-RAMP³ context, the Device NETDEV that is requesting the task should search on the network for available S&A NETDEVs with the required capabilities (described in the NSD), e.g., measuring luminosity conditions and, consequently can form a sensor group that measures luminosity. Facing a request for a group formation task from a Device NETDEV, the S&A NETDEV will only accept the task if it can fulfil the required parametrization and it is located on the same area as the machine that requested the task in the first place.

Every NETDEV is characterized by its task execution capabilities (NSD) and the area on the shop-floor where the correspondent equipment is located. The location on S&A NETDEVs can be calculated dynamically on sensor nodes that are using XBee communication protocol, using the incoming signal strength of the sensor node on several beacons for position estimation. Beacons are physical entities located in known strategic positions of the shop-floor, mainly in the limits of shop-floor sections like cells or production lines and are responsible by receiving messages from sensor nodes, assess their signal strength and position in order to assign the current relative location to S&A NETDEVs.

A. Methodology

Location systems on WSN is a very active research area and there is no universal solution for this topic. The main goal is to identify the physical location of a sensor node on the WSN. Each approach of node location is fitted to a specific operating environment, such as indoors or outdoors spaces like urban areas, forests or even underwater. In the industrial context, estimating the node positions in meters is not important, as the main goal is to find in which section on the shop floor the sensor nodes are located.

The algorithms for node location are made of two main components: 1) Estimation of distance or angle between two nodes and 2) Calculation of the node position. First, the distance or angle between two nodes must be estimated to be used on the calculation of the node position related to one or more beacons. Then, the information about the distance and the position is used by an algorithm to determine the node's location.

For distance estimation between the sensor node and the beacons, the Received Signal Strength Indicator (RSSI) method is used [2]-[4], which is a method that estimates the distance between two nodes based on the strength of the signal received on the gateway and a propagation model of the signal, in this case Free Space model [8][9], to convert the signal strength into distance. Since the accuracy provided by RSSI is enough for what it's intended in this scenario, this is the cheapest method to be implemented when compared to time delay and time difference based methods or signal angle/direction estimation methods, because measuring the signal strength doesn't required any extra hardware, such as transmitters and receivers of ultra-sounds, like in the Time Difference of Arrival (TDOA) [3] or specific antennas, like in

Angle of Arrival (AOA) / Direction of Arrival (DOA) [6][7], and no need for clock synchronization on the nodes, like required on Time of Arrival (TOA) / Time of Flight (TOF) [5].

The radio signal is highly susceptible to noise [10] caused by reflection, refraction, diffraction, scattering, fading, inter-symbol interference and shadowing, resulting in distance deviations in the end. This can be minimized by filtering the signal using a moving average to better approximate the path loss logarithmic curve. The path loss coefficient is determined dynamically using path loss log-distance model using measurements of RSSI between beacons, using (1), where $P(d)$ is the RSSI in dBm, $P(d_0)$ is the RSSI at a fixed reference distance from the transmitter d_0 , n is the path loss coefficient, X_σ is a normal random variable used to modulate, d is the distance in meters between transmitter and receiver, P_{TX} is the transmission power and A is the signal attenuation. Manipulating the formula, first the path loss coefficient is calculated using (2), where the RSSI and distance are between beacons. Then, (3) is used to calculate the distance between a sensor node and a beacon.

$$P(d) = P_{TX} + A - 10n \times \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1)$$

$$n = \frac{|P(d_0) - P(d)|}{10 \log d \times 2} \quad (2)$$

$$d = d_0 \times 10^{\frac{|P(d_0) - P(d)|}{10n}} \quad (3)$$

The node position is calculated using the distance estimation of the three anchor nodes closest to the sensor node with the Bounding Box method [2]. Bounding Box is a variation of the trilateration, which uses the position of three anchor nodes, with known positions and distances between them, to calculate the position of the sensor node. The position of the node is calculated by the interception of three circles, each one is centered on the anchor node and with radius equal to the distance to the unknown position node. With Bounding Box, the calculation complexity is reduced by replacing the circles by squares. The intersection of the different squares results on a rectangle, where the center is the estimated position of the node.

V. DISCUSSION

As discussed several times throughout the present paper, the use of WSNs is referred as a key element for the today's Manufacturing Systems, pushing forward the plug'n'produce concept, which is achieved by virtualizing shop-floor equipment into NETDEVs that can readily describe and detail their own capabilities and announce themselves into the network to other NETDEVs. NETDEVs are able to collaborate and execute shop floor tasks on demand, and therefore deliver an easy and flexible solution for the industrial domain.

As described, the collaboration between sensors by means of *Group Formation* task available at the S&A NETDEV

entity is, not only a way of reducing the communication entropy when several measurements from neighbor sensors need to be collected, but it also provides higher information about a set of sensors. Additionally, the Process Analyzer NETDEV provides feedback about the condition of the WSN making use of Sensor Validation techniques already explored in the literature and tested in manufacturing environments. Since all these functionalities refer to the software level of abstraction as a way of closing the loop for a ready solution to be used, also the hardware level was considered by means of location device functionality. This allows to know, with a certain degree of precision, the location of sensors in a restricted area, influencing and guiding how sensors should organize and collaborate among themselves, ensuring the system reliability and effectiveness.

Considering now a user perspective like Manufacturing System Designers or Technical Personnel of a Manufacturing company, there are benefits that should be highlighted. Based on the fact that most manufacturing environments are currently using wired sensors instead of WSN, the cabling complexity and savings in terms of time and cost can be reached. This means that no sensors need to be connected to a PLC or Machine Controller, which can be challenging due to the amount of sensors used and harsh locations. On the other hand, the easiness to integrate a new sensor into the system is achieved by only switching on a sensor node, which is automatically recognized as an S&A NETDEV becoming ready for use. This is referred as the plug'n'produce concept, that allows to rapidly react to any foreseen and unforeseen event, like sensor replacement, sensor addition for redundancy purposes in critical environment or in the case of sensor removal when disassembling a production line.

Another advantage of this approach is related with all the functionalities already available from a dedicated framework, releasing the user from being concerned about sensor collaboration and data processing. He only needs to take care of sensor integration using the S&A NETDEV template solution. From that point, information can be easily accessed, monitored and diagnosed. Thus, it is not required for the final user to know in detail, and mainly, to implement from scratch a WSN diagnostics system, but instead, he can focus on what to do when a certain malfunction has occurred and how to relate sensor group information with the product life-cycle in terms of process parameterization. This point is enhanced with the automatic process of forming, deforming and reacting to sudden changes in a sensor group, based on a certain task parameters and sensor location. Since the communication between NETDEV entities is based on a standardized task-driven XML language – DIL - it's very easy to implement a new system that encapsulates a machine, capable of communicating with these entities and easily interpret sensor information for process monitoring.

The main advantage of this is the formation of a self-reconfiguration capability when facing sudden sensor breakdown. A remedy for the breakdown diagnosed by the Process Analyzer NETDEV is embedded in the S&A NETDEVs collaboration, capable of handling a WSN

restructure, as described in the *Group Formation Fail* subsection. In a real manufacturing environment situation, the shop-floor operator only needs to look for the broke sensor (information already provided by the Process Analyzer NETDEV) and replace it by a new one at the same location as the broke one, and automatically the sensor group will reconfigure itself to take on board the new sensor, not being necessary to write or rewrite any lines of code or to disconnect and connect wires.

These functionalities, together with the automated process for diagnosing and logically organizing a sensor group, plus the fact that a standardized communication language is used, are the cornerstones for intelligent WSN in the factories of the future.

VI. CONCLUSIONS AND FUTURE WORK

Innovative intelligent systems have driven technology for years, and industry has followed this track as a way to improve reliability, responsiveness when facing requirements changes from customer side or due to production downtime, efficiency to minimize costs and effectiveness to increase production quality.

All these goals made the guidelines for the S&A NETDEV development, with functionalities to share information, self-organize, collaborate as a sensor group by using a location system for identifying the positioning of motes at the shop-floor level. Therefore, taking advantage of these functionalities can greatly influence the decrease of ramp-up, scheduled maintenance and unscheduled maintenance times, resulting on a competitive advantage in current harsh and fluctuating markets.

The main developments presented throughout the paper depict that, in terms of WSNs applicability in industry, there are open opportunities to explore, and much can be done to improve the currently used systems. Despite all functionalities presented in this paper, the clear benefits it can bring to the shop-floor and all the experience acquired from I-RAMP³, the acceptance of WSNs into industrial context needs to be worked out, by performing more pragmatic and real test-case demonstrators. The present work is a clear step forward into a reliable and flexible approach for industrial WSNs, aiming for paving the way into more intelligent manufacturing systems.

ACKNOWLEDGMENT

This research was supported by project I-RAMP³ (FoF.NMP.2012-3) – Intelligent Network Devices for fast Ramp-up – funded by the European Commission under the Seventh Framework Programme for Research and Technological Development.

REFERENCES

- [1] G. Gonçalves, J. Reis, R. Pinto, M. Alves, and J. Correia "A step forward on Intelligent Factories: A Smart Sensor-oriented approach." *Emerging Technology and Factory Automation (ETFA)*, 2014 IEEE. IEEE, 2014. pp. 1-8.
- [2] K. Vandenbussche "Fine-grained Indoor Localisation Using Wireless Sensor Networks." *Signal*, 75, 2005, p. 70.
- [3] A. Savvides, C. C. Han, & M. B. Strivastava "Dynamic fine-grained localization in ad-hoc networks of sensors." *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 166-179.
- [4] P. Bahl, V. N. Padmanabhan, & A. Balachandran "Enhancements to the RADAR user location and tracking system." technical report, Microsoft Research, 2000.
- [5] Y. Shang & W. Ruml "Improved MDS-based localization." *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 4. IEEE, 2004, pp. 2640-2651.
- [6] D. Niculescu & B.Nath "Ad hoc positioning system (APS)." *Global Telecommunications Conference, 2001. GLOBECOM'01*. IEEE. Vol. 5. IEEE, 2001, pp. 2926-2931.
- [7] N. B. Priyantha, A. K. Miu, H. Balakrishnan, & S. Teller "The cricket compass for context-aware mobile applications." *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 1-14.
- [8] T. L. Singal *Wireless communications*. Tata Mcraw Hill Education Private Limited, 2010.
- [9] C. Levis, J. T. Johnson, & F. L. Teixeira *Radiowave propagation: physics and applications*. John Wiley & Sons, 2010.
- [10] A. F. Molisch *Wireless communications*. John Wiley & Sons, 2007.
- [11] A. B. Sharma, L. Golubchik, and R. Govindan "Sensor faults: Detection methods and prevalence in real-world datasets." *ACM Transactions on Sensor Networks (TOSN)*, 6(3), 2010, p. 23.
- [12] J. Ravichandran and A. I. Arulappan "Data Validation Algorithm for Wireless Sensor Networks." *International Journal of Distributed Sensor Networks* 2013 (2013).
- [13] H. Ramamurthy, B. S. Prabhu, and R. Gadh, "Wireless industrial monitoring and control using a smart sensor platform," *Sensors Journal*, IEEE, 7(5), 2007, pp. 611-618.
- [14] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun "Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks", *Industrial Electronics, IEEE Transactions on*, 57(12), 2010, pp. 4219-4230.
- [15] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and I. Robla "A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends", *Sensors*, 9(6), 2009, pp. 4728-4750.
- [16] J. J. Evans "Wireless sensor networks in electrical manufacturing", *Electrical Insulation Conference and Electrical Manufacturing Expo, 2005. Proceedings*. IEEE, 2005, pp. 460-465.
- [17] X. Cao, J. Chen, Y. Xiao, and Y. Sun "Distributed collaborative control using wireless sensor and actuator networks," *Future Generation Communication and Networking, 2008. FGCN'08. Second International Conference on*. IEEE, 2008, pp. 3-6.
- [18] P. Neumann "Communication in industrial automation—What is going on?." *Control Engineering Practice*, 15(11), 2007, pp. 1332-1347.
- [19] K. Xin, X. Cao, J. Chen, P. Cheng and L. Xie "Optimal controller location in wireless networked control systems", *International Journal of Robust and Nonlinear Control*, 25(2), 2015, pp. 301-319.

- [20] Intelligent Network Devices for fast Ramp-up: I-RAMP3. [Online]. Available from: <http://www.i-ramp3.eu/> 2015.10.02
- [21] A. Presser, et al. "Upnp device architecture 1.1.", *UPnP Forum*, 22, 2008.
- [22] P. Baronti, et al. "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards.", *Computer Communications*, 30(7), 2007, pp. 1655-1695.

Self-Diagnosis and automatic configuration of smart components in advanced manufacturing systems

Rui Pinto, João Reis, Vitor Sousa, Ricardo Silva, Gil Gonçalves

Institute for Systems and Robotics
Faculty of Engineering of University of Porto
Porto, Portugal

e-mail: { rpinto , jpreis , vdsousa , rps , gil }@fe.up.pt

Abstract—One of the key elements for the next generation of Intelligent Manufacturing is the capability of self-diagnosis, where the machinery used can itself report any breakdown or malfunction based on data, and self-reconfiguration as a way to improve responsiveness in case of sudden requirement changes, either by customer request or production line downtime. All these capabilities allow for quicker and improved systems reliability, leveraging the critical production phases as ramp-up, scheduled and unscheduled maintenance. Based on these premises, the main intent of the project Intelligent Reconfigurable Machines for Smart Plug&Produce Production (I-RAMP³) is to develop innovative concepts such as NETWORK-enabled DEVICES (NETDEVs) acting as a technological shell to all the shop-floor equipment, converting it into an agent-like system and tackling the existing gaps between hardware and software for improving the European Industry.

Keywords—Wireless Sensor Networks; Intelligent Systems; Manufacturing Systems; Sensor Diagnostics and Validation; Over the air Programming.

I. INTRODUCTION

Considering the current European industrial panorama, there's still a discrepancy between the mass production hardware solutions and the easy access and monitoring of generated data using software implementations. The low abstraction level of controllers used nowadays in industry do not ease the integration with other existing solutions like Information Systems or other statistical analysis applications. Most of these systems rely on controllers and corresponding Human-Machine Interfaces (HMIs) to monitor the process, and no information is easily accessed for further analysis. Moreover, due to the low level of coding required for any change on these hardware solutions, it's hard to adapt to new production requirements and modify the process parameters. In terms of Assembly Line life-cycle, the impact of the previous constraints lead to high ramp-up times in the early stages of the production, as well as right after scheduled and unscheduled maintenance phases.

Based on these facts, the European Project I-RAMP³ aims for developing innovative concepts for shop-floor devices' virtualization, enabling the easy access, process monitoring and control, as a way to foster the European Industry competitiveness. This virtualization is accomplished by using an agent-like concept called NETWORK enabled-Device (NETDEV) with standardized communication, self-description of device's capabilities, negotiation techniques and

plug'n'produce concept for easy device integration. The NETDEV entities explored in this document are divided into Device NETDEVs, which virtualize shop-floor machinery, and Sensor & Actuator (S&A) NETDEVs, which encapsulates shop-floor sensors or motes.

One of the key factors explored in I-RAMP³ is the use of Wireless Sensor Networks (WSNs) and its capabilities for self-organization and self-diagnosis in the industrial domain. The flexibility in wireless communication, along with low energy consumption, reliable data acquisition and easy deployment *in situ* are just few of the benefits explored so far. Sensor data on a WSNs can be highly susceptible to errors due to external influences, communication conditions and network problems [18]-[21]. Together with the NETDEV encapsulation, sensors become self-aware and consequently diagnose themselves when a breakdown occurs.

Other intelligent feature about S&A NETDEVs is its self-reconfiguration capability. Since the Over-The-Air Programming (OTAP) concept was introduced to WSNs, updating a sensor node firmware on site turned out to be outdated and not efficient. This technology is used in the I-RAMP³ project not to update or reconfigure a sensor node firmware, but to program from scratch a new one connected to the network with no measurement capabilities what so ever. The goal is to force the network to configure the new sensor node with sensing capabilities according to the task needed to be performed at the moment on the system.

The paper is organized in five different Sections. Section II talks about the latest advances of WSNs in the European industry, mostly implemented on the I-RAMP³ project. Section III depicts the Sensor Data Validation techniques used in the present work, together with a quick overview about the latest applications of WSNs in various scenarios. Section IV presents the developments of OTAP and its detailed process in the I-RAMP³. Then, an open discussion about the benefits of a NETDEV-like approach and all its embedded functionalities taking into consideration the end users of the system is presented in Section V. Ultimately, the present paper ends up with an acknowledgement and final remarks about the developed technology and its significant importance as a next step for intelligent manufacturing.

II. WIRELESS SENSOR NETWORK IN INDUSTRY

Sensor usage on industrial applications has become extremely important, since monitoring the behavior of a machine is crucial to adapt its operation due to regular

changes on product demand. On a shop-floor environment, sensors should not be treated as an integrant part of a machine, but a separated component, which like complex machines, should be flexible enough to change its operations according to process demands. In I-RAMP³ were explored new concepts on WSNs applied in industry, aiming for the addition of an intelligence layer on sensors, which empower them to be as complex as machines, both sharing plug'n'produce features and both capable of communicating with each other on an agent-like system environment.

Intelligent WSNs rely on some features such as easy integration of sensor nodes from different manufactures using, e.g., the PlugThings Framework [1] technology, along with automatic calculation of the nodes' physical location, self-diagnosis capabilities using sensor data validation methods, and self-reconfiguration capabilities using OTAP technologies to reprogram new sensor nodes on the network.

A. Sensor Integration

In the I-RAMP³ project, the integration of multiple types of sensor nodes on the system is made using the PlugThings Framework, which contains a *Universal Gateway* (UG) to parse raw sensor data from the different sensor nodes. As can be seen in Figure 1, each sensor node of the network communicates directly to this gateway node, where the received measurements are processed and translated from raw data (stream of bytes) into readable form (measurement values). These data are compiled on Extensible Markup Language (XML) based format files that are part of the *Sensor & Actuator Abstraction Language* (SAAL), which is used to communicate with *Sensor & Actuator Abstraction Middleware* (SAAM), where all the intelligence related to the sensors is implemented. When the SAAM receives a new message from a sensor node, it will collect the sensor board identification number (ID) and the Media Access Control (MAC) Address that identifies the communication protocol. Both board ID and MAC Address are the unique identifier of a sensor node.

Joining a new sensor node to the network will imply the creation of a new S&A NETDEV corresponding to that sensor node, letting transparent to all the entities on the network what measuring tasks it can perform. Since a sensor node can have multiple sensors integrated, the corresponding S&A NETDEV will be able to perform different tasks related with the different sensor types of the sensor node. It will have one task per sensor integrated in the mote, being this way able to provide sensor information in a standardized way.

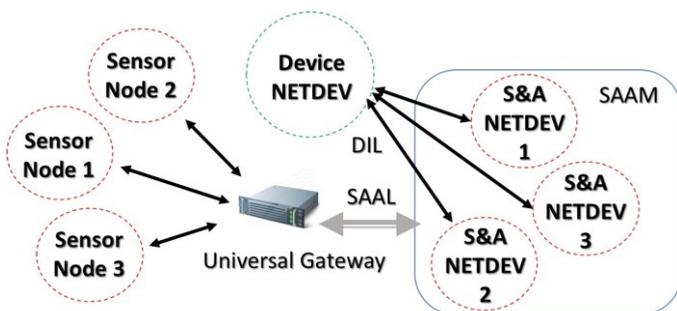


Figure 1. Sensor Integration on an I-RAMP³ Environment

B. Device NETDEV and S&A NETDEV

NETDEV communication is accomplished by using a task-driven language - Device Integration Language (DIL) - which is composed by four main XML schemas: NETDEV Self-Description (NSD), which describes the capabilities that a specific NETDEV can perform, by defining conditions, goals and process parameters; Task Description Document (TDD), which is a request for task execution, specifying the conditions, goals, process parameters and the period of the task execution; Quality Result Document (QRD), which is the result of a task iteration, detailing the process quality; Task Fulfillment Document (TFD), which is an acknowledge document that represents the task finalization. A task request process is represented in Figure 2.

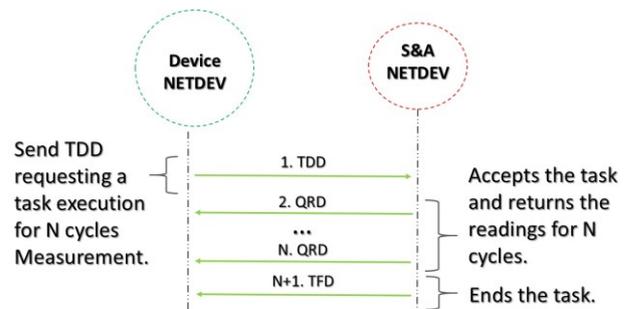


Figure 2. DIL Communication

This communication is initiated every time a NETDEV depends on one another to execute a task. Generally, S&A NETDEVs are requested by Device NETDEVs via TDD to execute tasks for sensing the environmental conditions during a given number of cycles. If the S&A NETDEV is able to execute the requested task, it will successfully acknowledge the request by answering with QRDs containing the sensor data, during the number of cycles specified. A TFD will be sent back to the Device NETDEV denying the task if the S&A NETDEV is not capable of meeting the task goals and conditions or if it is already occupied executing a task for other NETDEV entity. A TFD is also used to acknowledge a successful task execution finalization.

III. SENSOR DATA VALIDATION

Sensors are used at the shop-floor level to monitor the surrounding environmental and/or physical conditions of machines and all manufacturing components. The measured data will be used as an input for complex machines to control the manufacturing process and to adapt themselves according to these external conditions. This adaptation allows the machine to be flexible enough to change its variable inputs and internal processing, controlling the production process to maintain product quality despite fluctuations. Machine's process depends on data measured from sensors, so it's very important that these data stays the most reliable as possible when delivered to the machine. Data samples collected from sensors, especially from WSNs, are prone to be faulty due to internal and external influences, such as environmental effects,

limitations of resources, energy problems, hardware malfunctions, software problems, network issues, among others, as shown in [18]-[21]. Sensor data validation consists on a set of methods applied to the data provided by the sensors with the main goal of detecting anomalies and malfunctions on these sensors and take action accordingly on the corresponding S&A NETDEVs.

A. Methodology

Data validation methods are applied to data received from sensors. Finding deviations from normal sensor readings doesn't mean that they occur due to a malfunction of the sensor node, but rather due to an abnormal variation of conditions being measured. Despite being a sensor-based cause or a conditions-based cause, the WSN is self-aware and self-diagnosis of the task execution's process state.

Anomaly detection methods generally classify data into correct or faulty. There is no right method that works better than all the others and no method guarantees success, because they all depend on several factors such as type of monitored variable, the overall measurement conditions, the sensor used and the characteristics of the environment being perceived [2][15]. In [2][3] is proven that anomaly detection should not rely on just one method, but instead on a number of methods applied successively for detecting different types of data faults. Furthermore, there are methods [2] suitable to be used online, and other more complex and demanding on the processing level, suitable for offline validation, such as Bayesian Networks (BNs), Artificial Neural Networks (ANNs), Regression Techniques like Partial-Least Squares Regression, etc., used in many different contexts such as aerospace, energy, electric power systems, urban environment, among others [8]-[14]. Regarding S&A NETDEVs, techniques that provide a quick WSN diagnostics were used, such as Min/Max, Flat Line [3][5], Modified Z-Score [7] and No Value detection.

The Min/Max approach is based on a heuristic rule, which defines upper and lower bounds that refer to hardware specifications or/and conditions that are not likely to occur in the current context. Therefore, if sensed data is within bounds, data are likely good, otherwise, the sensor may be faulty. The Flat Line technique is based on temporal correlation of a big chunk of latest data collect. If the difference between successive data samples remains zero, this means that the sensor is probably faulty. Modified Z-Score is a statistical-based technique used as an outlier detection mechanism. It takes into account averaged values and deviations to assess if a certain value do not follows the same behavioral trend as the others. The No Value detection technique finds gaps in datasets. If the difference between the current time and the timestamp of the last measurement is unusually large, then probably the sensor has stopped the communication with the gateway.

B. Implementation

On I-RAMP³, the sensor data validation is characterized by four main steps, as shown in Figure 3: 1) First, raw data is acquired from the sensor nodes; 2) Raw data is converted into

a readable form by the UG and sent to the SAAM; 3) While a S&A NETDEV executes a task, the received sensor data is validated by a sequence of internal methods to detect anomalies; 4) If anomalies on data are detected, the corresponding S&A NETDEV is marked as probably faulty, which results, depending on the severity of the error detected, in the inability of accepting future task executions or termination of the current task's execution.

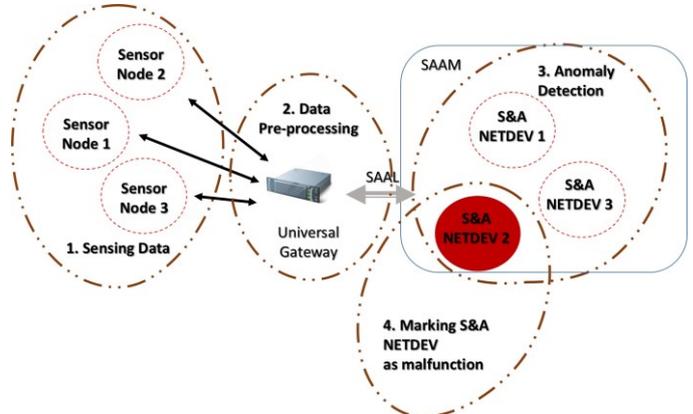


Figure 3. Data Validation approach on I-RAMP³

While the S&A NETDEV is executing a task, the dataset of the corresponding sensor node will go through two validation modules: Module A, which is intended for detecting sensor malfunctions and Module B, which is intended for detecting abnormal behavior from the sensor node.

Module A validates the received sensor data using Flat Line [3][5][17] and No Value detection methods, aiming to identify a malfunction sensor node. If Flat Line method returns positive for error detection, it means that, on the sensor node, the board is reading the same electrical quantity for an unusual amount of time, which means that the sensor doesn't detect any variation on the environment quantity being measured. Hence, it's most likely that the sensor is not correctly connected to the board. On the other hand, if the No Value method detects gaps in the dataset, most likely the battery as run out or the sensor node just broke down. Facing a malfunctioning sensor node, the corresponding S&A NETDEV is responsible to terminate prematurely the task execution, without any human interaction and making itself unavailable to take on other task requests. Module B is intended for methods that detect outliers, such as the Min/Max detection [3][5][16], which detects readings out of system limit thresholds, and the Modified Z-Score [4][5] that detects spikes and abnormal readings. This module returns a strong probability about the malfunctioning state of the sensor, despite lower than the one returned by Module A. This probability is based on the defective readings that, in this case, can be caused by sensor failing or abnormal behavior of the system itself. In such circumstances, the S&A NETDEV waits for the normal task termination to change its process state to unavailable (for future task executions), while a maintenance process doesn't occur on the corresponding sensor node.

IV. OVER THE AIR PROGRAMMING

OTAP is a technology developed originally to update firmware for mobile devices. Since the use of this type of equipment rely greatly on wireless internet access, OTAP has been used on the past years from manufactures and network operators to deliver firmware updates to equipment with internet access. However, because of the widely use of WSNs and the growing complexity of them, OTAP was taken to a new direction towards WSNs [22].

A WSN could have thousands of sensor nodes and the maintenance of these nodes could be very time-consuming. Therefore, since they must all be re-programmed one by one, this is not a very cost-effective solution. Moreover, the WSN may have nodes located in difficult access places, so updating firmware in sensor nodes on site can be challenging. Several sensor nodes from different manufactures are already embedded with the OTAP technology, which relies on updating firmware on sensor nodes from the gateway node, using the existing wireless communication between them, such as XBee, Wi-Fi or 3G.

A. OTAP Methodology in I-RAMP³

The WSN consists on different sensor nodes, gateway nodes connected to the UG and the communication topologies between them. The Sensor Nodes used in the I-RAMP³ that make the OTAP implementations possible are the Libelium Waspote PRO (v1.2) [23] sensor boards with the XBee module for the 802.15.4 communication protocol [24]. Updating firmware on the Waspote PRO (v1.2) requires using the Libelium OTA technology [6], which divides the OTAP process on two main steps: 1) Node discovery on the network and 2) Firmware upload. The OTA-Shell application [6] is used at the UG level to control the options available in OTA, sending commands to the sensor nodes to be reprogrammed. A firmware upload occurs when the shop floor operator replaces sensor node hardware due to a severe malfunction detection on a sensor node (using the methods discussed previously). The logical representation of OTAP methodology is depicted in Figure 4.

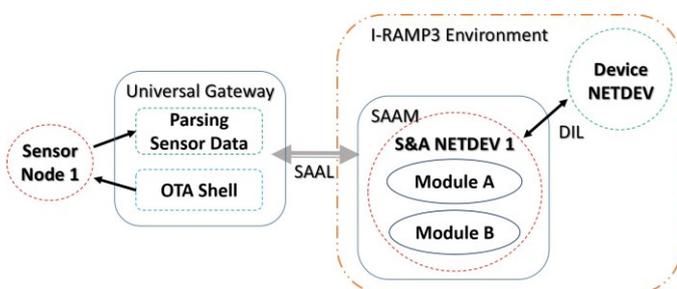


Figure 4. OTAP Methodology on I-RAMP³

When a S&A NETDEV is executing a task and a sensor node failure is detected, the malfunction could be caused by irreversible problems that require equipment replacement on the nodes, such as: 1) Replacement of the bad sensor/communication module; 2) Replacement of a bad sensor board; 3) Replacement of the entire sensor node. Since

sensors and communication protocols are not directly related with the program that is running on the sensor node, 1) doesn't require firmware update of any kind. On the other hand, when 2) or 3) occurs, a firmware update is required, which can be done traditionally or using the OTAP approach.

Traditionally, before a new sensor board is connected, it needs previously to be manually programmed with the right program. This approach may be counterproductive on a smart factory context, since the ramp-up time of replacing a sensor board could be very high. With the OTAP approach, when a new sensor board is connected, the sensor node is informed from the network of what to do, by being programmed automatically over the air. The basic idea is to previous store on the UG the replaced sensor node's program in form of an automatic generated binary image after compiling the code and program the new sensor node over the air with the stored binary image, replacing a malfunction one.

1) Replacement of Faulty Sensor/Communication Module

Malfunctions on the sensor node may have its root cause on specific components of the node, leading to the replacement of only the bad component. A malfunctioning S&A NETDEV detected by, e.g., a Flat Line could be possibly caused by a broken sensor that was used on the task execution requested and, therefore, the replacement process requires only the exchange of one sensor. On the other hand, if the malfunction is detected by, e.g., a No Value method, probably it is caused by problems on the communication protocol. The S&A NETDEV shuts down temporarily, until the component exchange is finished.

The moment a sensor or communication protocol replacement occurs and the sensor node is turned on, the S&A NETDEV will detect incoming readings from the same sensor board, as it used to, and associates this sensor node to the same S&A NETDEV making it available for task execution once again. If the communication protocol was replaced, the MAC Address associated with the S&A NETDEV is updated by the new one.

2) Replacement of Sensor Board

In the I-RAMP³ context, OTA is applied not for firmware update but for programming a new sensor board for the first time it joins the network, after replacing a failing sensor node. The process begins the moment a malfunction sensor node is detected during task execution, which imply replacing a failing sensor board, without exchanging the components connected to it, such as sensors and communication protocol. With the OTA approach, the shop floor operator avoids programming manually the new sensor board before it is connected to the system. The sensor board runs a program that sends to the gateway specific messages, meaning it is "Alive" and lacks contextual information, and waits to receive instructions for an OTA process. This "Alive" message is a defined string, containing information about the new sensor node, such as sensor board ID and MAC Address. Because only the sensor board is replaced, the MAC Address received on the "Alive" messages was already associated with an

existing S&A NETDEV, so the corresponding sensor ID will be updated by the new one. This means that the sensor board changes, but the virtual representation (S&A NETDEV) of the sensor node remains the same.

Since “*Alive*” messages are received instead of sensor readings and the sensor node is associated with an existing S&A NETDEV (due to the MAC Address), an OTAP process begins. First, SAAM identifies which program is the right one to be used for sensor programming via OTA, based on the previously created S&A NETDEV capabilities. Hence, SAAM orders the UG to start a new instance of the OTA Shell, using the identified binary image to program that specific sensor node. The UG runs the OTA Shell, which first scans the network to locate the new node to be programmed and sends the binary file to the identified node, which stores the file on the Secure Digital (SD) card. The sensor node reboots in order to start the execution of the new program, after receiving the program successfully. The program is copied from the SD card to the Flash Memory and the sensor node starts running the new binary file.

After restoring its configuration, the sensor node is ready to operate again, starting to measure and sending data to the corresponding S&A NETDEV, which changes its internal state becoming available for task execution.

3) Replace the Entire Sensor Node

The malfunctions detected may be severe to the point where none of the component on the sensor node can be saved, forcing the replacement of the entire node. When this happens, a new sensor node is connected to the network, which has a sensor node ID and MAC Address that are new in the system, resulting on the creation of a new S&A NETDEV, available to take requests for task execution.

The only way SAAM knows which tasks the new S&A NETDEV is able to perform, is by parsing the messages received from the sensor node and detect which are the sensor types connected to it. This occurs if the new sensor node is already programmed with the right program for the task pretended. On the other hand, if “*Alive*” messages are received, SAAM can't possibly know which tasks the new sensor node is able to perform, because it doesn't have any sensor readings and no background to associate the sensor node to an existing S&A NETDEV with capabilities already identified.

V. DISCUSSION

The use of sensors in the industrial domain for condition-based monitoring and machine's parameterization always was a key element in the industrial domain. In most recent technological trends in manufacturing, machines need to act and adapt according to the environmental conditions to perform its tasks as reliable, effective and efficient as possible. Sensors generally assess not only the machine's condition by means of, e.g., temperature – essential to monitor the temperature of motors used in Linear Axis; Humidity – in sealing applications, the skin formation is driven by several parameters, and one of them is humidity; Luminosity – when

using an optical sensor, most of the times is peremptory to calibrate the exposure time of the device in order to maximize the quality of image acquisition. Therefore, one of the cornerstones of I-RAMP³ is to explore the applicability of WSNs and all the benefits it can bring to manufacturing environments, shielded with innovative concepts as NETDEVs enabling peer-to-peer device communication, and also plug'n'produce that shortens the time of device readiness to use.

Based on fact the WSN can take part on the intelligent manufacturing systems, two main functionalities were explored in this work. As previously explained, the use of reliable WSN compels the use of Sensor Data Validation techniques to assess the sensor functioning conditions and diagnose when there's a sensor breakdown for rapid responsiveness of the maintenance personnel, and consequently shorten the production system's down-time. As a way of shorten the ramp-up time after a sensor breakdown, the use of OTAP is explored to automatically program a mote becoming ready to use in a matter of seconds.

Assuming a perspective of a shop-floor Operator or Maintenance Engineer, the use of NETDEV entities like S&A NETDEV for sensor virtual encapsulation reveals to bring many benefits to the manufacturing environment. Since there are many dependencies from machine's execution and sensor readings, if there's not available an online functionality to permanently assess the reliability of sensors in case of faulty data, it can lead to machine damage and even put at risk the safety of shop-floor personnel. The use of NETDEV entities as a shell on the shop-floor devices can avoid this situations and also improve the knowledge about the process life-cycle. Moreover, in terms of System Design, since all these sensor validation techniques are embedded on the S&A NETDEV, it avoids the technical personnel to know in detail and implement the used techniques for sensor validation. In terms of integration, as previously explained, NETDEVs have a standardized way of communication, so easily any tool or software solution can interact with this system.

On the other hand, when a sensor is faulty, there's always the need to change the sensor for a new one, or schedule a downtime for sensor maintenance purposes. In the case of sensor exchange, being the most used practice due to cheap cost of motes when compared to machine, OTAP is a flexible and quick way of reverberate it. In terms of flushing a mote with the correspondent code for execution, the OTAP approach used in I-RAMP³ is a totally automatic process that needs no physical direct interaction with the computer, which is most of the times made using a Universal Serial Bus (USB) cable, and not knowing which program is needed for flushing. On a shop-floor Operator's perspective, the only thing needed to perform sensor node programming is knowing which one is faulty, remove the mote, put a new one in the same position and switch it on. As described in Section IV, the process of mote identification in the network and program to be used to flush the mote is totally automatic. This way, only the physical mote removal and addition is necessary.

Additionally, and out of I-RAMP³ bounds but still worth to highlight, is OTAP use to easily exchange the functionality of a sensor node with limited physical access. An example of that is changing the acquisition frequency of environmental conditions due to changes on customer requirement, having to a direct impact on the process. Another example is to automatically reprogram a mote to interpret different sensors physically connected with the sensor board. Once again, no manual flushing is needed and only the physical exchange of sensors is necessary.

VI. CONCLUSIONS AND FUTURE WORK

Automation and responsiveness always made part of the industrial mindset, where more flexibility and reliability means better production, better production means more customer confidence and more customer confidence means more competitive advantage toward more income.

I-RAMP³ is a key enabler for this kind of philosophy since it aims to convert a production environment into an agent-like system, and therefore, to allow for machine-to-machine communication, self-aware capabilities due to NETDEV encapsulation becoming self-diagnosable and self-reconfigurable with all the functionalities developed so far in the project. The NETDEV concept, together with online and automatic Sensor Data Validation, and along with OTAP, are the main foundations that can turn the use of WSN in the industrial domain into a reality, paving the way for next generation of Factories of the Future. Based on all the aforementioned topics, we can state that the use of sensors still have an important evolution to take place in industry, and all the technological baseline is being yield.

ACKNOWLEDGMENT

This research was supported by project I-RAMP³ (FoF.NMP.2012-3) – Intelligent Network Devices for fast Ramp-up – funded by the European Commission under the Seventh Framework Programme for Research and Technological Development.

REFERENCES

- [1] G. Gonçalves, J. Reis, R. Pinto, M. Alves, and J. Correia "A step forward on Intelligent Factories: A Smart Sensor-oriented approach." *Emerging Technology and Factory Automation (ETFA)*, 2014 IEEE. IEEE, 2014. pp. 1-8.
- [2] N. Branisavljevic, Z. Kapelan, and D. Prodanovic "Improved real-time data anomaly detection using context classification." *Journal of Hydroinformatics*, 13(3), 2011, pp. 307-323.
- [3] A. B. Sharma, L. Golubchik, and R. Govindan "Sensor faults: Detection methods and prevalence in real-world datasets." *ACM Transactions on Sensor Networks (TOSN)*, 6(3), 2010, p. 23.
- [4] L. S. Jayashree, S. Arumugam, and A. R. Meenakshi "A communication-efficient framework for outlier-free data reporting in data-gathering sensor networks." *International Journal of Network Management*, 18(5), 2008, pp. 437-445.
- [5] J. Ravichandran and A. I. Arulappan "Data Validation Algorithm for Wireless Sensor Networks." *International Journal of Distributed Sensor Networks* 2013 (2013).
- [6] Libelium. [Online]. Available: http://www.libelium.com/uploads/2013/02/over_the_air_programming.pdf. [retrieved: September, 2015]
- [7] B. Iglewicz and D. Hoaglin "How to detect and handle outliers", *The ASQC Basic References in Quality Control: Statistical Technique 16*. Milwaukee, WI: American Society for Quality Control, 1993.
- [8] O. J. Mengshoel, et al. "Probabilistic Model-Based Diagnosis: An Electrical Power System Case Study", *Systems, Man and Cybernetics, Part A: Systems and Humans*, IEEE Transaction on, 40(5), 2010, pp. 874-885.
- [9] O. J. Mengshoel, A. Darwiche, and S. Uckun "Sensor Validation using Bayesian Networks." In *Proc. of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS-08)*, Los Angeles, CA, 2008.
- [10] C. Jianhong, L. Hongkun, S. Deren, and L. Wei "A hybrid data-driven modeling method on sensor condition monitoring and fault diagnosis for power plants", *International Journal of Electrical Power & Energy Systems*, 71, 2015, pp. 274-284.
- [11] A. Messai, A. Mellit, I. Abdellani, and A. Massi Pavan "On-line fault detection of a fuel rod temperature measurement sensor in a nuclear reactor core using ANNs", *Progress in Nuclear Energy*, 79, 2015, pp. 8-21.
- [12] S. Dauwe, et al. "Multi-criteria anomaly detection in urban noise sensor networks", *Environ. Sci.: Processes Impacts*, 16(10), 2014, pp. 2249-2258.
- [13] K. Zhang, S. Shi, H. Gao, J. Li "Unsupervised Outlier Detection in Sensor Networks Using Aggregation Tree", *Advanced Data Mining and Applications*, 4632, 2007, pp. 158-169.
- [14] C. C. Castello, J. Sanyal, J. S. Rossiter, Z. P. Hensley, J. R. New "Sensor Data Management, Validation, Correction, and Provenance for Building Technologies." Technical paper TRNS-00223-2013.R1. In *Proceedings of the ASHRAE Annual Conference and ASHRAE Transactions 2014*, Seattle, WA, June 28-July 2, 2014.
- [15] J. Bertrand-Krajewski, J. Bardin, M. Mourad, and Y. Branger "Accounting for sensor calibration, concentration heterogeneity, measurement and sampling uncertainties in monitoring urban drainage systems." *Water Science & Technology*, 47(2), 2003, pp. 95-102.
- [16] M. Mourad and J. L. Bertrand-Krajewski "A method for automatic validation of long time series of data in urban hydrology." *Water Science & Technology*, 45(4-5), 2002, pp. 263-270.
- [17] G. Olsson, M. Nielsen, Z. Yuan, A. Lynggaard-Jensen, and J. P. Steyer "Instrumentation, control and automation in wastewater systems." (2005).
- [18] G. Tolle, et al. "A macroscope in the redwoods." *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 51-63.
- [19] G. Barrenetxea, et al. "Sensorscope: Out-of-the-box environmental monitoring." *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*. IEEE, 2008, pp. 332-342.
- [20] N. Ramanathan, et al. "Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks." *Center for Embedded Network Sensing* (2006).
- [21] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler "An analysis of a large scale habitat monitoring application." *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 214-226.
- [22] I. Adly, H.F. Ragai, A. El-Hennawy, and K. A. Shehata "Over-The-Air Programming of PSoC sensor interface in wireless sensor networks." *MELECON 2010 - 15th IEEE Mediterranean Electrotechnical Conference*. IEEE, 2010, pp. 997-1002.
- [23] Libelium. [Online]. Available: http://www.libelium.com/uploads/2013/02/waspnote-technical_guide_eng.pdf. [retrieved: September, 2015]
- [24] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards." *Computer communications*, 2007, 30(7), pp. 1655-169.

Comparing Knowledge Representation Forms in Empirical Model Building

Hao Wang

Leiden Institute of Advanced
Computer Science
Leiden University
Leiden, The Netherlands
e-mail: h.wang@liacs.leidenuniv.nl

Ingo Schwab

Karlsruhe University of Applied
Sciences
Karlsruhe, Germany
e-mail:
Ingo.Schwab@HS-Karlsruhe.de

Michael Emmerich

Leiden Institute of Advanced
Computer Science
Leiden University
Leiden, The Netherlands
e-mail:
m.t.m.emmerich@liacs.leidenuniv.nl

Abstract—Empirical models in engineering practice often come from measurements of the machines but might also be generated from expensive simulations to build so-called surrogate models. From an abstract point of view can be seen as approximations of functions that map input variables to output variables. This paper describes and conceptually compares different function approximation techniques, with a focus on methods from machine learning, including Kriging Models, Gaussian Processes, Artificial Neural Networks, Radial Basis Functions, Random Forests, Functional Regression, and Symbolic Regression. These methods are compared on basis of different criteria, such as speed, number and type of parameters, uncertainty assessment, interpretability, and smoothness properties. Besides, a particular focus is to compare the different ways of how knowledge is represented in these models. Here we compare the families of functions used to build the model and which model components (structures, parameters) are provided by the user or learned from the available data. Although this paper is not about benchmarking, some numerical examples are provided that illustrate the typical behavior of the methods.

Keywords—*Function Approximation; Machine learning; Process Modelling; Model Formation; Symbolic Regression*

I. INTRODUCTION

Contemporary engineering design is heavily based on computer simulations or work piece experiments. The results are used not only for design verification but, even more importantly, to adjust parameters of the system to have it meet given performance requirements. Unfortunately, accurate simulations or real life experiments with machines or work pieces are often very expensive. High-fidelity simulations can take with evaluation times as long as hours or even days per design, making design automation using conventional methods impractical. These and other problems can be alleviated by the development and employment of so-called surrogates that reliably represent the expensive, simulation- or experiment based model of the system or device of interest. They are often analytically tractable and can give answers in a much faster or less costly way. Once the model is set up, the parameter values which produce the desired output from the given input can be retrieved fully automatically and without any delay. More importantly, the models can inter- or extrapolate values and thereby predict

response values for new input parameters. In this way empirical model building is a regression task. Methods already exist, but it is still subject to intense research. In particular advances were made in the field of machine learning where by the use of computers significantly more complex models can be studied than it was possible before the ‘digital age’. This paper gives an overview over machine learning methods and classical methods that can be used in empirical model building. The goal is to compare them on basis of a wide range of mainly conceptual (qualitative) properties. In particular, we study how knowledge is represented with these methods and which knowledge can be introduced by the user or has to be learned by the method. We deliberately avoided an evaluation of the different algorithms on benchmark problems, as we think that such a benchmark is beyond the scope of a workshop paper as it would involve proper tuning/setting of model parameters in order to provide a fair comparison. Instead, we provide numerical examples in order to demonstrate typical behavior of certain model types.

This paper is organized as follows. In Section II, the most commonly exploited approximation models are briefly introduced, illustrating their corresponding capabilities and limitations. In addition, some intrinsic properties e.g., model uncertainty assessment are discussed in detail. In Section III, a numerical comparison is made through modelling the data sampled from a 2-D Rastrigin function. In Section IV, we compare the models in terms of their design principles as well as their underlying mathematical structures and parameters. Finally, Section V concludes the paper and points out the further researches beyond this work.

II. BACKGROUND AND RELATED WORK

In this section we first briefly describe the most commonly used function approximation models. Then, some important properties of the models: symbolic representation, model uncertainty assessment and universal function approximation capability are discussed.

A. Function Approximation Models

1) Kriging

Kriging is a stochastic interpolation/regression approach, which originates from earth science [1] and originally targets at problems in geo-statistics and mining. Note, that the

Kriging method is also termed as Gaussian Process Regression [2] in the statistical machine learning literature, although the latter is restricted to normal distributions and typically provides a Bayesian interpretation of the predictor. Kriging assumes that the observed input-output data is the realization of a random field, and based on this assumption estimates correlation parameters and then computes the best linear unbiased estimator to predict the output value for a given set of input variables. It offers a local assessment of the prediction uncertainty, known as the Kriging variance at any unobserved data point.

The Kriging method interpolates the output at unknown data sample by modeling the response values as a realization of a random process y , which is a sum of a function $\mu(\cdot)$ and a centered Gaussian random field ε [2].

$$y(x) = \mu(x) + \varepsilon(x) \quad (1)$$

Despite the fact that a spatial index is used (from \mathbb{R}^n space) it is common in the literature to call the random field also "Gaussian Process" for the multi-dimension case. Moreover, unlike Gaussian Process Regression, the Kriging framework can also be used for non-Gaussian random fields.

The centered Gaussian random field ε is completely defined by specifying the covariance function $k(\cdot, \cdot)$, that solely depends on the relation between the input vectors.

$$k(x, x') = Cov[\varepsilon(x), \varepsilon(x')] = E[\varepsilon(x)\varepsilon(x')] \quad (2)$$

The user has to choose the structure of the covariance function, which must be positive definite. It normally depends on the similarity between the inputs, the parameters of the covariance function are learned from the available data, e.g., by maximum likelihood estimation.

When the function $\mu(\cdot)$ is assumed to be constant and unknown, the method is called Ordinary Kriging (OK). If $\mu(\cdot)$ is a functional regression model it is called Universal Kriging. In OK, the predictions are made based on the posterior distribution conditioning on the training set (X, y) , which is shown in (1). (here: $\mathcal{N}(m, s^2)$ denotes the normal distribution with mean m and variance s^2)

$$y^t | X, y, x^t \sim \mathcal{N}(m(x^t), s^2(x^t)) \quad (3)$$

Where x^t is target (unknown) input to make the prediction. The posterior mean function is the estimator and the variance can be used to compute uncertainty margins. For details, refer to [2].

2) Support Vector Machines

Support Vector Machines (SVMs) [3] [4] are supervised learning algorithms which are originally designed for classification tasks. The original SVM algorithm was introduced in Computational Learning Theory conference (COLT-92) by Boser, Guyon, Vapnik [4]. It is now well-established in machine learning. SVMs target at optimally solving non-linear classification tasks, using two techniques

to achieve the goal: maximum-margin separation and kernel functions.

The simplest form, linear SVM construct a hyper plane or set of hyper planes in a high-dimensional space. Linear SVM can be considered as a perceptron but improve it by performing an optimal classification. Informally, a binary classification is achieved by finding two hyper planes. Both of them separate the data while the distance between them is maximized. The region bounded by these two hyper planes is called the margin. The motivation behind this approach is, in general, the larger the margin the lower the generalization error of the classifier. The corresponding linear classifier is known as the maximum margin classifier. It is also equivalent to the perceptron of optimal stability.

In order to extend the capability of linear SVMs, the well-known kernel trick [5] is applied to implicitly map the samples to be separated to the (normally) high-dimension feature space. This allows for a linear maximum-margin separation in the feature space even if the problem is not linearly separable in the input space.

3) Neural Networks

Artificial neural networks (ANNs) [6] [7] [8] is a broad family of statistical learning algorithms inspired by biological neural networks, whose structure is presented as systems of interconnected artificial neurons. The first artificial neuron, a simple linear classifier, was proposed in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pitts [8]. Each artificial neuron is modeled as a transformation of a linear combination of its input. Given the input to a neuron is x (not necessarily the input to the whole network), the output O from an arbitrary neuron is:

$$O = \varphi \left(\sum_{i=1}^n w_i \cdot x_i \right) \quad (4)$$

Note that w_i is the weight for the linear combination while $\varphi(\cdot)$ presents the transformation, or called activation function more commonly. There are various activation functions that have been applied. For example, linear, Heaviside step function, sigmoid and hyperbolic tangent.

There are many types of ANNs. If there are no cycles (including self-connections) in the network, then we could divide the network into layers, which is termed as feed forward neural network [9]. This type of ANNs are built by choosing the number of layers and the class of activation function for each neuron. Many well-known neural networks belong to this category, for example, multi-layer perceptron and radial basis function networks. If cycles or self-connections exist in the network, it is generally called recurrent neural network [10].

The feed forward neural network is of great interest here because any real-valued continuous function can be approximated arbitrarily close by a multi-layer perceptron with just one hidden layer according to the universal approximation theorem [11] [12] (see subchapter D of this

article). Therefore, the feed forward network design is expected as good approximation method. However, due to the curse of dimensionality [13], it will be unavoidable that, even for smooth functions, the amount of data needed for precise approximation tends to grow exponentially with the dimensionality, unless more precise assumptions on the model class are made.

A variety of training techniques have been employed and tested. Among them, the back-propagation technique [14] is the most popular and influential. The general idea is to express the training error (loss) function as the squared difference between one true target and the approximated value, which is differentiable with respect to the weights. Consequently, we could compute the partial derivative of the error to each weight so that some efficient optimization techniques based on gradients can be applied.

4) Radial Basis Function Networks

Radial basis functions (RBFs) [15] are a special class of real-valued functions whose value only depends on the distance to the central point. In another word, any function h having the property $h(x) = h(|x|)$ is a radial basis function. A classical and commonly used type is the Gaussian:

$$h(x) = \exp\left(-\frac{||x - c||}{r^2}\right), \quad (5)$$

where c is the center and r^2 is its radius. Many other types are also available, e.g., multi-quadric. Please refer to [15] for more details. Radial basis functions are typically used for function approximations, in which multiple RBFs are linearly weighted as the predictor:

$$\hat{y}(x) = \sum_{i=1}^m w_i h(|x - c_i|) \quad (6)$$

Note that m is the number of the RBFs, each associated with a different central point c_i and a combination weight w_i . From the neural network perspective, the RBFs approximation method can be viewed as an artificial neural network, using radial basis function as activation function and thus is also called radial basis function network (RBFN) [16]. The RBFN is normally trained in two steps. Firstly, the central points of the RBF functions are chosen in an unsupervised manner. Usually, random sampling or k-mean clustering is used for this step. Secondly, note that the predictor $\hat{y}(\cdot)$ is differentiable with respect to the weights. Therefore, any well-established learning method involving derivatives can be employed to train the RBFN [17], e.g., using the back-propagation method [14] to find optimal weights.

RBFN is also closely related to the Kriging method we have described earlier. Giannakoglou [18] introduced an approach on how to employ RBF networks for exact

interpolation in the sense that results for points in the training set are reproduced exactly. This kind of RBFN leads to the same equations as they are used in the prediction step of Simple Kriging [19].

5) Polynomials and Splines

As an extension to the linear model, polynomial regression models the relation between the response variable and the input variables as a polynomial, which represents a non-linear underlying assumption/knowledge. Therefore, it has been widely applied on the data involving nonlinear phenomena. The polynomial maps the input in R^n to high dimensional space R^m ($m > n$) by introducing interactions/correlations between input components. The free parameters (to estimate) in the polynomial regression is the coefficient of the polynomial plus the degree of the polynomial. Although polynomial regression models a non-linear relation, it is still linear to the unknown coefficients and the transformed input in R^m .

Second order polynomial functions are often used in real live scenarios. However, the use of polynomials as a global approximation only makes sense if the initial landscape, is unimodal, which is often the case in engineering problems.

The coefficients are usually determined by the (ordinary) least squares estimation method, which minimizes the variance of the estimators of the coefficients. The degree of the polynomial is usually specified by the user.

As an extension to simple polynomial functions, Splines [21], which are piecewise polynomials defined on disjoint domain partitions, are usually preferred for approximation purposes due to their ability of avoiding the instability from Runge's phenomenon [22] in high degree polynomial fitting. As for univariate regression of order n (the highest order of all the polynomials), each piecewise polynomial defined on disjoint interval is chosen such that the derivatives of each pair of connecting polynomial at the connection point should be equivalent up to order $n - 1$, which leads to the continuous and continuously differentiable of the whole spline curve.

The mostly applied splines are B-splines ('B' stands for basis), particularly cubic B-splines. The importance of B-splines is mainly due to the fact that any spline curve can be represented as a weighted linear combination of B-Splines. For each piecewise polynomial, the B-splines up to a certain order can be evaluated recursively by De Boor's algorithm [23]. Using B-splines, a function approximation task can be achieved by fitting a spline function composed of a weighted sum of B-splines, using the least-squares method.

6) Random Forests

Ensemble learning algorithms [26] [27] construct a set of classifiers and then label new data sets by taking a (weighted) vote of their predictions, e.g., random forest [28]. The main principle behind ensemble methods is that the prediction quality increases when a set of learning algorithms are grouped together. There are different types of algorithms for ensemble learning, i.e., boosting [29], bootstrap aggregation [30] or stacking [31].

Random forests are also a type of ensemble learning. They combine different decision tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The calculation of the resulting class (classification problem) or regression output is the most straightforward. For classification the majority of the class output of the decision trees are calculated. For a regression problem accordingly the mean of the trees are the result of the model.

The advantages of random forest is mostly its fast running speed, both in training and prediction/classification and easy interpretability. In addition, they are able to deal with unbalanced and missing data, which is quite common in the real application. Its weaknesses are that when used for regression they cannot predict beyond the range in the training data, and they may over-fit data sets that are particularly noisy.

7) Symbolic Regression

In the previous subsection, we assumed that we already know the structure of the output. After that we choose the best simplified model and fit the free parameters of the model. From this point of view Symbolic Regression is much more powerful. In this case, the function is composed of an arbitrary (but predefined) set of mathematical symbols, forming a valid expression of a parameterized function. Like other statistical and machine learning regression techniques symbolic regression also tries to fit observed experimental data. But unlike the well-known regression techniques in statistics and machine learning, symbolic regression is used to identify an analytical mathematical description and it has more degrees of freedom in building it.

A set of (basic) operators is predetermined (e.g., add, multiply, sin, cos) and the algorithm is mostly free in concatenating them. In contrast to the classical regression approaches which optimize the parameters of a prescribed structure, here the structure of the function is free and the algorithm optimizes the parameters and the structure.

There are different ways to represent the solutions in symbolic regression. For example, informal and formal grammars have been used in genetic programming to enhance the representation and the efficiency of a number of applications including symbolic regression.

Since symbolic regression operates on discrete representations of mathematical formulas, non-standard optimization methods are needed to fit the data. The main idea of the algorithm is to focus on searching promising areas of the target space while abandoning unpromising solutions (see [24] [25] for more details). In order to achieve this, symbolic regression uses the main mechanisms of Genetic and Evolutionary Algorithms. In particular, these are mutation, crossover and selection which are applied to an algebraic mathematical representation.

The representation is encoded in a tree. Both the parameters and the form of the equation are subject to the target space of all possible trees that representing

mathematical expressions. The operations are nodes in the tree and can be mathematical operations such as additions (add), multiplications (mul), abs, exp, etc. The terminal values of the tree consist of the function's input variables and real numbers (constants). The input variables are realized by the values of the training dataset.

In symbolic regression, many initially random symbolic equations compete to model experimental data in the most promising way. Promising are those solutions possessing a good compromise between better prediction quality of the observed data and the length of the mathematical formula.

Mutation in a symbolic expression can change the mathematical type of formula in different ways. For example, a div is changed to an add, the arguments of an operation are replaced (e.g., change $2*x$ to $3*x$), an operation is deleted (e.g., change $2*x+1$ to $2*x$), or an operation is added (e.g., change $2*x$ to $2*x+1$).

The objective in symbolic regression, like in other machine learning and data mining algorithms, is to minimize the regression error on the training data. After an equation reaches a desired level of accuracy, the algorithm returns the best equation or a set of good solutions (the Pareto front). In many cases the solution reflects the underlying principles of the observed system.

B. Symbolic vs Subsymbolic Representation

In the perspective of symbolic representations, the previously described methods can be categorized according to whether they are using symbolic or subsymbolic model representations.

As Smolensky [32] noted, the term subsymbolic paradigm is intended to suggest symbolic representations that are built out of many smaller constituents: "Entities that are typically represented in the symbolic paradigm by symbols are typically represented in the subsymbolic paradigm by a large number of subsymbols" (p. 3).

The debate over symbolic versus subsymbolic representations of human cognition is this: Does the human cognitive system use symbols as a representation of knowledge? Or does it process knowledge in a distributed representation in a complex and meaningful way? E.g., in neural networks the knowledge is represented in the parameters of the model. It is not possible to determine the exact position of the knowledge.

From this point of view, the syntactic role of subsymbols can be described as the subsymbols participate in numerical computation. In contrast, a single discrete operation in the symbolic paradigm is often achieved in the subsymbolic paradigm by a large number of much finer-grained operations. One well known problem with subsymbolic networks which have undergone training is that they are extremely difficult to interpret and analyze. In [33], it is argued that it is the inexplicable nature of mature networks. Partially, it is due to the fact that subsymbolic knowledge representations cannot be interpreted by humans and that they are black box knowledge representations.

C. Estimation Uncertainty

Commonly, the mean square error (MSE) of the predictor, which measures the average of the squared error over the validation data set in the cross validation, is the most accessible error information from the models. It is used as an objective function (loss function) to facilitate the model fitting. For the regression model assuming homoscedasticity (constant noise variances), MSE is also an estimation of the uncertainty measure of the predicted values. When the noise variances are assumed to be non-constant, MSE gives no clue to the uncertainty measure of predicted values.

As a advantage of Kriging, it provides the MSE of the estimator or so-called Kriging variance as a built-in feature. It is of significant importance in machine learning as well as global optimization. It directly shows the regions where Kriging model might perform badly (high variance). The Kriging variance is determined by the relative location between the training data, the location of the input to predict as well as the covariance structure.

D. Universal Function Approximators

As mentioned earlier, multi-layer neural networks can be considered as universal function approximators. The formal statement of universal approximation theorem [12] states that neural nets with single hidden layer can approximate any function which is continuous on n -dimensional unit hypercube. In [11] Cybenko has showed that a continuous function on a compact set can be approximated by a piecewise constant function. And a piecewise constant function can be represented as a neural net as follows. For each region where the function is constant, use a neural net as an indicator function for that region. Then build a final layer with a single node, whose input linear combination is the sum of all the indicators, with a weight equal to the constant value of the corresponding region in the original piecewise constant function. With this idea every continuous function can be represented with a neural network.

While Cybenko's result is an approximation guarantee Kolmogorov [34] proved that that a neural network provides an equality. Additionally, with heterogeneous transfer functions it can be proved that only $O(n^2)$ nodes are needed. It should be mentioned that Cybenko's result, with using only one type of activation function, is more relevant to machine learning.

III. ILLUSTRATIONS

Despite all the discussions in this paper, the behavior of the models is still quite vague at this moment. Thus, a small illustration of the model behavior would be necessary. We try to fulfill this task by showing the capability of modelling methods on the well-known 2-D Rastrigin function, whose highly rugged response surface is depicted in Fig. 1. By drawing 1000 points using Latin hypercube sampling in $[-5, 5]^2$, we build a Kriging, a polynomial regression and a RBFN model. In addition, SVMs with linear and polynomial

kernels are also constructed. The polynomial regression and SVM with polynomial kernel look similar and generally capture the global quadratic structure of the function but smooth out the surface. The SVM with linear kernel is expected to be a 2-D plane. The RBFs performs even better than SVMs due to the fact that it also shows a "bumpy" surface compared to the real surface. The Kriging model both reproduces the global trend of the function and includes small fluctuations although they are too small in scales.

IV. MODEL PROPERTY COMPARISONS

In order to obtain a more accessible view of the similarities and differences among all the function approximation methods described in this paper, we summarize the major feature, characteristic and properties of these models in two perspectives.

On one hand, we summarize the properties of models which are implied by their corresponding design principles in Table 1. Those intrinsic properties includes whether the model is symbolic or subsymbolic, the uncertainty assessment and time complexity, etc.

The models presented can be classified by whether they are designed based on symbolic or subsymbolic representations. As discussed in section II.B, such property determines whether the model knowledge can be understood by human.

Uncertainty measurement is another important aspect, providing additional knowledge on the quality/confidence of the model. In this case, Kriging model is distinguishing because the exact mean square error is available for the predicted values, as depicted in section II.C.

These models also differ in the interpretability. Some models have clear and meaningful explanations, e.g., the linear relation between the output and the input in linear regression. However, models like multi-layer perceptron has no direct implications.

The training methods vary on these models due to corresponding underlying assumptions and model complexity. Commonly used methods include least square estimation, maximization likelihood, back-propagation, cross-validation and mathematical programming. In some models (e.g., SVM), an additional training method is needed for the additional hyper-parameters. For example in SVM, quadratic programming is used to find the model weights while the cross-validation could also be applied to fit the parameters in the kernel function.

Despite the theoretical elegance of some modeling algorithms, the time complexity is crucial in the real application, where some of them might be not computationally feasible on large dataset. Kriging takes very high $O(n^3)$ effort for the model training compared to $O(n)$ of linear regression.

On the other hand, in terms of the structure and parameters contained in the model, we are also interested in what kind underlying mathematical structures the models assume/built upon, what mathematical structures/parameter could be learned from the training data or should be fixed by

the user. As a simple example, linear models assume a linear structure, where no parameters needed to provide by the users and the coefficients are learned from the data. In contrast, for symbolic regression and random forest, the tree structure is used. Such comparisons are listed in Table 2.

V. CONCLUSION

In this paper, we describe the most commonly used function approximation methods. The basic properties of them are discussed briefly. The meaning and effect of symbolic/subsymbolic representation is depicted. We also compare the ability of obtaining uncertainty measure for the models. We construct several models on the 2-D Rastrigin function and demonstrate the performance of approximation of these models. Finally, two tables are made to summarize and compare the essential conceptual properties of the models, where most of the important aspects are covered.

One finding is that already before fitting the model many decisions are made by the user. These decisions might restrict the capability of the models and ultimately this will unintentionally influence the prediction results. Universality properties do not practically solve these issues, as they are stated for a model whose size tends to infinity. A more meaningful approach to find models that can be used with confidence might be to use self-assessment of the

error/model consistency provided by the method itself (Kriging method) or to build models that can be interpreted by humans (symbolic regression, random forests). The symbolic regression framework is particularly interesting because it also frees the user from the burden of deciding on a symbolic model representation a-priori, in cases where no ‘natural’ functional expression can be assumed.

In the future work, it would be interesting to investigate how to combine a quantitative performance assessment with the qualitative assessment of methods, on which our work focused. Also, symbolic regression can naturally be combined with some of the other machine learning techniques, for instance by learning the structure of kernel function.

ACKNOWLEDGMENT

This work was supported by the EU Project I-RAMP3 (Intelligent Network Devices for fast Ramp-up), project homepage <http://www.i-ramp3.eu/>. I-RAMP3 is co-financed by the European Commission DG Research under the 7th Framework Program.

This work was also supported by Netherlands Organization for Scientific Research (NWO) project PROMIMOOC.

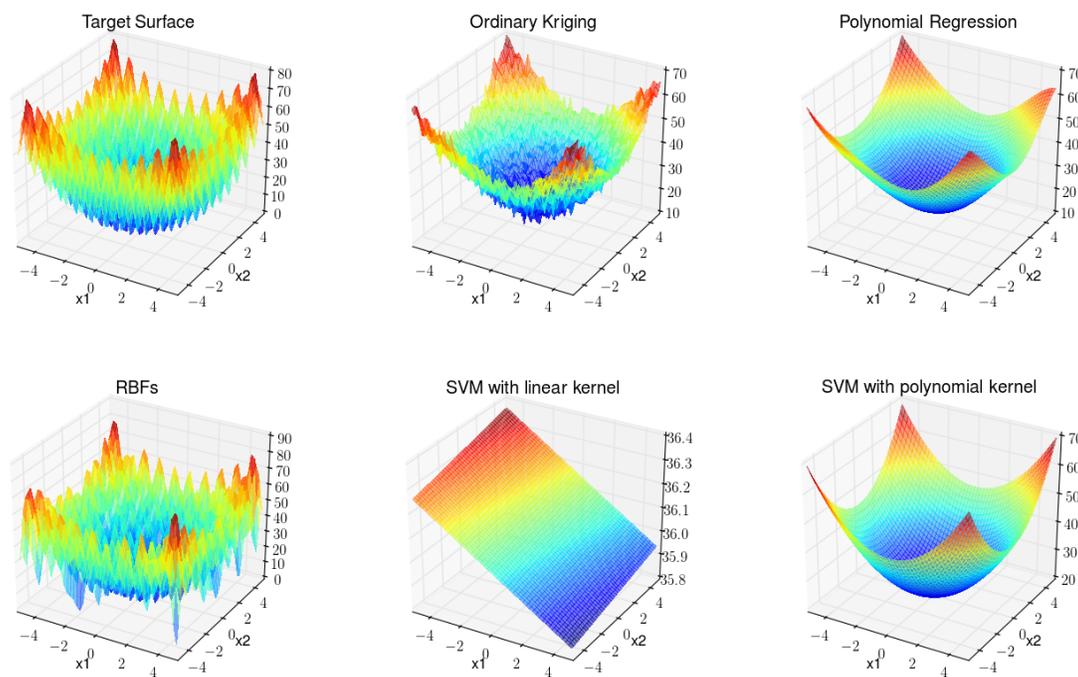


Figure 1 . On 2-D, the function approximations from 6 methods on Rastrigin function.

TABLE I. COMPARISON OF FUNCTION APPROXIMATION AND CLASSIFICATION MODELS BASED ON DIFFERENT PROPERTIES.

Methods	symbolic/ subsymbolic	uncertainty assessment	consistency check	interpretability	computational effort	training criterion/ method	exact interpolation	smoothness	Main purpose
Regression Models									
Linear Regression	subsymbolic	Mean Squared Error, global	Mean Squared Error	good	very small ($O(n)$)	least squares	no	yes, differentiable	function approximation
Polynomial Regression	subsymbolic	Mean Squared Error, global	Mean Squared Error, parsimony	good	scalable ($O(n^k)$), k is degree of polynomial	least squares	no	yes, differentiable	function approximation
Symbolic Regression	symbolic	Mean Squared Error, global	parsimony (Occam's razor)	excellent	scalable/high, typically heuristics are used	least squares	no	yes, differentiable	function approximation,
Random Process Models									
Ordinary Kriging	subsymbolic	Mean Squared Error, local; conditional variance	Likelihood value	medium	high in number of training points ($O(m^3)$), fast prediction ($O(m)$)	maximum likelihood, best linear unbiased prediction of a random process	yes	yes, continuous and for some correlation functions differentiable	function approximation
Universal Kriging	subsymbolic	Mean Squared Error, local; conditional variance	Likelihood value	medium	high in number of training points ($O(m^3)$), fast prediction ($O(m)$)	maximum likelihood, best linear unbiased prediction of a random process	yes	yes, continuous and for some correlation functions differentiable	function approximation
Support Vector Machines									
SVN with linear kernel	subsymbolic, variable structure	Distances of training vectors to the separation plane.	penalty function value	medium	medium (quadratic programming)	minimization of distances to separating hyperplane	no	yes	Classification(binary)
SVN with RBF kernel	subsymbolic	Mean squared error	penalty function value	medium	medium (quadratic programming)	minimization of distances to separating hyperplane, after transformation	no	yes	Classification(binary)
SVN with polynomial Kernel	subsymbolic	Mean squared error	penalty function value	medium	medium (quadratic programming)	minimization of distances to separating hyperplane, after transformation	no	yes	Classification(binary)
Decision tree models									
Decision trees	structural	Global training errors	training error rates	good	fast	greedy split algorithm	no	discontinuous	Classification
Random Forests	structural	Global training errors	training error rates	good	fast-medium	randomized search	no	discontinuous	Classification
Rule Ensembles	structural	Distance from decision threshold (local), training error (global)	not applicable	good	depending on number and effort to compute rules, typically heuristics are used	error minimization (diverse methods, Michigan/Pittsburgh style)	no	discontinuous	Classification
Piecewise defined functions									
Splines	subsymbolic	no	not applicable	medium	medium, typically quadratic time complexity, prediction is very fast	smoothness maximization	no	optimal	function approximation
K Nearest Neighbors	subsymbolic	no	not applicable	good	very fast training and prediction	averaging, transduction	only for $k=1$	discontinuous	function approximation, binary class separation
Connectionist Models									
Radial Basis Functions	subsymbolic, variable structure+	Mean Squared Error	not applicable	intransparent	prediction in linear time, training time depends on number of RBF centers $O(n^3)$	least squares	only, if no. of basis functions equals or exceeds no. of training points	continuous, typically also differentiable	
Multilayer Perceptron	subsymbolic, variable structure+	Mean Squared Error	not applicable	intransparent	fast, training time depends on number of neurons	least squares, backpropagation	no	continuous, differentiable	

TABLE II. COMPARISON OF MODELS, ROLES OF USER, LEARNING COMPONENT AND FRAMEWORK

Methods/Model Components	Structures fixed by framework	Structures fixed by user	Structures learned	Parameters fitted from data	Parameters set by user
Regression Models					
Linear Regression	linear function	no	no	coefficients	no
Polynomial Regression	polynomial function	no	no	coefficients, degree of polynomial	degree of polynomial
Symbolic Regression	Use of expression trees	Detailed grammar of expression tree, terminal symbols, expressions used	Symbol set, Terminal Symbols, Structure of Function tree	Numerical constants in formula	no
Random Process Models					
Ordinary Kriging	Linear predictor	Correlation function type		Mean value, global variance, correlation hyperparameters	no
Universal Kriging	Linear predictor , regression function (free choice)	Type of regression function and type of the correlation function	no	parameters of regression function	no
Support Vector Machines					
SVM with linear kernel	linear hyperplane as separator	type of SVM (e.g. squared deviations, Tchebycheff)	no	support vectors	no
SVM with RBF kernel	RBF kernel	type of RBF Kernel, type of SVM (e.g. squared deviations, Tchebycheff)	no	kernel parameters	no
SVM with polynomial Kernel	polynomial function	type of SVM (e.g. squared deviations, Tchebycheff)	no	kernel parameters	no
Decision tree models					
Decision trees	tree structure, separation by hyperplanes	no	tree structure	thresholds, parameters of tree	no
Random Forests	set of decision trees, separation by hyperplanes	no	tree structures, number of trees		no
Rule Ensembles	way to combine rules (e.g. weighting)	rule set, rules are generated automatically in some cases	subset selection of rules, in some cases structure of rules	weights of rules	no
Piecewise defined functions					
Splines	Basis functions, way to superpose basis functions	Type of splines, type of basis functions	no	parameters of basis functions	no
K Nearest Neighbors	no	distance function, averaging function	no	no	Number of neighbors K
Connectionist Models					
Radial Basis Functions	Type of functions (positive definite basis functions based on distance)	kernel function type, number of kernel functions, method to select the RBF centers	no	weights of radial basis functions, smoothness parameters	number of RBF centers
Multilayer Perceptron	Superposition of activation functions	activation function type, topology of network (number of neurons, number of layers)	no, although some advanced neural networks have the capability to learn topology	weights of activation functions and smoothness parameters in activation functions	no

REFERENCES

- [1] D. Krige, "A statistical approach to some basic mine valuation problems on the Witwatersrand," *Journal of Chemical, Metallurgical, and Mining Society of South Africa*, vol. 52, no. 6, pp. 119-139, December 1951.
- [2] C. E. Rasmussen, and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*: The MIT Press, 2005.
- [3] C. Cortes, and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, Pittsburgh, Pennsylvania, USA, 1992, pp. 144-152.
- [5] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171-1220, 2008.
- [6] C. M. Bishop, *Neural networks for pattern recognition*: Oxford university press, 1995.
- [7] D. O. Hebb, *The organization of behavior: A neuropsychological theory*: Psychology Press, 2005.
- [8] W. S. McCulloch, and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115-133, 1943.
- [9] S. Haykin, *Neural Networks: A Comprehensive Foundation*: Prentice Hall PTR, 1998.
- [10] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [11] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303-314, 1989.
- [12] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251-257, 1991.
- [13] E. Novak, *Deterministic and stochastic error bounds in numerical analysis*: Springer-Verlag Berlin, 1988.
- [14] A. E. Bryson, W. F. Denham, and S. E. Dreyfus, "Optimal programming problems with inequality constraints," *AIAA journal*, vol. 1, no. 11, pp. 2544-2550, 1963.
- [15] M. D. Buhmann, *Radial basis functions: theory and implementations*: Cambridge university press, 2003.
- [16] D. S. Broomhead, and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks," *Complex Systems* vol. 2, pp. 321-355, 1988.
- [17] F. Schwenker, H. A. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural networks*, vol. 14, no. 4, pp. 439-458, 2001.
- [18] K. Giannakoglou, "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence," *Progress in Aerospace Sciences*, vol. 38, no. 1, pp. 43-76, 2002.
- [19] M. Emmerich, "Single-and multi-objective evolutionary design optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421-439, August 2006.
- [20] L. Magee, "Nonlocal behavior in polynomial regressions," *The American Statistician*, vol. 52, no. 1, pp. 20-22, 1998.
- [21] K. L. Judd, *Numerical methods in economics*: The MIT Press, 1998.
- [22] C. Runge, "Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten," *Zeitschrift für Mathematik und Physik*, vol. 46, no. 224-243, pp. 20, 1901.
- [23] C. De Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50-62, 1972.
- [24] K. J. Holyoak, J. H. Holland, and J. H. Holland, *Induction: Processes of inference, learning, and discovery*: The MIT Press, 1989.
- [25] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*: The MIT Press, 1992.
- [26] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*: CRC Press, 2012.
- [27] R. Maclin, and D. Opitz, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, 1999.
- [28] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [29] L. Breiman, "Arcing classifier (with discussion and a rejoinder by the author)," *The annals of statistics*, vol. 26, no. 3, pp. 801-849, 1998.
- [30] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123-140, 1996.
- [31] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49-64, 1996.
- [32] P. Smolensky, "On the proper treatment of connectionism," *Behavioral and brain sciences*, vol. 11, no. 01, pp. 1-23, 1988.
- [33] D. A. Robinson, "Implications of neural networks for how we think about brain function," *Behavioral and brain sciences*, vol. 15, no. 04, pp. 644-655, 1992.
- [34] V. M. Tikhomirov, "On the Representation of Continuous Functions of Several Variables as Superpositions of Continuous Functions of one Variable and Addition," *Selected Works of A. N. Kolmogorov, Mathematics and Its Applications (Soviet Series)* V. M. Tikhomirov, ed., pp. 383-387: Springer Netherlands, 1991.

Test Platform for the Performance Evaluation of OPC-UA Servers for Fast Data Transfer Between Intelligent Equipment

Flavio González Vázquez

Ultraclean Technology and Micromanufacturing
Fraunhofer Institute for Manufacturing Engineering and Automation (IPA)
Stuttgart, Germany
e-mail: Flavio.Gonzalez@ipa.fraunhofer.de

Abstract— The ubiquity and widespread support of the Object Linking and Embedding for Process Control (OPC) Unified Architecture protocol in the automation industry and also outside it enable millions of intelligent manufacturing devices to effectively communicate and exchange information between each other in a secure and standardized manner. On the other hand, the ever growing need for large data transfers for predictive maintenance, process visualization and Internet of Things, among others, requires a precise knowledge of the protocol and system limitations in order to plan migrations or new installations. In this paper, a flexible test platform for the performance evaluation of OPC-UA systems is presented, along with preliminary findings and comparative performance measures of three different categories of PC-based OPC-UA systems.

Keywords— *Object Linking and Embedding for Process Control Unified Architecture (OPC-UA); performance evaluation; intelligent manufacturing systems; test platform; industrial automation.*

I. INTRODUCTION

The requirements of quick and cost-effective integration and data visualization of heterogeneous intelligent automation systems has rapidly popularized the usage of OPC Unified Architecture (OPC-UA) as communication protocol in the automation industry. In comparison with the classic Object Linking and Embedding for Process Control (OPC) protocol that has been in use since 1996 for Windows-based systems, OPC-UA achieves platform independence, better scalability and a more secure approach based on newer standards, among other benefits. However, with now the majority of PLC and equipment manufacturers supporting the OPC-UA protocol, and given the fact that the protocol is based on the Transmission Control Protocol (TCP) that does not have real-time requirements, questions about the performance of the protocol and specially of different OPC-UA server implementations arise when planning new installations or considering upgrading existing ones. Different applications like inter-equipment data transfer for processing, process visualization (Supervisory Control And Data Acquisition, or SCADA), data transfer for archival, predictive maintenance or Internet of Things (IoT) networks have different requirements, but a common one is the need for transferring large amounts of information. A decisive factor during the planning and realization phase is

the knowledge of the protocol and system limitations and performance numbers when a large data volume is involved. Although tools or mechanisms [1] already exist to provide some information, and performance tests have been made ([2] Chapter 13 “Performance”, or [3]), in order to systematically test a possible OPC-UA system and its performance measures and to provide decisive information to support an OPC-UA system choice based on custom and mixed criteria, a test platform for the benchmarking of OPC-UA systems has been developed.

This paper contains five additional sections. The first one, Section II, presents the requirements of the developed test platform. Section III describes the parameters under consideration for the study, while the test procedure itself and the details about the considered use-cases and information of interest are discussed in Section IV. Section V describes the setup for the tests and in Section VI some preliminary results are presented, as well as an outlook for future work on this matter.

II. TEST PLATFORM

In order to gather data in a consistent and reproducible manner, a test program was developed to permit the execution of test procedures in a flexible way. The technical goals set for the implementation were:

- a) The test procedure should not be hardcoded in the program, in order to allow quick modifications of the test procedures for fast parameter fine-tuning.
- b) Test variations should be easy to describe, so that experiments based on previous tests are easier to create and execute.
- c) It should be possible to store, reproduce and execute different test procedures.

```
### TEST
test: Concurrent requests
name: concurrent_requests
cycles: 100

concurrency: 1
resource: ns=4;s=MAIN.valueDesc
measure

cycles: 25

count: 1
concurrency: 40
measure

count: 40
measure
```

Figure 1. Sample test description document.

d) Test execution times should be stored in a machine-readable format in order to allow further data processing like statistic generation.

In order to enable the execution of tests in a flexible way, a test description document was created. One or multiple tests can be described in a single file, and measurements can be issued with a simple command. A sample test described on this format is shown in Figure 1. The commands used will be described on the coming sections.

III. PARAMETERS UNDER TEST

For the results discussed in this paper, the following parameters were considered, and thus are available in the test description document for adjustment:

a) Node. The OPC-UA node under test can be specified with this parameter, effectively selecting the amount of data that will be transferred per node.

b) Number of cycles. For the purpose of calculating average execution times, the total number of execution cycles can be defined per measurement, and the measurements can be averaged per cycle (combinable with concurrency and number of nodes per request).

c) Concurrency. This parameter defines the number of requests that are sent at the same time to the server, combinable with number of cycles and number of nodes per request. A pool of threads is initialized, and all request workers are instantiated to fetch the defined node at the same time. The test execution waits for all threads to be completed; therefore, the measurements derived from using the concurrency parameter include the amount of time taken by the longest request thread.

d) Number of nodes per request. As described in [2], p. 125, OPC-UA requests can contain a list of nodes to read or write in order to reduce overhead. As such, this parameter determines the number of nodes fetched per read/write request. If this parameter is set to any value greater than 1, the same node is requested multiple times. The number of nodes per request is combinable with the number of cycles and concurrency parameters.

e) Security. Determines the preference for a secured, encrypted endpoint when connecting to an OPC-UA server. OPC-UA endpoints are sorted in descending order of security, and when security is set to *on*, the first endpoint from the available list is chosen, otherwise the last one when set to *off*.

IV. TEST PROCEDURE AND DESCRIPTION OF GATHERED DATA

In order to draw representative conclusions with the different tests, four variables were made available on all PLC programs with varying sizes:

- 1) Real variable with a total memory usage of 32 bits (REAL), representing a typical single numerical variable.
- 2) Array of 512 integer numbers of 16 bit each totaling 1 kilobyte of payload (PAGE), representing an

average visualization page containing multiple variables.

- 3) Byte array of 20.000 elements (SDD), representing a sample self-description document in text format.
- 4) Byte array of 65.535 elements (IMAGE), representing a small image in binary format.

With the aforementioned resources at the disposal on the PLCs, the following tests were conducted:

- a) Data volume. The amount of time required to fetch nodes containing variables or different data sizes was tested. Each of the 4 variables was requested in blocks of 1 or 40, one thousand times (one thousand requests of one node/40 nodes each).
- b) Grouped requests. The benefit and reduction of overhead by fetching multiple nodes in one request was tested. The 4 variables were fetched in grouped requests of 40 and 400 variables per request and run 100 and 50 cycles respectively, and the times were, after calculating the average time per variable fetched, compared with single variable requests.
- c) Concurrent requests. The channel and server efficiency responding to multiple concurrent requests were tested by sending multiple requests at the same time for fetching individual nodes.
- d) Security. The impact of the transport encryption was the main purpose of this test. The four variables under test were fetched thousand times in a single variable per request basis, once through a secure endpoint, once through an unsecured endpoint.

The test description document is interpreted by the test platform program line by line, creating new test instances as required and configuring the test instance appropriately. Upon reading a *measure* command, the current test instance is executed and the execution times and other data are gathered in a file. The following data is contained in the produced file as the result for the test, in addition to the test name and timestamp:

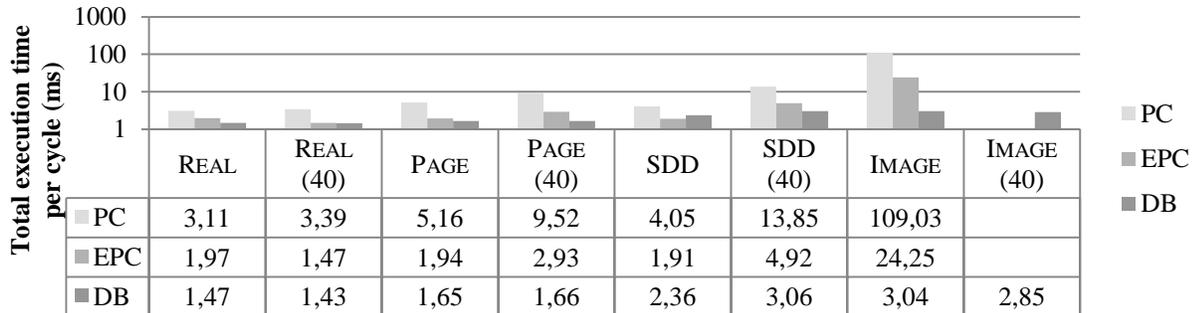
- a) Number of resources n_r fetched in total during the test, calculated as $n_r = n_c \cdot n_t$ where n_c is the number of nodes per request, and n_t is the number of threads spawned at the same time.
- b) Total duration t taken for the complete test instance to execute.
- c) Average time per cycle \bar{t}_c calculated as $\bar{t}_c = t/c$ where c is the number of execution cycles as defined in the test description document (or 1 by default).
- d) Average time per resource \bar{t}_r calculated as $\bar{t}_r = \bar{t}_c/n_r$.
- e) Standard deviation σ calculated as $\sigma = (\sum_{c=0}^{n_c} (t_c - \bar{t}_c)^2)/n_c$.
- f) Coefficient of variation c_v calculated as $c_v = \sigma/\bar{t}_c$.

V. EQUIPMENT UNDER TEST AND SETUP

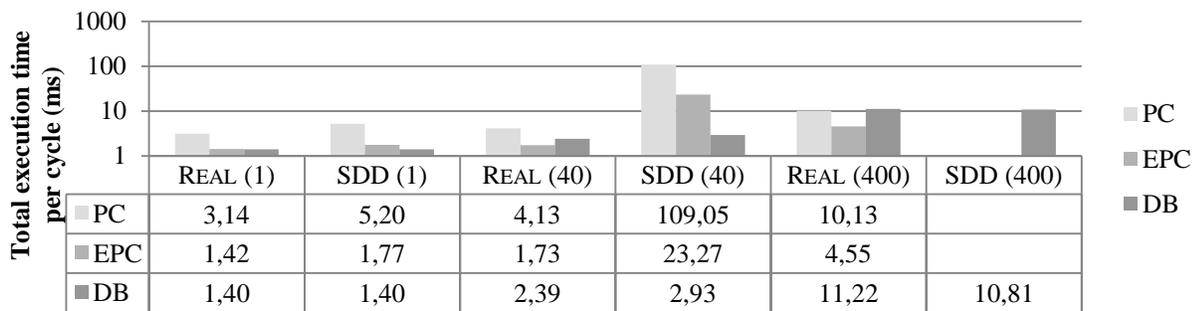
The aforementioned test procedure was executed against three PC-based PLCs running compatible OPC-UA

interfaces: a high-end Microsoft Windows 7-based PLC (PC) running on a dual core processor and using traditional hard disk drives for storage, a lower-end Microsoft Windows Embedded-based PLC (EPC) running on a single core processor and using a flash-based memory storage device,

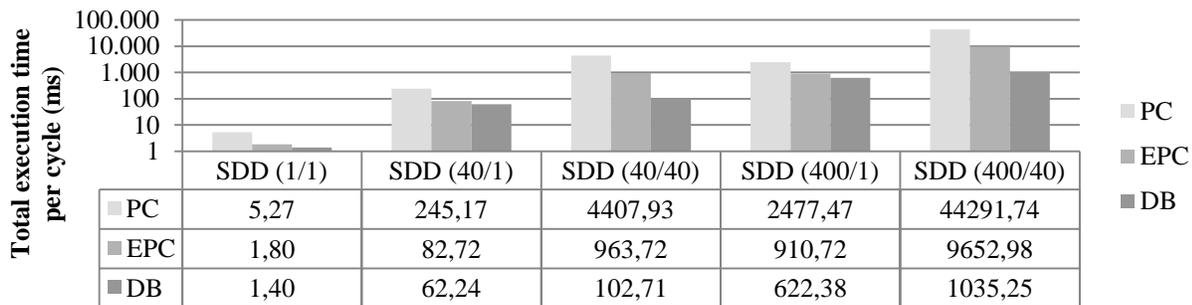
and a development board running an embedded OPC-UA implementation. The client test platform was executed in a standard PC computer, running Microsoft Windows 7 and a .NET OPC-UA stack implementation, and connected directly



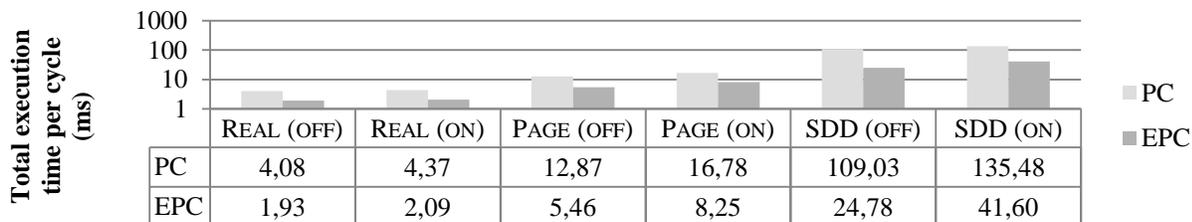
a) Data volume test. Total execution times for the four different variables by fetching 1 and 40 nodes per request, respectively.



b) Grouped requests test. Total execution times for the REAL and SDD variables by fetching 1, 40 and 400 nodes per request, respectively.



c) Concurrent requests test. Total execution times for the SDD variable by spawning 1, 40 and 400 parallel threads, with 1 and 40 nodes per request (n_t/n_c).



d) Security test. Total execution times for the Real, Page and SDD variables through an unsecured and an encrypted channel, respectively.

Figure 2. Comparison of execution times of the different tests cases.

and separately for each test with every equipment under test with a Cat 6 network cable working in a 100 Mbit/s operation mode. Since in most of the cases the OPC-UA server implementation is not an interchangeable component in the PLC, the complete system was tested, measuring the total time taken by a request to be serialized and processed by the OPC-UA client (t_0), the request to travel to the OPC-UA server (t_1), the request to be deserialized and processed by the OPC-UA server (t_2), the variable to be fetched internally on the PLC (t_3), the variable to be processed internally on the PLC (t_4), the variable to be returned to the OPC-UA server (t_5), the response to be processed and serialized by the OPC-UA server (t_6), the response to travel back to the OPC-UA client (t_7) and the response to be deserialized and processed by the OPC-UA client (t_8), as depicted in Figure 3. The test cases described in Section IV were executed on the three PLCs, yielding the execution times depicted in Figure 2.

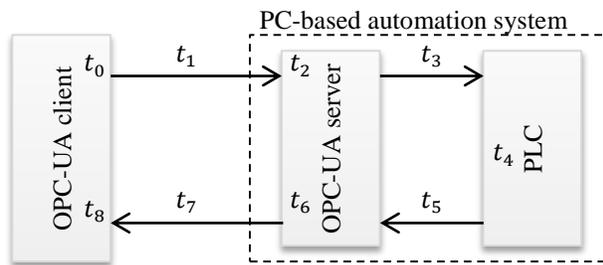


Figure 3. System under test and different processing and travel times.

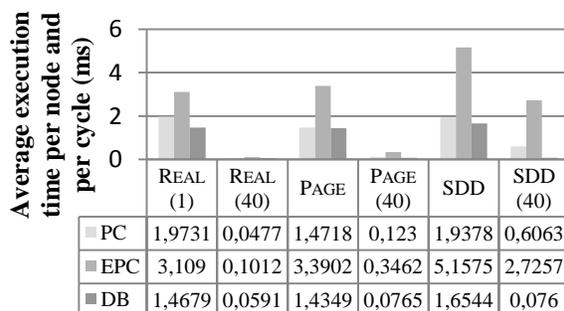


Figure 4. Comparison of average execution times (of 1000 execution cycles) per node when fetching the REAL, PAGE and SDD variables in single requests versus 40 nodes per request.

VI. CONCLUSION AND FUTURE WORK

From the gathered preliminary results, some conclusions can be drawn:

- a) As expected, requests with single nodes take much overhead, and grouping many variables into a request saves time on all tested implementations, as shown in Figure 4.
- b) Given the low RAM available on the EPC and the comparatively slower flash storage used for the virtual memory, larger variables take a considerable time impact, when memory swapping is required and slows down the entire system.

- c) Sending multiple requests in parallel is generally a good idea if a large amount of nodes need to be requested at a given time, as all implementations showed drastic time savings by fetching multiple variables separated into requests sent at the same time. Combining multiple threads with multiple nodes per request further improve time savings.
- d) As expected, choosing a secure and encrypted endpoint has a speed penalty, although it might not be a deciding factor if security is a requirement. In the tests, a Basic128Rsa15 security policy and a security mode of Sign & Encrypt was used as required by two of the three OPC-UA implementations that implemented secured endpoints.
- e) Since the platform tested the whole system, the development board outperformed the other two higher-end systems as the internal communication allowed for faster data retrieval.

In this paper, an OPC-UA benchmarking tool has been presented, in addition to some preliminary results. The testing platform enables the capture of performance information in a flexible and reproducible way, and makes it possible to describe tests using any combination of parameters permitting the compilation of exactly the required data to make decisions about the amount of information that it is possible to transfer in a given application, or the speed limitations when planning a facility. The results presented here are preliminary, and more statistics and parameters are foreseen to be included in the testing application, configurations to better simulate the conditions that a facility will face, as well as a more detailed test scenario including specific PLCs with varied configurations and optimizations. Comparisons between software- and hardware-based PLCs, as well as with variations in the network equipment (like cable length, class and age) and bus couplers are planned to be the subject of further evaluation.

ACKNOWLEDGMENT

F.G.V. would like to thank Pablo Mayer and Fabian Böttinger for their help building and configuring the equipment under test described on this paper, as well as for their invaluable advice on this project.

REFERENCES

- [1] "Resource Efficiency Testing, Step by Step Instructions (whitepaper)," 16 May 2014. [Online]. Available: https://opcfoundation.org/wp-content/uploads/2014/05/Certification_Resource_Efficiency_Testing.pdf. [Accessed 22 May 2015].
- [2] W. Mahnke, S.-H. Leitner, and M. Damm, OPC Unified Architecture, Springer-Verlag Berlin Heidelberg, 2009.
- [3] C. Salvatore and G. Cutuli, "Performance evaluation of OPC UA," in 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Bilbao, 2010, pp. 1-8.
- [4] J. Lange and F. Iwanitz, OPC, From Data Access to Unified Architecture, VDE Verlag, 2010.