PATTERNS 2018

The Tenth International Conferences on Pervasive Patterns and Applications

ISBN: 978-1-61208-612-5

February 18 - 22, 2018

Barcelona, Spain

PATTERNS 2018 Editors

Herwig Mannaert, University of Antwerp, Belgium

Christian Kohls, TH Köln, Germany

Yuji Iwahori, Chubu University, Japan

Hiroyasu Usami, Chubu University, Japan

Peter De Bruyn, University of Antwerp, Belgium

PATTERNS 2018

Forward

The Tenth International Conferences on Pervasive Patterns and Applications (PATTERNS 2018), held between February 18 - 22, 2018 - Barcelona, Spain, continued a series of events targeting the application of advanced patterns, at-large. In addition to support for patterns and pattern processing, special categories of patterns covering ubiquity, software, security, communications, discovery and decision were considered. It is believed that patterns play an important role on cognition, automation, and service computation and orchestration areas. Antipatterns come as a normal output as needed lessons learned.

The conference had the following tracks:

- Patterns basics
- Patterns at work
- Discovery and decision patterns

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 PATTERNS 2018 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 PATTERNS 2018. 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 PATTERNS 2018 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope PATTERNS 2018 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 pervasive patterns and applications. We also hope that Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

PATTERNS 2018 Chairs

PATTERNS Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Claudia Raibulet, University of Milano-Bicocca, Italy
Sneha Chaudhari, Carnegie Mellon University, Pittsburgh, USA
Valerie Gouet-Brunet, IGN, LaSTIG, MATIS, France
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria
Markus Goldstein, Ulm University of Applied Sciences, Germany
Stan Z. Li, Institute of Automation - Chinese Academy of Sciences, China
Reinhard Klette, Auckland University of Technology, New Zealand
Jacqueline Daykin, King's College London, UK / Aberystwyth University, Mauritius Branch Campus

PATTERNS Industry/Research Advisory Committee

Charles Perez, Paris School of Business, France
Christian Kohls, TH Köln, Germany
Krzysztof Okarma, West Pomeranian University of Technology, Szczecin, Poland
George A. Papakostas, Eastern Macedonia and Thrace Institute of Technology, Greece
Adel Al-Jumaily, University of Technology, Australia

PATTERNS 2018

Committee

PATTERNS Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Claudia Raibulet, University of Milano-Bicocca, Italy
Sneha Chaudhari, Carnegie Mellon University, Pittsburgh, USA
Valerie Gouet-Brunet, IGN, LaSTIG, MATIS, France
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria
Markus Goldstein, Ulm University of Applied Sciences, Germany
Stan Z. Li, Institute of Automation - Chinese Academy of Sciences, China
Reinhard Klette, Auckland University of Technology, New Zealand
Jacqueline Daykin, King's College London, UK / Aberystwyth University, Mauritius Branch Campus

PATTERNS Industry/Research Advisory Committee

Charles Perez, Paris School of Business, France
Christian Kohls, TH Köln, Germany
Krzysztof Okarma, West Pomeranian University of Technology, Szczecin, Poland
George A. Papakostas, Eastern Macedonia and Thrace Institute of Technology, Greece
Adel Al-Jumaily, University of Technology, Australia

PATTERNS 2018 Technical Program Committee

Mourad Abbas, STRCDAL, Algeria
Adel Al-Jumaily, University of Technology, Sydney, Australia
Adel M. Alimi, University of Sfax, Tunisia
Berkay Aydin, Georgia State University, USA
Johanna Barzen, Universität Stuttgart, Germany
Abdel Belaïd, LORIA, France
Hatem Ben Sta, Université de Tunis - El Manar, Tunisia
Nadjia Benblidia, Saad Dahlab University - Blida1, Algeria
M.K. Bhuyan, IIT Guwahati, Assam, India
Silvia Biasotti, CNR – IMATI, Italy
Julien Broisin, University of Toulouse - Institut de Recherche en Informatique de Toulouse (IRIT), France
Michaela Bunke, Universität Bremen, Germany
Jocelyn Chanussot, Université Grenoble Alpes, France
Amitava Chatterjee, Jadavpur University, Kolkata, India
Sneha Chaudhari, Carnegie Mellon University, Pittsburgh, USA

Moin Nabi, SAP Innovation Center Network, Berlin, Germany
Serena Nicolazzo, University Mediterranea of Reggio Calabria, Italy
Antonino Nocera, University Mediterranea of Reggio Calabria, Italy
Jean-Marc Ogier, Université de La Rochelle, France
Krzysztof Okarma, West Pomeranian University of Technology, Szczecin, Poland
Hélder Oliveira, INESC TEC, Portugal
Alessandro Ortis, University of Catania, Italy
Mrutyunjaya Panda, Utkal University, India
George A. Papakostas, Eastern Macedonia and Thrace Institute of Technology, Greece
Giuseppe Patane', CNR-IMATI, Italy
Christian Percebois, IRIT - University of Toulouse, France
Charles Perez, Paris School of Business, France
Agostino Poggi, DII - University of Parma, Italy
Giovanni Puglisi, University of Cagliari, Italy
Rafael Radkowski, Iowa State University, USA
Claudia Raibulet, University of Milano-Bicocca, Italy
Giuliana Ramella, CNR - National Research Council, Italy
José Raúl Romero, University of Córdoba, Spain
Theresa-Marie Rhyne, Independent Visualization Consultant, USA
Alessandro Rizzi, Università degli Studi di Milano, Italy
Gustavo Rossi, UNLP - La Plata, Argentina
Antonio-Jose Sanchez-Salmeron, Universitat Politecnica de Valencia, Spain
María-Isabel Sanchez-Segura, Carlos III University of Madrid, Spain
José Santos Reyes, University of A Coruña, Spain
Muhammad Sarfraz, Kuwait University, Kuwait
Alessandra Scotto di Freca, Università degli studi di Cassino e del Lazio Meridionale, Italy
Giuseppe Serra, University of Udine, Italy
Isabel Seruca, Portucalense University, Portugal
Patrick Siarry, Université Paris-Est Créteil, France
Mu-Chun Su, National Central University, Taiwan
Yusuke Takezawa, Tokyo Denki University, Japan
Dan Tamir, Texas State University, USA
Shanyu Tang, University of West London, UK
J. A. Tenreiro Machado, Polytechnic of Porto, Portugal
Daniel Thalmann, Institute for Media Innovation (IMI) - Nanyang Technological University, Singapore
Mati Tombak, University of Tartu, Estonia
Andrea Torsello, Universita' Ca' Foscari Venezia, Italy
Hiroyasu Usami, Chubu University, Japan
Michael Vassilakopoulos, University of Thessaly, Greece
Nicole Vincent, Université Paris Descartes, France
Panagiotis Vlamos, Ionian University, Greece
Krzysztof Walczak, Poznan University of Economics, Poland
Patrick Wang, Northeastern University, USA
Laurent Wendling, Paris Descartes University (Paris 5), France
Reuven Yagel, The Jerusalem College of Engineering, Israel
Hongchuan Yu, Bournemouth University, UK
Pong C. Yuen, Hong Kong Baptist University, Hong Kong

Copyright Information

Table of Contents

Improving Speech Emotion Recognition Based on ToBI Phonological Representation *Lingjie Shen and Wei Wang*	1
How an Optimized DB-SCAN Implementation Reduces Execution Time and Memory Requirements for Large Data Sets *Ole Kristian Ekseth and Svein-Olaf Hvasshovd*	6
Finding and Implementing Patterns for Creative Spaces *Christian Kohls*	12
Automatic Construction of Image Data Set Using Query Expansion Based on Tag Information *Yuki Mizoguchi, Takahiro Yoshimura, Yuji Iwahori, and Ryosuke Yamanishi*	18
Automatic Construction of Large Scale Image Data Set from Web Using Ontology and Deep Learning Model *Takahiro Yoshimura, Yuki Mizoguchi, Yuji Iwahori, and Ryosuke Yamanishi*	21
Colon Blood Vessel Detection Based on U-net *Hiroyasu Usami, Yuji Iwahori, Naotaka Ogasawara, and Kunio Kasugai*	23
Control Transition Interfaces in Level 3 Automated Vehicles: Four Preliminary Design Patterns *Alexander G. Mirnig and Manfred Tscheligi*	25
Massively Parallel Optical Flow using Distributed Local Search *Abdelkhalek Mansouri, Jean-Charles Creput, Fabrice Lauri, and Hongjian Wang*	31
Improved Gas Detection Algorithm for FTIR-Based Hyperspectral Imaging System Using Normalized Matched Filter *Hyeong-Geun Yu, Jae-Hoon Lee, Dong-Jo Park, Hyeon-Woo Nam, and Byeong-Hwang Park*	37
Evolvable Documents – an Initial Conceptualization *Marek Suchanek and Robert Pergl*	39
On the Variability Dimensions of Normalized Systems Applications: Experiences from an Educational Case Study *Peter De Bruyn, Herwig Mannaert, and Philip Huysmans*	45
On the Evolvability of Code Generation Patterns: The Case of the Normalized Systems Workflow Element *Herwig Mannaert, Peter De Bruyn, Koen De Cock, and Tim Van Waes*	51

Improving Speech Emotion Recognition Based on ToBI Phonological Representations

Lingjie Shen ,Wei Wang
Machine Learning & Cognition lab, School of Education Science, Nanjing Normal University
Nanjing, China
Email: 602249910@qq.com, 769370106@qq.com

***Abstract*—The improvement of Speech Emotion Recognition (SER) relies on the classifiers and features. In terms of feature selection, so far, most of the research only uses a large set of acoustic features which cannot shed lights on the relationship between emotion and phonology. In our study, we improve SER by combining acoustic features and phonological representations together. We improve the SER on the public IEMOCAP database by combing acoustic and phonological features together under leave-one-speaker-out cross validation framework. Support vector machine, logistic regression, multi-layer perceptron and deep learning method of convolutional neural network (CNN) are used in our experiment. With phonological representations, CNN provides 60.22% of unweighted average recall (UAR) on categorical emotion recognition on utterance level which is now the state-of-the-art. When compared to the conventional baseline system based only on acoustic features, the proposed system with combing features gets 7.15% improvement of UAR in four basic emotion classification.**

Keywords-speech emotion recognition; acoustic features; phonology; deep learning.

I. INTRODUCTION

Automatic emotion recognition from speech has been an active research area in past years, which is of great interest for human computer interactions. It has wide applications ranging from computer tutoring applications to mental health diagnostic application [1]. Since the speech recognition has already changed people's life, detecting emotion from the speech is another challenge to improve the user-friendly human machine interaction.

Automatic Speech Emotion Recognition (SER) has been an active research area in past decades and is of great interest for human computer interactions. An efficient human emotion recognition system will help to make the interaction between human and computer more natural and friendly. It has wide applications ranging from computer tutoring applications to mental health diagnostic applications [1].

Accuracy of speech emotion recognition mainly relies on two factors, i.e., classifiers and features. In terms of features used in SER, acoustic features have been used as the dominant features in the literature. These acoustic features include frame-level features called Low Level Descriptors (LLDs) and their corresponding functionals which are used to map LLDs in the segment level to the utterance level. Most research of automatic emotion recognition usually relies on a large set of features and the reasons are as follows. First, so far there is no "standard" feature set for generic speech SER. Second, it is not clear which speech features are the most powerful in distinguishing emotions. Third, the acoustic variability introduced by the existence of different sentences, speakers, speaking styles, and speaking rates adds another obstacle to feature selection because these properties directly affect most of the common extracted speech features such as pitch and energy contours [2]. Therefore, most studies apply a large "brute-force" feature selection method which captures the dynamic temporal character of the contours of acoustic features over segments corresponding to different tasks [3], and this has been shown to outperform modeling the temporal dynamics on the classifier level [4]. During the last ten years, different acoustic feature sets used for various speech tasks have been proposed and have become widely-used feature sets that are beneficial for researchers in comparing their results on the same task [4][5][6][7][8].

Although there is a clearly perceived connection between emotions and phonology [9], researchers still have not formed a satisfactory model linking the emotion and prosody though these large feature sets are correlative with phonology[10]. There is not an accurate mapping between emotions and phonology. Hence, our goal of this study is to find out the emotionally salient phonological features with ToBI label systems and figure out the relationship between phonology and emotions. Then, as indicated by Liscombe [9], paralinguistic information can be conveyed via both segmental information and suprasegmental information that describes phonological information, such as pitch, intonation stress, rhythm and duration. We therefore combined acoustic features obtained from segmental information and phonological representations obtained from suprasegmental information to further improve the speech emotion recognition.

In this paper, we present experiments on the IEMOCAP dataset conveying four basic emotions. The extracted feature vectors are used to develop a support vector machine, logistic regression, multi-layer perceptron and a convolutional neural network as classifiers to recognize the emotional state in the offline system. Two different classes of feature vectors were evaluated: (1) acoustic features and (2) fusion features of acoustic and ToBI [11] features.

The paper is organized as follows. Section 2 provides us with the related work and literature on the IEMOCAP

 ISBN: 978-1-61208-612-5

database. Section 3 gives the methodology including emotional models, features and classifiers used in our study. Experiments and results are presented in Section 4. Experiments include two parts. Finally, discussion and conclusion are presented in Section 5.

II. RELATED WORK

Recent studies on the IEMOCAP database, including the classifiers, features, labels and results of classification are presented in Table Ⅰ. From the Table, we can observe that on the utterance level, the best unweighted average recall (UAR) on the IEMOCAP database is 58.46% using hierarchical binary Bayesian logistic regression [12]. On the frame level, the best UAR is 60.89% using a convolutional neural network [13]. The algorithm research on emotion recognition changes from a traditional machine learning method to a deep learning method (i.e., convolutional neural network, recurrent neural network, etc.). Some studies extract emotionally salient parts of speech by the attention mechanism method [14], which is successfully applied in image and speech recognition fields. Most of the research is starting to focus on solving some problems which might be encountered in the wild rather than exploring new features to improve accuracy. Further, more modals like face, gesture and linguistic information are being added in emotion recognition. Another observation from the literature review is that the Geneva Minimalistic Acoustic Parameter Set (eGeMAPS feature set) [5] performed better than the low complexity Logmel filter-banks. This is contrary to results from the field of computer vision, where in recent years features extracted from raw data by convolutional layers have outperformed hand-crafted features and achieved state-of-the-art results in various tasks [15].

In terms of phonological features about emotions, Busso et al. [16] explore what aspects of the pitch contour are the most emotionally salient. This study presents an analysis of the statistics derived from the pitch contour. The results indicate that gross pitch contour statistics such as mean, maximum, minimum and range are more emotionally prominent than features describing the pitch shape. The study explores the devotion of pitch features to speech emotion recognition and forms an emotional profile from acoustic features. However, we still have no explainable results from this research to interpret the relationship between emotion and phonology.

To find the interpretable relation between emotion and phonology, there has been some research using the ToBI system to find the salient cues of emotions. Iliev et al. [17] use ToBI features to recognize angry, happy and sad. The authors also combine the acoustic features together with ToBI features to improve speech emotion recognition. However, they only use ToBI features relating to tonal information while omitting the break indices which also carry information about emotion. They also neglect the sequential information of ToBI features encoded in an utterance. Cao et al. [10] explore the phonological cues from the ToBI system to study the relationship between acted perceptually unambiguous emotion and phonology. They aim to analyze the predictive power of discrete characterizations of intonations in the ToBI framework to discriminate specific emotions. The study indicates that the discrete features from the ToBI system are comparable to the acoustic features but are not robust for sentence-independent emotion classification tasks. Another limitation of this study is that the database is not public, therefore the outcome is not objective. However, this study provides us with some hints about the relation between phonological cues and specific emotions. Our study is inspired by this work and we further attempt to improve speech emotion recognition based on the IEMOCAP database with deep learning method by incorporating phonological representations.

TABLE Ⅰ. THE LITERATURE OF EMOTION RECOGNITION ON IEMOCAP DATABASE

Classifiers	UAR (%)
Hierarchical binary Bayesian logistic regression [12]	58.46
Support vector machine (SVM) [18]	50.64
Convolutional neural network (CNN) [13]	58.28
Bidirectional Long-short term Memory Recurrent Neural Network (BLSTM) [14]	58.8

III. METHODOLOGY

We introduce the dataset and the features used in the experiment in this section. The features include acoustic features and phonological representations, respectively. We combine these two kinds of features together to improve speech emotion recognition and compare the classification performance with baseline system using the acoustic features only.

A. Data description

The database used in this work is the interactive emotional dyadic motion capture (IEMOCAP) database which contains approximately 12 hours of audio-visual data from five mixed gender pairs of actors [19]. Each recorded session lasts approximately 5 minutes and consists of two actors interacting with each other in scenarios that encourage emotional expression. In this study, we only focus on the audio channel to perform speech emotion recognition. We use the categorical tags of this database. Specifically the categorical tags that we are considering in the IEMOCAP corpus are: neutral, angry, happy, sad (we merge happy and excitement together as happy). In total, the data used in our experiments comprises 5531 utterances with an average duration of 4.5 s.

B. Acoustic features

The openSMILE toolkit [20] is chosen to extract the acoustic features and the baseline feature set of Interspeech 2010 paralinguistic challenge [7] is used for our tests. This extension intends to better reflect a broader coverage of paralinguistic information assessment. As shown in Table 3, it consists of 38 basis LLDs. 21 functionals are applied to the above 34 LLDs and their corresponding delta coefficients, while 19 functional are applied to 4 F0 related LLDs and their corresponding delta coefficients. In addition, the durations and F0 onsets are also considered and included into the feature set. Thus, the final acoustic features vector has a dimension of 1582 as shown in Eq. (1):

$$f_a = (a_1, a_2, a_3, \dots, a_{1582}) \quad (1)$$

where $a_1, a_2, a_3, \ldots, a_{1582}$ are the values of 1582 acoustic features.

C. Phonological representations

We use ToBI [21] to generate Phonological representation. TOBI labels encode the underlying phonological representation of an utterance primarily in terms of perceived pitch targets (H) high and (L) low and disjunctures between words (break 0-4 from minimal to strong). Perceptually prominent syllables, primarily due to pitch excursions but also lengthening and intensity, are associated with pitch accents that could consist of single tonal targets (H,!H*,L*), or bi-tonal combinations, most commonly L+H*,L*+H,H+!H*;!H represent a target down stepped from a preceding H target, and "*" corresponds to the tone aligned with the stressed syllable. For prosodic chunking, breaks 0 and 1 correspond to regular fluent word transitions, 2 to a perceived disjuncture with no salient tonal marking, 3 marks an intermediate phrase associated with H-,L-, or !H- targets. Regarding the break indices, diacritics describing uncertainty and disfluency were not used since the nature of the data elicitation minimized these phenomena and we wanted to mitigate data sparsity. Additionally, break 0 was not used. Every sentence's prosodic features consists of the times of every phonological representation in this sentence generated from the AuToBI [11] and become the one-hot-vector with fixed length. We have 141 phonological representations in total from AuToBI and Table Ⅱ presents the list of these features. The phonological feature is formulated as Eq. (2):

$$f_p = (p_1, p_2, p_3, \ldots, p_{141}) \tag{2}$$

where $p_1, p_2, p_3, \ldots, p_{141}$ are 141 prosodic events.

IV. EXPERIMENT AND RESULTS

We compare four classifiers' performance on speech emotion recognition based on acoustic features and fused features. The classifiers consist of traditional machine learning algorithms and deep learning method. The outcome shows that CNN outperforms other classifiers with fused features.

A. Experiment: emotion recognition using both two kinds of features

The motivation of this experiment is twofold. First, this experiment will compare the results with acoustic features only and with fused features (i.e., acoustic and phonological features) respectively. Second, we will implement several kinds of machine learning methods to improve the performance of speech emotion recognition on classifier level.

A ten-fold leave-one-speaker-out cross-validation scheme was employed in experiments using the nine speakers as training data and the one speaker as test data. Normalization is a critical step in emotion recognition.

The normalization method has an effect to experiment results. The goal of normalization is to eliminate speaker and recording variability while keeping the emotional discrimination. For this analysis, z-score normalization is implemented on all data, meaning that our speech emotion recognition is speaker-independent.

TABLE Ⅱ. LIST OF PHONOLOGICAL REPRESENTATIONS BASED ON TOBI LABELS.

Phonological Representations	Examples	Numbers
Break indices	Break indices 1 Break indices 3 Break indices 4	3
Phrasal tones	L- H- !H-	5
Pitch accent	H* !H* L+H*	6
Bigrams – pitch accent	H*,H* H*,!H* !H*,!H*	27
Bigrams – pitch accent with phrasal tones	H*,L- L*+H,INTONATIO NAL_BOUNDARY !H*,INTONATION AL_BOUNDARY	30
Bigrams – phrasal tones with pitch accents	L-,H* L-,!H* H-,!H*	48
Bigrams – phrasal tones	L-,L- L-,H- H-,H-	22

The classifiers used in our experiment are SVM with complexity 1, Logistic Regression (LR), Multi-Layer Perceptron (MLP) and CNN. Multi-layer perception has two hidden layer and hidden size is 50 and 20 respectively with Rectified Linear Unit (ReLU) activation function. The architecture of CNN is shown in Figure 1 and configuration is as follows. We use one-dimension convolution because the feature is one dimension. We have two convolutional layers each followed with one max pooling layer, one dense layer with 200 neuros. To provide a probabilistic interpretation of the model's output, the output layer utilizes a softmax nonlinearity instead of the nonlinear function used in previous layers. The activation function used in CNN is the ReLU due to its advantage over other activation functions, such as computational simplicity and faster learning convergence [22]. The base learning rate is set to 10-4 and optimizer is Adam [23]. The epoch is 10 and the training batch size is 32.

For support vector machine, logistic regression and multi-layer perception classifiers which are not deep learning methods, to avoid the curse of dimensionality, feature reduction is a necessary preprocessing. We use principle component analysis (PCA) to reduce the acoustic feature and fused feature (i.e., the concatenated acoustic and phonological features with dimension 1723) and we choose 100 and 120 as the number of components of acoustic feature and concatenated fused feature respectively. The number of components of PCA is chosen based on the best performance under SVM, LR and MLP in our pre-experiment. For CNN, we do not have to reduce the feature dimension due to CNN's advantages of sparse connectivity and shared weights.

The baseline system in our experiments is the classifier with acoustic features only. Under the baseline system, it is objective to validate the predictive power of phonological features in speech emotion recognition and see the improvement of UAR after phonological features are added.

As it is standard practice in the field of automatic speech emotion recognition, results are reported using Unweighted Average Recall (UAR) as Eq. (3) to reflect imbalanced classes [7].

$$\text{UAR} = \frac{1}{N}\sum_{i=1}^{N}\frac{c_i}{n_i} \tag{3}$$

where c_i is the number of correct examples of class i predicted by the classifier, n_i is the total number of examples of class i and N is the number of classes.

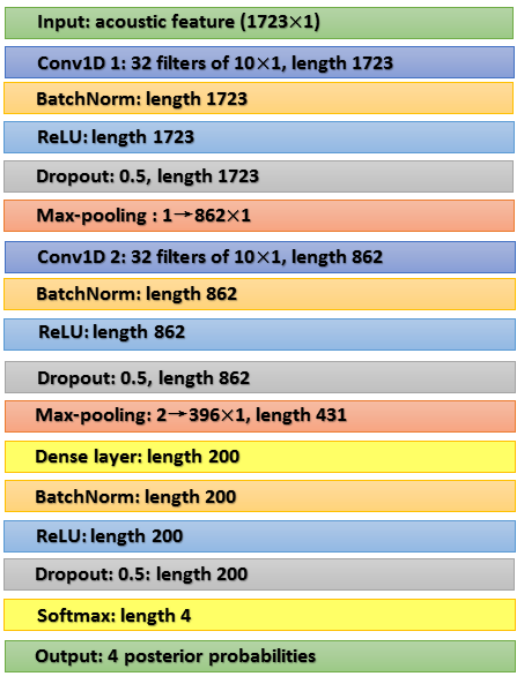

Figure Ⅰ. Topology of CNN

B. Results

The experiment results are shown in Table Ⅲ. In our four basic emotion recognition, our proposed method to improve speech emotion recognition with acoustic and phonological features using deep learning method (i.e., convolutional neural network) provides 60.22% of UAR which becomes the state-of-the-art on the utterance level speech emotion recognition and achieves 3.1% of improvement compared with the same classifier with acoustic feature only. The best UAR of the same database is 60.89% on frame level using CNN [12] and 58.46% on utterance level with hierarchical binary decision tree with speaker-dependent normalization [11]. The greatest improvement on four emotion recognition is 7.15% of UAR with multi-layer perception.

In general, the performance of the SVM is the worst. The performance of the deep learning method (i.e., convolutional neural network) is the best and outperformed significantly the other classifiers on four basic emotions recognition.

The improvement of speech emotion recognition by adding phonological features means that expertise knowledge can help machines improve their recognition rate because it is close to human perception and this complementary information is from thousands of years of humans' summarization which is more abstract but more discriminative and useful.

TABLE Ⅲ. THE RESULT OF EXPERIMENT.

Classifiers	**Features**	
	Acoustic	***Acoustic and Phonological Representations***
	UAR (%)	UAR (%)
SVM	49.91	51.35
LR	55.18	57.33
MLP	51.40	58.55
CNN	57.12	**60.22**

V. DISCUSSION AND CONCLUSION

From the result we can see that our proposed method to emotion recognition reaches the state-of-the-art using deep learning method of CNN by adding phonological representations. Phonological features represent people's knowledge summarization about prosodic representations and proves to be correlative to emotion. Therefore, adding expert knowledge to speech emotion recognition could further improve SER and shed light on the perceptual relationship between emotions and phonology. Our work presents evidence that discrete phonological representations have the potential to inform future feature development for emotion recognition and can lead to overall improved performance.

However, the limitation of this study is that the number of efficient phonological features is not rich. The reasons might be that the open source code to automatically recognize phonological representations is not so complete and sometimes it cannot recognize some very salient and evident phonological representations.

In the future, we will try to analyze which kinds of phonological features have discriminative power to specific emotions. We will also explore the cross-language, cross-culture and cross-humans speech emotion recognition to improve the SER' generalization.

ACKNOWLEDGMENT

This study is supported by the National Social Science Foundation of China (BCA150054)

REFERENCES

[1] Jin, Q., Li, C., Chen, S., & Wu, H., Speech emotion recognition with acoustic and lexical features. 2015: p. 4749-4753.

[2] Banse, Rainer, and Klaus R. Scherer. "Acoustic profiles in vocal emotion expression." Journal of personality and social psychology 70.3 (1996): 614.

[3] Schuller, B., Batliner, A., Steidl, S., & Seppi, D., Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. Speech Communication, 2011. **53**(9): p. 1062-1087.

[4] Schuller, B., S. Steidl, and A. Batliner. The Interspeech 2009 Emotion Challenge. in INTERSPEECH 2009, Conference of the International Speech Communication Association. 2009.

[5] Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., & Truong, K. P., The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. IEEE Transactions on Affective Computing, 2016. 7(2): p. 190-202.

[6] Schuller, B., Steidl, S., Batliner, A., Schiel, F., & Krajewski, J. The INTERSPEECH 2011 Speaker State Challenge. in INTERSPEECH 2011, Conference of the International Speech Communication Association. 2011.

[7] Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., & Narayanan, S. S. The INTERSPEECH 2010 paralinguistic challenge. in INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September. 2010.

[8] Schuller, B. W., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, A., Burkhardt, F., ... & Mohammadi, G. The INTERSPEECH 2012 speaker trait challenge. in INTERSPEECH 2012, Conference of the International Speech Communication Association. 2012.

[9] Liscombe, J.J. Prosody and Speaker State: Paralinguistics, Pragmatics, and Proficiency. 2007.

[10] Cao, H., Benus, S., Gur, R. C., Verma, R., & Nenkova, A. Prosodic cues for emotion: analysis with discrete characterization of intonation. in Speech prosody. 2014.

[11] Rosenberg, A. AuToBI - A tool for automatic ToBI annotation. in INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September. 2010.

[12] Lee, C.C., et al., Emotion recognition using a hierarchical binary decision tree approach. Speech Communication, 2011. 53(9–10): p. 1162-1171.

[13] Fayek, H.M., M. Lech, and L. Cavedon, Evaluating deep learning architectures for Speech Emotion Recognition. Neural Networks, 2017.

[14] Mirsamadi, S., Barsoum, E., & Zhang, C. (2017, March). Automatic speech emotion recognition using recurrent neural networks with local attention. In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on (pp. 2227-2231). IEEE.

[15] Keren, G., & Schuller, B. (2016, July). Convolutional RNN: an enhanced model for extracting features from sequential data. In Neural Networks (IJCNN), 2016 International Joint Conference on (pp. 3412-3419). IEEE.

[16] Busso, C., S. Lee, and S. Narayanan, Analysis of Emotionally Salient Aspects of Fundamental Frequency for Emotion Detection. IEEE Transactions on Audio Speech & Language Processing, 2009. 17(4): p. 582-596.

[17] Iliev, A. I., Zhang, Y., & Scordilis, M. S. (2007, June). Spoken emotion classification using ToBI features and GMM. In Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on (pp. 495-498). IEEE.

[18] Mariooryad, S. and C. Busso, Exploring Cross-Modality Affective Reactions for Audiovisual Emotion Recognition. IEEE Transactions on Affective Computing, 2013. 4(2): p. 183-196.

[19] Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., ... & Narayanan, S., IEMOCAP: interactive emotional dyadic motion capture database. Language Resources and Evaluation, 2008. 42(4): p. 335.

[20] Eyben, F., Wöllmer, M., & Schuller, B. (2010, October). Opensmile: the munich versatile and fast open-source audio feature extractor. In Proceedings of the 18th ACM international conference on Multimedia (pp. 1459-1462). ACM.

[21] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., ... & Hirschberg, J. ToBI: A standard for labeling English prosody. in International Conference on Spoken Language Processing, Icslp 1992, Banff, Alberta, Canada, October. 1992.

[22] Glorot, X., A. Bordes, and Y. Bengio, Deep Sparse Rectifier Neural Networks, in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, G. Geoffrey, D. David, and D. Miroslav, Editors. 2011, PMLR: Proceedings of Machine Learning Research. p. 315--323.

[23] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

How An Optimized DBSCAN Implementation Reduces Execution Time And Memory Requirements For Large Data Sets

Ole Kristian Ekseth and Svein-Olaf Hvasshovd
Department of Computer Science (IDI)
NTNU
Trondheim, Norway
email: oekseth@gmail.com and sophus@ntnu.no

***Abstract*—In data-analysis the use of approximate cluster algorithms has received broad popularity. A popular cluster algorithm is the DBSCAN cluster algorithm. While a number of software libraries provide support for the latter, they provide poor performance when analysing high dimensional data. In this work we address this issue. We present a novel method and implementation which significantly boosts the performance of DBSCAN. The result is a software which reduce the memory consumption by $10^3 GB$ for large data sets while reducing the execution time by 600x+ (for important similarity metrics). This artilce presents a high-performance appraoch to identify answers to region based similarity queries. While our work is tuned towards the application of DBSCAN, our novel approach for high-performance filtering of pairwise similarity scores may be used in a number of cluster algorithms. Therefore, the proposed method and software manages to address issues which are known to hamper high dimensional data analysis.**

Keywords: ***clustering; similarity metrics; data analysis; performance.***

I. Introduction

A center piece in high productive research is the availability of accurate and fast data analysis. The authors of [1] observe how application of established cluster algorithms require accurate knowledge of cluster algorithms and their configuration. The works of [2], [3], [5]–[7] observe that the size in data sets outgrows the speed of computer software. To address the performance issues numerous algorithms are presented every year, eg, with respect to algorithm permutations of k-means [8], SLINK [9], MCL [10], etc. A strategy for performance improvement concerns the application of *K-D tree* [11], which is used to reduce the time cost of pairwise similarity metric computation. For many clustering algorithms the major time consumer is the task to compute pairwise similarity, *e.g.*, through application of the Euclidean pairwise similarity metric. Figure 1 measures the time cost of different strategies for computation of pairwise similarity. The growth curve in time consumption (Figure 1) implies that it is unfeasible to apply "Density-based spatial clustering of applications with noise" (*DBSCAN*) on large data sets, *i.e.*, when using established software.

The use of heuristics, such as *K-D tree*, has many weaknesses. The work of [12] observes that in the analysis of "high-dimensional data, indexing schemes such as k-d tree do not work well" [12]. This is due to difficulties in designing a splitting criteria for rows which correctly captures the geometry of a high dimensional space. To overcome this issue many alternative approaches are described, such as with respect to "k-means mini-batch" [13], "k-means++" [14], and "k-medoid". The results in Figure 4 demonstrate how the *Expectation–Maximization* (EM) based cluster algorithms of "k-means++" [13] and "k-means" [8] identify cluster predictions with poor accuracy for cases where the type of data normalization and similarity metric is inaccurate. (While Figure 4 focuses on prediction accuracy on synthetic data sets, the complete set of measurement results may be generated through the scripts described in sub section V-C.)

While iterative cluster algorithms such as "k-means" permutations apply strategies for centrality to capture core traits in networks, there are alternative strategies which are faster and (for some use cases) more accurate, *e.g.*, with respect to the

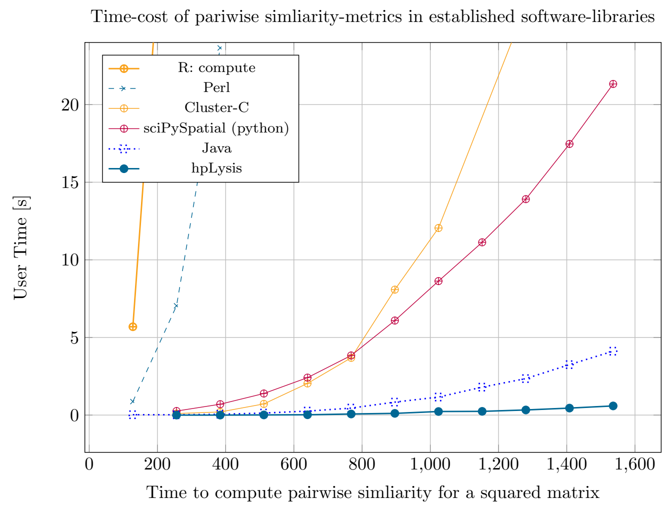

Fig. 1. The time cost of pairwise similarity metrics. The above figure captures the time cost of different strategies for computation of the Euclidean pairwise similarity metric. We observe how an naive python implementation causes a 600x+ execution timeperformance overhead.

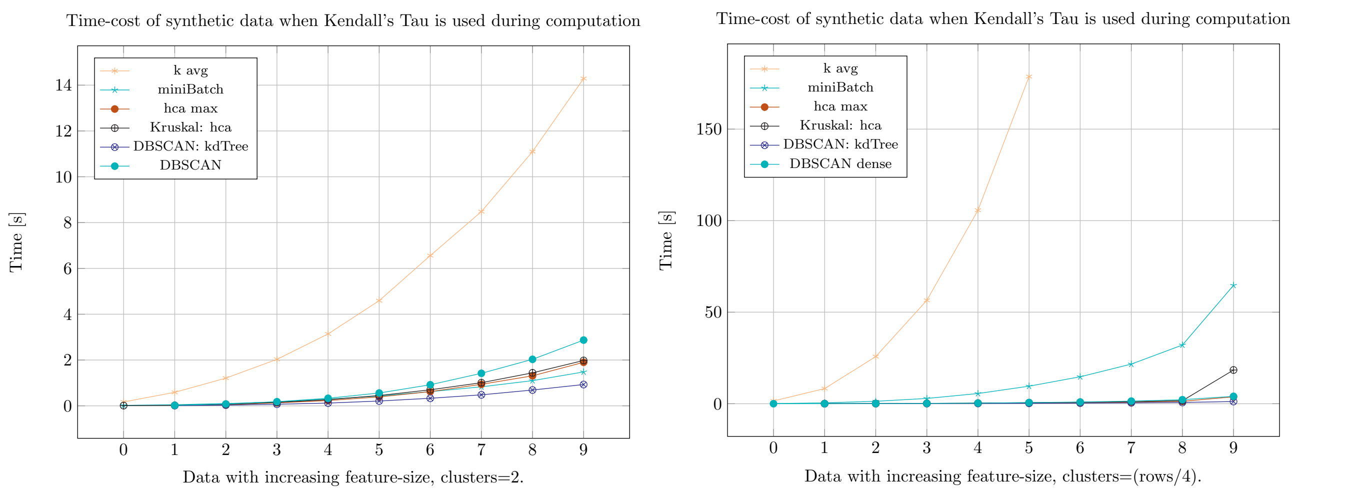

Fig. 2. Execution time between different cluster algorithms. The above figure measures the execution time on synthetic data sets with well defined clusters, thereby avoiding the predictions from being limited to the evaluated real life data sets. From the figure it is clear that the relative performance improvements of *DBSCAN* increases with increased feature size and cluster count.

DBSCAN algorithm [15]. The popularity of *DBSCAN* stems from its low execution time (Figure 2) and accurate predictions (Figure 4). The challenge in application of *DBSCAN* concerns its high execution time for data sets with many features [12]. An approach which addresses the latter may therefore increase the applicability of *DBSCAN*, hence improving a number of efforts in data mining. In brief a major challenge of the *DBSCAN* cluster algorithm concerns its high execution time and inaccurate cluster prediction results.

Application of our high performance machine learning library *hpLysis* [16] offers the ability to address the latter issue. The hpLysis software provides support for 320+ pairwise similarity metrics and 20+ unsupervised cluster algorithms. However, its high memory consumption makes it unfeasible for data analysis on large data sets.

In this work we unify the expressiveness of the hpLysis software with the requirements of large scale data analysis:

1) time and memory: design a software which reduces the execution time by 600x+ (Figure 1) while reduces the memory consumption by a factor of 10^7x (Section III);
2) prediction quality: support accurate similarity metrics, hence improving prediction accuracy by 100x+ (sub section V-B);
3) applicability: a new library for high performance inference of filtered similarity metrics from a dense evaluation of pairwise similarity metrics (sub section V-C).

The remainder of the paper is organized as follows. Section II briefly surveys related approaches, and Section III describes the approach. Section IV describes an approach to evaluate the influence of the data mining strategy proposed in this paper, a method which is applied in the result Section V. This paper ends with a brief summary of observations in Section VI.

II. Related Work

An evaluation of current approaches for improving prediction accuracy and execution time of *DBSCAN* cluster

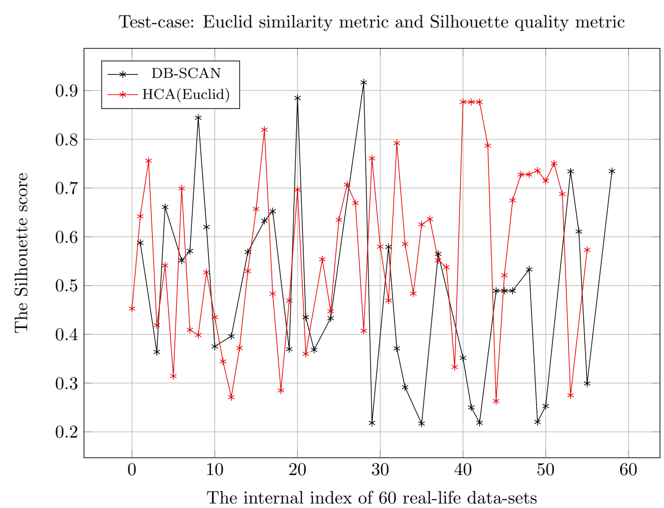

Fig. 3. A quality comparison of "SLINK" [9] and *DBSCAN*. The above figure describes the prediction accuracy of cluster algorithms on 60+ real life data sets found in the *real-kt* data set [16]. In the evaluation of prediction accuracy the "Silhouette Index" [17] is used.

algorithms [15] reveals issues in:

1) prediction quality: while software for *DBSCAN* and pairwise similarity metrics are focused on similarity metrics (sub section V-B), existing approaches are limited to Minkowski based metrics;
2) time and memory: software for data mining does not apply low level performance tuning strategies (Figure 1).

In below we identify key approaches in software and algorithms for *DBSCAN*. The importance of efficiently supporting the 320+ established similarity metrics stems from the purpose of clustering, *i.e.*, to accurately identify groups of entities. An example concerns the comparison of features which use different scales, an use case where Minkowski based metrics

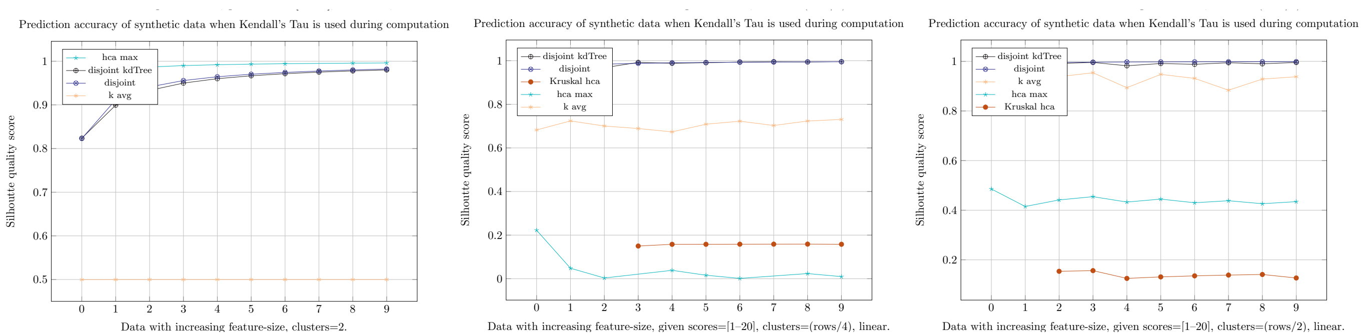

Fig. 4. The quality of cluster algorithms. The above figure evaluates the "Silhouette" [17] prediction accuracy for the cluster algorithms described in Figure 2. Each of the sub figures describe different cluster topologies.

fail.

A. Prediction quality

DBSCAN is asserted to provide a high degree of prediction quality in minimal execution time [18], [19]. A weakness in review articles such as [18], [22] concerns their lack of discussion with respect to the different *DBSCAN* implementations. In *DBSCAN* there are two different approaches for optimization of algorithm: to either use a *K-D tree* approach [11] in *DBSCAN* [3], [20] or to access the relationships from a matrix of scores, *e.g.*, as seen in the work of [21]. "*K-D tree* nearest neighbour (kdNN) [etc.,] has many characteristics like: simple, fast, and provide completely balanced tree [etc.,] has several significant drawbacks such as: need extreme search, time consuming, and impulsively divide points into two equal parts which may miss out on data structure" [22]. While [19] evaluates the performance of *DBSCAN* [22], they do not consider the implication of replacing *K-D tree* with a dense similarity metric computation, a view which is in contrast to [12].

B. Execution Time and Memory Consumption

For *DBSCAN* software implementations concerned with supporting high dimensional data analysis, the main time cost is associated with the computation of similarity metrics (Figure 1). To address this issue *K-D tree* [11] is used, a strategy which reduces the execution time. While the time complexity of a matrix based *DBSCAN* approach is of $O(n^2 f)$ (where n is the number of vertices in the graph and f is the features), a *K-D tree* approach has a time complexity of $O(n * log(n))$ [11]. The inaccurate cluster prediction accuracy of *K-D tree* application [22] motivates the computation of a *dense* similarity matrix: to apply filters during *run time*, hence reducing the memory consumption from n^2 to $n * k$, where k denotes the maximum number of adjacency vertices to be stored for each vertex. Established software does not provide support for the latter, *e.g.*, with respect to the "scikit-learn" software library [23]. A different aspect concerns the application of parallel hardware to reduce the time cost of computations, eg, through application of GPUs in [21]. In [21] the authors assert that they manage a 100x performance improvement when compared to a CPU based implementation. The measurements presented in sub section V-B identifies how the accurate choice of similarity metrics is critical for prediction accuracy, hence the limited applicability of their approach. A different aspect concerns their comparison basis, where Figure 1 demonstrates how a poor implemented C/C++ implementation results in a 30x+ performance-overhead, while the use of a naive R implementation introduce a 3000x+ performance penalty.

III. METHOD: RUN TIME FILTERING AND HARDWARE CLOSE OPTIMIZATION

The optimized implementation of *DBSCAN*, described in this paper, focuses on addressing the performance of dense similarity metric computation. The idea is to update the computation of each of the 320+ pairwise similarity metrics, supported by the *hpLysis* machine learning library [16], through introduction of run time filtering. This Section focuses on how to update the 320+ established metrics without introducing performance penalties to the hardware close optimization strategies applied by the *hpLysis* software: to compute similarity between $O(n^2)$ pairs of measurements without using $10^{12}B = 10^3 GB$ for a data set with 10^6 rows.

In order to provide support for computation of dense pairwise similarities without requiring the storage of all pairwise similarities, the *hpLysis* is updated with a new data structure and related logic. The result is a performance optimized approach which store $n * k$ relationships instead of n^2 relationships, where n represents the number of feature rows, while k denotes the user specified number of best performing vertices to remember. In the high performance computation of similarity metrics *hpLysis* makes use of the optimization strategy described in [24].

To address the requirement of result filtering during computation of pairwise similarity metrics, *hpLysis* is updated with new logics to post process each *matrix tile*. A *matrix tile* represents an $matrix[i...(i+b)][m...(m+b)] \in matrix$ where i is a row index in the matrix, m is a column index (in the matrix) and b is the block size. Each of the evaluated *matrix tiles* are merged after the tiles are computed. This strategy translates into a significant reduction in memory consumption without introducing penalty to execution time nor parallelism:

1) arithmetic: the time cost of similarity metrics regards the computation of each *matrix tile*, hence the introduction of post processing steps (separately for each *matrix tile*) does not result in a noteworthy increase in execution time;
2) parallelism: each thread computes separately for a block of row combinations, hence the merging of *matrix tiles* does not include any performance overhead with respect to communication nor memory delays;
3) memory consumption: for a matrix with 10^9 rows and where only the 10 best row similarities are of interest (for each vertex) the proposed approach enables a 10^8x reduction in memory requirement (when compared to the default approach), *i.e.*, from $10^{18}B = 10^9GB$ to $10^{10}B = 10GB$.

Therefore, the proposed approach enables accurate and efficient evaluation of large scale data sets. A complexity in the *matrix tile* merge step concerns the use cases where a *ranked step* is to be applied, *i.e.*, where each row is to hold the best performing relationships. In the merging of *matrix tiles* each row is evaluated separately. For the *ranked step* sorting is required, for which the "quick-sort" algorithm [25] is used. Hence, the time cost of a *ranked step* in matrix filtering is $O(n * log(n))$, where n is the number of vertices. The latter time cost is insignificant when compared to the overall time cost of dense similarity computation of $O(n^2 f)$, where f is the number of column features. Therefore, the proposed merging procedure will not increase the overall execution time of similarity metric computation.

The *parallel computation* support is enabled through a new compile time added parameter to the *hpLysis* software. For the parallel computation strategy to produce correct results, a requirement has to ensure that the row identities of a given matrix block is separately handled by each computer thread (*i.e.*, not shared by multiple threads). In our implementation of the parallel thread scheduler, the scheduler delegates tile matrix blocks $matrix[i....m][0...|cols|]$ to a given computer thread, hence the parallel scheduler avoids the need for a *writing lock* when updating the sparse result containers.

IV. Experimental Setup

This article seeks to improve data mining through provision for a fast *DBSCAN* implementation in support of 320+ pairwise similarity metrics.

A. Execution Time and Prediction accuracy

To evaluate differences in execution time the following strategies are applied:

1) similarity metrics: through application of established strategies to capture the isolated time cost of similarity metric computations;
2) cluster algorithms: evaluate the 20+ cluster algorithms supported in the *hpLysis* software, thereby identifying the benefits of the proposed approach in application of data mining;
3) data topology: how data sets capturing different topological use cases are influenced by the approach, for which the prediction accuracy is evaluated.

In order to capture the benefit of the large number of published software we relate software implementation to execution time, results which for brevity are summarized in Section V.

B. Prediction quality: Unsupervised parameter estimation to avoid bias in comparison of prediction accuracy

A challenge in the evaluation of large data sets concerns the difficulty in manually identifying correct parameter configurations. The severity is further increased with respect to the importance of choosing a dedicated similarity metric, and the types data optimization strategies to apply. "The performance of the JarvisPatrick algorithm and DBSCAN depend on two parameters: neighborhood size in terms of distance, and the minimum number of points in a neighborhood for its inclusion in a cluster" [26]. In order to compare unsupervised cluster algorithms on large data ensembles without introducing bias, the following requirements need to be satisfied:

1) similarity metrics: the *hpLysis* library is used to evaluate the prediction influence of similarity metrics and strategies for data normalization;
2) algorithm configurations: iteratively explore/evaluate numerous cluster parameters for *DBSCAN* and "k-means";
3) prediction quality: different metrics for external cluster validation is used, *e.g.*, with respect to "SSE" [8], "Dunn's Index" [27], "Silhouette" [17], and "VRC" [28], etc.

The above list identify strategies which addresses issues known to introduce bias in evaluation approaches for cluster algorithms. In the identification of 'k' clusters in EM based cluster algorithms, *e.g.*, k-means, "SLINK" [9] is used as an initial pre step. The "SLINK" result is then partitioned into disjoint clusters for different 'k' counts, and where the 'k' with the best cluster validity score (*e.g.*, measured through "Silhouette" [17]) is selected.

V. Result: Empirical evaluation

The proposed strategy to optimize *DBSCAN* manages to address issues in execution time and prediction accuracy of data mining efforts. The figures included in this paper exemplifies how the following requirements are handled in our approach:

1) time and memory: the design of a new approach for run time filtering during dense similarity metric computation (Section III) decrease the performance discrepancy between *off the shelf computer hardware* versus *cluster algorithms*;
2) prediction quality: how application of the 320+ supported similarity metrics address issues in prediction accuracy of *DBSCAN*.

 ISBN: 978-1-61208-612-5

A. Time cost and Memory consumption

To evaluate the influence of different implementation strategies and data topologies, we measure the difference in execution time between alternative strategies. Figure 1 captures the implication of different approaches for implementing similarity metrics. To investigate the applicability of *DBSCAN* in data analysis 20+ cluster algorithms [16] are applied to 100+ real life data sets and 320+ pairwise similarity metrics, where a subset of the measurements are included in Figure 2. The improvement in both prediction accuracy and execution time, as identified in this paper, is due to the low level implementation strategies which we apply. Section III describes how the proposed software implementation manages to reduce the memory requirements from n^2 to $n*k$, where n is the number of input rows, and k is the number of best scoring pairs to use. Therefore, the software described in this paper enables an n/k reduction in memory requirements. For $k = 2$ our approach implies a reduction factor of $10^9/10^2 = 10^7$ with respect to the memory requirement.

Popular/established/optimized software implementations provide support for only a limited number of similarity metrics, *e.g.*, where “sci-kit-learn” [23] is designed only for Minkowski based metrics. Hence, users are required to write their own support for similarity metrics, which for “R” introduce a 3000x+ performance penalty (Figure 1). The measurements presented in this paper demonstrates how the combination of high performance *DBSCAN* implementation and accurate similarity metrics reduces both the execution time and increases the prediction accuracy (Figure 3). While Figure 3 presents a subset of the measurements on the *real-kt* data set [16] data set, Figure 4 measures the prediction accuracy on synthetic data sets with well defined clusters, measurements which agree in the prediction accuracy of *DBSCAN*.

B. Prediction difference: DBSCAN compared to other unsupervised cluster algorithms

The related work Section II exemplifies how the execution time of *DBSCAN* is highly reliant on the topology of micro benchmark data, an observation verified in Figure 3. The findings presented in this paper represents a small subset of the observations. An example concerns the influence of pairwise similarity metrics in data analysis. The measurement scripts in sub-section V-C identifies how accurate choice of pairwise similarity metrics improves prediction accuracy of more than 100x. Hence, the accurate choice of similarity metrics relates directly to an increase in the prediction accuracy. To ensure representativeness of the described prediction differences, both measures for *internal metrics* (*e.g.*, “Silhouette”) and *external metrics* (eg,“Rand’s Index” [29] and “Adjusted Rand’s Index (ARI)” [30]) have been evaluated. In brief they provide consistent prediction results.

When discussing prediction accuracy this paper focuses on *internal metrics*, an approach which is designed to avoid certain algorithms from gaining a favorable bias during the construction of micro benchmarks. To exemplify the latter, an evaluation of the well known IRIS flower data set (with three established cluster partitions) with the MINE similarity metric [31] reveals how there are only two clusters present, while Euclidean clearly identifies 2.3 clusters in the IRIS flower data set. While the latter result may indicate that Euclidean provides a better closeness to the gold standard, an investigation into the feature similarities reveal patterns in the features which clearly supports the MINE assumption (of two clusters in the IRIS data set), *i.e.*, for which we have observed how the application of *external metrics* may present results which diverge from visual perspectives of cluster similarity.

C. Reproducibility

The proposed software is implemented in the *hpLysis* software, and made accessible through:

1) *DBSCAN*: integrated in the *hpLysis_api.h*;
2) sparse similarity: accessible from the *hp_api_sim_dense.h* API, an interface for inferring sparse sets of similarity metrics from a dense evaluation of pairwise similarity metrics;
3) terminal interface: *x_hp_sim_dense* enables computation of sparse similarity and sparse *DBSCAN*,

The above interface provides access the high performance optimization strategy described in Section III. The implementation level details of the result merging is found in the *insertBlockOf_scores__sorted__kt_list_2d(..)* function (*kt_list_2d.h*, where latter file is found in the *hpLysis* repository).

To enable validation and extension of the performance measurements included in this paper, a number of new use cases have been included into *hpLysis*, such as:

1) effect of pairwise similarity metrics: *tut_sim_15_manMany_Euclid__selectBestInsideMetricComp.c*;
2) accuracy of cluster algorithms: *tut_clust_dynamicK_7_find_nCluster_multipleData_allSim_xmtPreFiltering.c*;
3) execution time measurements: *tut_time_1_data_syntetic_wellDefinedClusters.c* and *tut_time_2_clustAlg_syntAndReal.c*.

The above measurement scripts makes use of the synthetic data sets constructed from *hp_clusterShapes.h* in combination with the data ensembles of *real-kt*, *kt-mine*, *shapes* (located in the hpLysis repository).

VI. Conclusion and Future Work

This paper has presented a new approach to compute *DBSCAN* on large data sets, both with respect to sparse data sets, dense matrices, and dense matrices with missing values/scores. The results which are presented demonstrate how the application and support of 320+ similarity metrics enables significant performance boosts to cluster algorithms. Through application of a approach for run time filtered computation of dense similarity matrix, this paper manages to address issues known to hamper fast and accurate analysis of many dimensional data sets, as described in Section III. While our proposed approach manages to reduce execution time by a factor of 600x+ (Figure 1), the memory consumption is reduced by a factor of 10^7x.

This paper demonstrates how the application of high performance implementation strategies enables a boost in prediction

 ISBN: 978-1-61208-612-5

accuracy and execution time. The combination of results from Figure 1 and Figure 2 reveals how the application of accurate similarity metrics combined with low level implementation may significantly improve prediction accuracy of data mining. In the measurement Section V we have demonstrated how accurate parameter identification of *DBSCAN* settings and similarity metrics enables *DBSCAN* to predict clusters which a high degree of prediction accuracy, *i.e.*, when compared to default approaches. Therefore, *DBSCAN* offers the potential for accurate identification of clusters.

In this work we have addressed a shortfall with respect to established approaches. To enable accurate computation of similarity without being constrained by memory thresholds and unfeasible execution time of current approaches. Therefore, we assert that the software and methodology for accurate and high performance data mining, presented in this paper, may improve a number of established approaches in data analysis.

A. Future Work

A challenge in application of *DBSCAN* concerns the need to identify accurate configuration thresholds. We plan to address the weakness of automated parameter identification in *DBSCAN*, *i.e.*, to provide a methodology and software for accurate and fast evaluation of threshold parameters.

Acknowledgements

The authors would like to thank MD K.I. Ekseth at UIO, Dr. O.V. Solberg at SINTEF, Dr. S.A. Aase at GE Healthcare, MD B.H. Helleberg at NTNU–medical, Dr. Y. Dahl, Dr. T. Aalberg, Dr. J.C. Meyer, and K.T. Dragland at NTNU, and the High Performance Computing Group at NTNU for their support.

References

[1] Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. Science Advances **3**(5), 1602548 (2017)

[2] Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. Acm Sigkdd Explorations Newsletter **6**(1), 90–105 (2004)

[3] Narayanan, R., Ozisikyilmaz, B., Zambreno, J., Memik, G., Choudhary, A.: Minebench: A benchmark suite for data mining workloads. In: Workload Characterization, 2006 IEEE International Symposium On, pp. 182–188 (2006). IEEE

[4] Consortium, U., *et al.*: Uniprot: the universal protein knowledgebase. Nucleic acids research **45**(D1), 158–169 (2017)

[5] Mekkat, V., Natarajan, R., Hsu, W.-C., Zhai, A.: Performance characterization of data mining benchmarks. In: Proceedings of the 2010 Workshop on Interaction Between Compilers and Computer Architecture, p. 11 (2010). ACM

[6] Asanovic, K., Bodik, R., Catanzaro, B.C., Gebis, J.J., Husbands, P., Keutzer, K., Patterson, D.A., Plishker, W.L., Shalf, J., Williams, S.W., et al.: The landscape of parallel computing research: A view from berkeley. Technical report, Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley (2006)

[7] Lu, S.-L., Karnik, T., Srinivasa, G., Chao, K.-Y., Carmean, D., Held, J.: Scaling the x201C memory wall x201D designer track. In: Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference On, pp. 271–272 (2012)

[8] Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

[9] Sibson, R.: Slink: an optimally efficient algorithm for the single-link cluster method. The computer journal **16**(1), 30–34 (1973)

[10] Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res **40**, 1575–1584 (2002)

[11] Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software (TOMS) **3**(3), 209–226 (1977)

[12] Hamerly, G.: Making k-means even faster. In: Proceedings of the 2010 SIAM International Conference on Data Mining, pp. 130–140 (2010). SIAM

[13] Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1177–1178 (2010). ACM

[14] Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007). Society for Industrial and Applied Mathematics

[15] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al.*: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)

[16] Ole Kristian Ekseth: hpLysis: a high-performance software-library for big-data machine-learning. https://bitbucket.org/oekseth/hplysis-cluster-analysis-software/. Online; accessed 06. June 2017

[17] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)

[18] Shah, G.H., Bhensdadia, C., Ganatra, A.P.: An empirical evaluation of density-based clustering techniques. International Journal of Soft Computing and Engineering (IJSCE) ISSN **22312307**, 216–223 (2012)

[19] Kockara, S., Mete, M., Chen, B., Aydin, K.: Analysis of density based and fuzzy c-means clustering methods on lesion border extraction in dermoscopy images. BMC bioinformatics **11**(6), 26 (2010)

[20] Michael Hahsler: A C-package-library for efficient DB-SCAN computation. https://github.com/mhahsler/dbscan. Online; accessed 06. June 2017

[21] Andrade, G., Ramos, G., Madeira, D., Sachetto, R., Ferreira, R., Rocha, L.: G-dbscan: A gpu accelerated algorithm for density-based clustering. Procedia Computer Science **18**, 369–378 (2013)

[22] Otair, M.: Approximate k-nearest neighbour based spatial clustering using kd tree. International Journal of Database Management Systems **5**(1), 97 (2013)

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**(Oct), 2825–2830 (2011)

[24] Drepper, U.: What every programmer should know about memory. Red Hat, Inc **11**, 2007 (2007)

[25] Hoare, C.A.: Quicksort. The Computer Journal **5**(1), 10–16 (1962)

[26] Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters **31**(8), 651–666 (2010)

[27] Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics **4**(1), 95–104 (1974)

[28] Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)

[29] Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336), 846–850 (1971)

[30] Yeung, K.Y., Ruzzo, W.L.: Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics **17**(9), 763–774 (2001)

[31] Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large data sets. science **334**(6062), 1518–1524 (2011)

Finding and Implementing Patterns for Creative Spaces

Christian Kohls
Department of Computer Science
TH Köln
Gummersbach, Germany
e-mail: christian.kohls@th-koeln.de

***Abstract*— Ideation and design thinking are important skills that should be acquired in higher education to prepare students for modern work requirements. We have implemented several workspaces at our university that foster creative work by providing tools, materials and an inspiring atmosphere. To make the spaces fit to the goals of creative thinking we have identified good practices as design patterns. The patterns are based on existing solutions and have led and justified our design decisions. In this article we will discuss the benefits of the pattern approach to find meaningful solutions. The mining process will be outlined and some of the resulting patterns are presented. Finally, we will reflect how the patterns have helped us to achieve our goals. The patterns can be used by other universities and companies to create similar spaces as each pattern describes the solution in a generative way.**

Keywords-design patterns; innovation; creative space

I. INTRODUCTION

We planned to create several spaces for innovate work at our campus. The motivation was to increase the creative potential in student projects. Thus, the space should be enabling in a sense that it provides all the tools for design thinking, including ideation tools, building materials for rapid prototyping, ways to investigate the problem domain and test ideas as they are developed. The appropriate methods should be supported by both classic and digital tools. Moreover, students should learn the methods while they are using them. Creative spaces should be both enabling and educating. In addition, creative spaces should be efficient, i.e., no setup time, no boring meetings where only a few dominant persons speak, and low thresholds to use tools and technology that make the creative process more effective. The rooms should also be exciting and help students to experience their university as a place of innovation. The spaces should promote project-based and skill-driven learning, establishing a result-oriented "maker culture". Our goal is to create four different spaces, each addressing different activities: an innovation space for generating ideas, a thinking space for retreat and reflection, a maker space for prototyping and testing, and a planning space for managing projects. The need for different spaces for the various activities is based on the pattern Ecosystem of Workspaces [1], see Figure 1, and the design principles for creative spaces [2].

Our means to get there is to do identify existing patterns and develop new patterns for creative space. We use the design pattern approach to generalize and analyze existing good solutions.

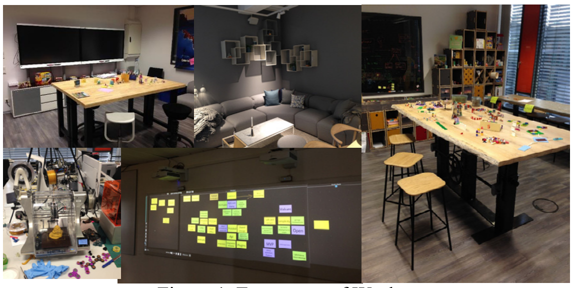

Figure 1. Ecosystem of Workspaces

The rest of this paper is organized as follows. Section II discusses related work, including the impact of creative space for university development, existing design frameworks and the design pattern approach. Section III describes the mining process along with example patterns. Section IV reflects on the lessons learnt and how one can test the patterns both qualitatively and quantitatively. Section V draws conclusions and outlines future work.

II. RELATED WORK

In this section we explore the design of creative spaces in universities and compare frameworks and design patterns as planning tools.

A. *Impact of creative space for university development*

The importance of interior design in academic teaching contexts has been highlighted recently. The Horizon Report [3] has picked up this topic in recent years, again and again, under the themes "Maker Space," "Bring your own device," "Flipped Classroom," "Internet of Things," "Wearables," and "Redesigning learning spaces". An approach to these concepts should be of strategic importance. A commission of experts of the "Hochschulforum Digitalisierung" (Panel for digitization in higher education) concludes that "maker spaces and creative spaces" and "digital collaboration tools" are among the key technologies of a digital university.

B. *Existing room concepts and design frameworks*

There are several frameworks that deal with the design of innovative spaces and spaces that blend classic and digital tools. For example, the TACIT framework is derived directly from a blended space approach [4] by identifying

 ISBN: 978-1-61208-612-5

key elements, such as territoriality, awareness, control and interaction. Another framework for understanding the role of the physical environment in innovation focusses on the related goals of such spaces [5]. The framework distinguishes between strategic and symbolic goals, innovation efficiency and effectivity, enhanced teamwork, stronger involvement of stakeholders and expanding entrepreneurial skills on a general level. It operationalizes these objectives and differentiates between factors for the planning of the space as well as factors for the evaluation of the degree of attainment. There is also a conceptual framework for spaces to foster and sustain innovation that compares the criteria of different frameworks [6]. All of these frameworks have in common that they name important principles, criteria and directives for planning and testing. However, when it comes to the concrete design of a new space, they remain very general and often too abstract. Therefore, in our view, the design pattern approach of Alexander et al. [7][8] seems to be a better alternative. It keeps the general reasoning but provides tangible and holistic advice to implement solutions.

C. Design Patterns

The idea to capture balanced patterns of good design has its origin in the field of architecture. Christopher Alexander wanted to create a language which helps ordinary people to participate and express their wishes and demands in the design of towns, buildings and constructions [7][8]. The approach was adapted in many other fields, especially in the domains of software design [9] and software education (Pedagogical Pattern Project). There have been several patterns on education and e-learning [10]-[12]. According to Alexander, "each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice [8]." As such, patterns capture the essence of recurrent good solutions. They describe both the solution form and the process to generate this form. Likewise, "an e-learning pattern captures the regularities of good practices in order to reuse the proven methods, scenarios and content forms in new contexts addressing new design tasks. The core idea is to not reinvent the wheel but to preserve what has been successful in the past. Beside the explicit description of good (successful) pedagogical methods, tools, media formats, resources, and scenarios design patterns reason about the adequate use of such solutions [13]." Patterns clearly state the design problem and design solution in a structured way and provide a "comprehensive set of design ideas [14]." They can be used to discuss different aspects of design expertise and provide a lingua franca for joint course design [15].

The reflective analysis of patterns requires that a description of patterns discusses how and why a solution works. A form or intervention is neither good nor bad in itself but produces different consequences depending on the context. Patterns are never seen in isolation but capture the environments in which they work, often by relating to other patterns. Patterns can describe specific actions to overcome existing seams. But more important, they explicitly state what the problem of the existing seam is. Problems are discussed by analyzing conflicting forces from different perspectives, i.e., social, technological, political, and economical factors. This integrates the views and requirements from different stakeholders and makes the relations between different levels explicit. For example, choosing a specific social media strategy can open new options as well as limiting the learner's choice. The curriculum can drive decisions on both the level of the learning organization as well as the individual teacher's/learner's behavior. Individual interests can conflict with organizational strategies but also support new ways of collaboration. There are already some pattern languages about creative collaboration, presentation and learning [16].

III. PATTERNS FOR CREATIVE SPACE

We will first describe the pattern mining process. Then, we will discuss the identified pattern categories and present some examples.

A. Pattern mining

Identifying and documenting design patterns is called pattern mining. There are different approaches for pattern mining [17]. The approach taken here is a typical mix of methods of inductive and deductive research:

- Deriving requirements and factors from frameworks
- Analysis of existing good practices for interior design
- Visits to tradeshows and exhibitions, browsing product brochures, explore existing rooms
- Testing of classic and digital tools
- Participatory design sessions with later users of the space (students, docents)
- Evaluation of different design options with mockups

In the process of identifying the general forms of the patterns, the findings from general frameworks can be considered as the driving factors. Moreover, we found recurring structures in case studies of existing facilities: i-Land [18], ICE [19], and iRoom [20]. Furthermore, numerous analog tools (e.g., methods cards, innovation games) and digital tools (e.g., iPads with pen, touch screens, tabletops) were tested in the laboratory or at exhibitions. Various design options for the room were played with, for example using mockups to explore the optimal size for a center table that should be used as a "stage" for the creative process. Various pattern candidates have emerged from this analysis. Pattern candidates that we can support empirically by several examples and that we can explain plausibly using establishes theories, have been selected to plan the rooms.

B. Pattern categories

The identified patterns can be characterized and clustered into categories of different problem groups: work materials and tools, atmosphere, session management, navigation and blended interaction. The first category is about the materials and tools that should be available in creative spaces. Since we cannot always predict which materials will be needed in a

 ISBN: 978-1-61208-612-5

creative session, an abundance of materials should be available to give users freedom of choice. Boxes with work materials are frequently found in design thinking contexts. The room should provide several THOUGHT TRIGGERS that provide impulses for ideas. Examples are random images, story cubes, and stimulus words both in physical and in digital form. For visualization, we provide A BUNCH OF PENS AND PAPERS as well as digital tablets and interactive whiteboards. CURIOUS THINGS are unusual objects to stimulate lateral thinking. Such objects can be spread around or hidden as a surprise element in containers (SURPRISE). Furthermore, various project templates (WORK TEMPLATES, e.g., Business Model Canvas) should be provided for analytical processes. The filled out templates can be digitized automatically and possibly trigger other actions (TEMPLATE TO ACTION), e.g., automatic sorting into various digital lists (such as pros and cons), or using drawings within other programs (games, living mockups). Figure 2 shows some example materials and tools.

Figure 2. Materials and tools

The atmosphere of the room also plays a special role in creative work. For this reason, we have identified many patterns in the physical space, such as ambience, plants, high-quality furniture, bar stools and workbenches, sofas, pictures and sound. QUALITY FURNITURE suggest to use high quality and extraordinary furniture. In a valuable environment, people feel more valued. The space should support different MOODS with illumination and sound. We also need a PLACE OF RETREAT, where participants can relax and let the mind wander (phase of incubation). To avoid distractions, the room should have NO CABLES, be SOUNDPROOF BY CHOICE and have only soft angles.

Session management requires solution patterns to save sessions, restore them and take them into other (physical) rooms without seams, i.e., the transfer is automatically supported by cloud based solutions that can show the same content in different rooms. A backup functionality must be available for data security. To retrieve lost data back, restore functionality must be given. The rollback enables the applications and systems to return to a prior state. This is often a standard feature in online tools such as Google Docs or Wikipedia. Nevertheless, we consider the explicit documentation to be important as a model for the design of new tools. ROOM RESET refers to a quick way to reset all systems and return them to their original state. This solution is very important to avoid revealing work results to the wrong eyes, i.e., if the next work group enters the room all prior writings should disappear. Room reset features are often found in software systems for interactive whiteboards (such as SMART Boards, Microsoft Surface Hub). If a group of users returns to the room a week or a month later, they should continue to work where they had stopped. Hence, it must be possible to access a history to prior results. All devices in the room are restored to this state and provide the same set-up of the last event (ACCESS TO OLD SESSIONS). Since there are several independent innovation rooms, we need a way to take the complete session to another room. This is also relevant for break-out sessions. This leads to the pattern CROSS-ROOM INTERACTION.

The navigation space must also be considered in a creative room concept. There are different solutions that help users to find orientation inside the room - for example, some analogue and digital tools are self-explanatory. However, in a space that offers many possibilities, not everything can be grasped at first sight or is intuitively usable. For orientation and guidance, we can use TANGIBLE INSTRUCTIONS such as menu cards with brief instructions, overview maps or “cookbooks” that provide step-by-step guidance and highlight which building blocks are useful. Using QR codes, these physical elements can in turn call up video instructions for smartphones or wall displays. Icons of digital user interfaces should be fully self-explanatory. Users should see at first glance which action is trigged if a button is pressed. Objects in the room need to be placed in such a way that users are encouraged to use them. For example, Lego bricks and sticky notes should not be hidden in a cabinet. They should be accessible right away and offer a high affordance.

Perhaps the most interesting category is Blended Interaction, as it is concerned with solutions that enable a seamless integration of the physical and digital world. The pattern PHYSICAL TO DIGITAL regards physical actions and artefacts that can be digitized immediately. When a user sketches with a pencil on a digital surface or a sheet of paper, it is to be digitized immediately by a touch-sensitive surface or a camera. The complementary pattern is DIGITAL TO REAL. Objects created in the digital space should not remain only digital but are brought in to the real world. Examples are projections onto physical objects, walls or on the floor, the printing with photo or 3D printers, as well as the controlling of robots or Raspberry Pi modules. Moreover, physical objects are used as simple triggers for interaction between the digital and physical world (PHYSICAL OBJECTS TRIGGER ACTIONS). An example is digitizing a sketch by pressing a physical button. Rooms that contain several digital devices need some means to connect them and share data (DEVICE ORCHESTRATION). Examples are: sending brainstorming items from smartphones to interactive walls, sketch characters or labyrinths on tablets and use them on game arcades, or selecting the content for large displays using small interactive tablets. One step further, we can orchestrate the connected devices to one large unit (COUPLED DEVICES). This way several independent devices can be connected to form a larger contiguous workspace. Examples are: dual projections of interactive whiteboards controlled by one computer, small info screens that connect to a large info screen, interconnected wall screens, tablets on a coffee table forming a temporary interactive tabletop.

C. Pattern examples

This section contains two detailed pattern examples and three short pattern overviews as additional examples.

1) Hybrid Space (Pattern)

The dichotomy of digital and non-digital artefacts is resolved in a hybrid space by seamlessly bridging different types of artifacts, making digital data touchable and graspable, enhancing physical objects with digital information and digitizing physical objects.

Context: We live in the real physical world, digital and connected devices are ubiquitous. Very often both worlds exist in their own space, and are only superficially interrelated. Things we can sense (feel, touch, see, hear, etc.) are more meaningful to us. We like to tinker with real things. Very often we either work in the physical world only or in the digital world only.

Problem: Working in the physical world limits the way of sharing and manipulating objects – the digital has much more to offer. However, the digital world limits the richness of interactions to a predefined set. Moreover, the digital world lacks the embodiment of actions, such as playing and tinkering with objects physically with your hands (or other body parts) and arranging them spatially in a natural way.

Forces: Ease of use. Interaction with objects is easier when they are real and tangible. Digital objects can be easily manipulated, edited, shared and distributed.

Representations. Objects of the real world often can be represented as virtual objects (for example a sticky-note can be a real piece of paper or a virtual object). Physical object of our world can be augmented with additional information and functions using digital tools. The mode of optimal representation depends on what you want to do with the object. Do you want to touch it? Move it in real physical space? Or do you want to share it across remote locations and work on it with remote participants? Very often both is needed.

Solution: Therefore, merge both physical and digital space into one hybrid space where real world objects can be digitized and represented in virtual space, and the digital space can be brought into the real world using tangible objects and devices.

Both the physical and digital worlds have their own strengths and weaknesses. To empower the best of both we need to enrich both worlds with each other seamlessly. Every smartphone is capable of digitizing physical objects by simply making a photo. Special apps can support the extraction of meaningful information: identifying single sticky notes, mapping whiteboards to images, recognizing objects and gestures. Document cameras can quickly scan 2D documents and 3D objects and send them to digital work spaces. Objects of the real world can also trigger the display of meaningful digital information. For example, a QR code or a specific layout on book pages or playing cards can start-up programs or blend in additional information. This additional information can appear on the smartphone screen (augmented reality) or on additional screens. On the other hand, the information of the digital world can be projected into the real world. Each smartphone becomes a window to the digital world. Projectors can display information on walls and even on objects. Physical objects can be enriched with small displays or other outputs (audio, motion) based on digital processes. Objects in the physical world are always at hand (you don't have to turn on a computer), and many actions are more natural in the real world. For example, feeling the surface and weight of objects gives you direct clues how to hand and work with them. Setting up new arrangements is fast, straight-forward and requires no training. Writing and drawing on paper still provides a richer experience than most digital tools offer. The interaction of surface properties, pen type, layers of color etc. is still a richer experience.

You may expect: The real and digital worlds merge. Benefits of the digital world are mapped to real world objects. Real world objects can be represented seamlessly in the digital world, thus making sharing and manipulation easier. Information can be mapped to real world objects, allowing rich learning experiences. Tinkering with objects – real and digital – is encouraged and creative learning is more likely to happen.

However: To make the transfer between both worlds seamless, a lot of hardware and software is needed. Thus, the experience is often bound to special rooms. It requires a lot of knowledge about methods and technology to unleash the full potential. While playing with things is a good and encouraging experience, it can also lead to using the technology for the sake of using technology.

Examples: Document cameras to scan objects and show them on an interactive whiteboard, apps to photograph sticky notes and bring them to the digital world, 3D printers to make virtual objects tangible, digital twins (computerized companions of physical assets), or putting cards on a scanner and map the content to virtual objects.

2) Orchestrate digital devices (Pattern)

Context: Each device has different strengths. If you switch between the appropriate tools you can use the benefits of each of them. Some are good for drawing, others for showing or interacting. In a room with several devices, users should always choose the tool that fits best to the task. Since tasks often lead to different activities, users may want to switch digital tools seamlessly.

Problem: Using different tools in isolation may offer the best of each world. However, it does not offer new ways of collaboration. It limits collaboration, because switching from one tool to another interrupts the flow of working and thinking.

Forces: There are two types of devices, the ones which are used individually and the ones that are used together. Devices offer different functions, haptic characteristics, etc. Tools which are ideal for inputs aren't equally ideal for outputs and the other way around. At the one hand a large display is very hard to control by touch or even with the mouse pointer, at the other hand a small display is not suitable or appropriate to present information to groups of people. Each device has its own affordances. Which device is the best, may change in the work process. Then it's important to switch between the devices seamlessly without losing the work results that have been created to that point.

Solution: Orchestrate digital devices in such a way that you can seamlessly use and transfer objects between them. That way users can freely choose where to display, create, edit and use objects. This choice can depend on individual preferences or different modes of usage.

You can connect different devices using the patterns for ad-hoc connection. To share data between devices you have to run them either in a local WIFI or connect them via a cloud service. The devices need a shared (and protected) data storage, e.g., a place where one device can upload files such as images, and another device can download the files. To notify other devices about new available data, the devices need to observe this shared file space or listen to events. One implementation option is a central event server that distributes events and notification. One tool can upload a file and notify the event server. For example, someone could draw a picture on an interactive whiteboard and save it to a specific location on the file system. A local process observes the folder and uploads the drawing. It also notifies other devices that a new drawing is available. Tools that are interested in this kind of data can automatically download the file and use it for their own purposes (e.g., use the drawing as a sprite in a game or a new screen in app mock-up).

You may expect: Open interfaces enable device orchestration and inter-device communication.

However: The devices have to use the same application or a compatible software to be able to orchestrate. Not all hardware and software vendors provide open APIs for seamless sharing, and often a hack is needed.

Examples: Users draw app mock-ups on a large digital whiteboard that allows to edit several screens at the same time. The sketched screens can be compiled to a Living Mockup that runs on tablets or smartphones instantly. Tablets can be used as interactive wall pictures. They can offer different actions or methods. Users choose a specific method on the wall tablet and large working templates are open on a digital whiteboard automatically. Users draw on a tablet and the drawn elements become game sprites on an arcade machine instantly.

3) *Thought Triggers (Pattern Overview)*

Problem: You need a push into a new direction.

Forces: We tend to step into the same direction. We might be preoccupied and biased by your known solutions. Old thought patterns need to be disrupted. There is a need for external source to disrupt your thought patterns.

Solution: Use textual or visual stimulus as thought triggers: Ask different questions, take in a new perspective, or restructure your problem.

4) *Sticky Notes (Pattern Overview)*

Problem: High effort to change structures makes exploration less likely.

Forces: Add ideas quickly, parallel. Move categories and concepts. Spatial arrangement implies meaning. Explore alternatives. Easy setup, use everywhere.

Solution: Use sticky-notes to write down your ideas.

This pattern description is a good example for a solution that is well known. However, the description makes the underlying problem and benefits explicit. It does not simply recommend sticky notes but explains why this tool works.

5) *Playfulness (Pattern Overview)*

Problem: Your thoughts are constrained by conventions and fear of breaking rules.

Forces: Positive feelings make you more creative and productive. Be less serious. Turn expectation upside down. Seek ambiguity, connect unrelated concepts, draw new analogies. Get rid of order, rules and assumptions. Allow exploration without consequences.

Solution: Put yourself and your team into a mood of fun and playfulness.

IV. LESSONS LEARNT IN USING THE PATTERNS

This section discusses how design patterns have helped to plan, create and use creative spaces. We also draw conclusions for the use of patterns in general.

A. Understanding of forces and consequences

During the planning process, the patterns helped us to check concrete technologies with regard to the properties we had required. The lower abstraction layer created by patterns helps to make concrete implementation decisions and to justify them. By discussing the consequences of a design pattern, not only the positive effects but also possible limitations become transparent. For example, there are some patterns that may have a negative impact on users' sense of privacy, such as automatic speech analysis and cameras in the room. Through the explicit naming of subsequent problems it is possible to search for further solutions.

B. Justification for design decisions

Patterns also help to convey the elements of the space better to address other stakeholders. The forces provide solid justifications for design decisions – even for unconventional elements. For example, the use of Lego bricks and innovation card games may confuse some people. The patterns, however, explain what has driven us to use these elements: their combinatory qualities to explore, the playfulness to challenge assumptions, and their inspiring potential.

C. Manage complexity

Furthermore, the patterns have helped break up a complex spatial composition into individual components that can be considered nearly independent while still fitting into the overall concept. The harmonious interaction is ensured by cross-references between the patterns.

D. Identification of deficits

To some extent design patterns also reveal technological deficits. The design pattern DEVICE ORCHESTRATION is indeed not an utopia but derived empirically from concrete examples of linking devices. Yet the interplay of all digital devices in the room is currently not fully implemented due to missing protocols and technical interfaces. Nevertheless, DEVICE ORCHESTRATION remains a valid solution. It is just that the implementation is more complex and cannot be implemented for all devices in the short term. For many devices and systems, it is a missing solution. Similar constraints as well as unexpected potentials have been uncovered by identifying the patterns.

E. Empirical support

Patterns that are not sufficiently empirically confirmed yet are considered as proto-patterns, that is proposals for a better future. However, most of the identified patterns are based on existing space solutions and usage concepts. Nevertheless, only future will show whether the rooms are accepted and used as we intended. The actual use will thus lead to a confirmation or falsification of the patterns inductively determined. So far we have observed that the rooms are highly frequented and students ask for more of such facilities. At the moment we are running a quantitative survey about the user experience. A questionnaire covers the activities, satisfaction and outcomes of a session. We are also measuring the use of tools and their impact on the design results qualitatively. Since most design projects require written reports about the process, we are able to analyze how the room was used. Moreover, many design sessions are led by facilitators who can observe how students explore and use the room. Based on these findings, the patterns will be refined and adapted.

V. Conclusions and Future Work

The identified design patterns have been implemented to a large extent for the innovation rooms we have designed. They have proved to be a very good planning tool. The design patterns are based on the observation, analysis and experimentation with existing solutions. They do not replace the mentioned frameworks, but make them a step more concrete and thus empirically verifiable. The frameworks also provided a theoretical foundation for the establishment of the solutions. Our next goal is to put all our patterns into an action-oriented booklet and a deck of cards. So, through experience it will be easier to close the gap of missing system components. New transformation projects will benefit, respectively. The patterns help not only in the concrete design process but provide arguments why certain interventions for a more creative work environment of the students are conducive.

References

[1] J. Quillien, “Clever Digs. How Workspaces Can Enable Thought,” Ames, IA: Culicidae Press, 2012.

[2] C. Kohlert, and S. Cooper, “Space for creative thinking: Design principles for work and learning environments,” Munich: Callwey, 2017.

[3] S. Adams Becker, M. Cummins, A. Davis, A. Freeman, C. Hall Giesinger, and V. Ananthanarayanan, “NMC Horizon Report: 2017 Higher Education Edition,“ Austin, Texas: The New Media Consortium, 2017.

[4] D. Benyon, and O. Mival, “Blended Spaces for Collaboration”, Computer Supported Cooperative Work (CSCW), vol. 24, no. 2-3, pp. 223-249, 2015.

[5] J. Moultrie, M. Nilsson, M. Dissel, U.E. Haner, S. Janssen, and R. van der Lugt, “Innovation spaces: Towards a framework for understanding the role of the physical environment in innovation,” Creativity and Innovation Management, vol. 16, no. 1, pp. 53-65, 2007.

[6] F.O. Bustamante, J.I.P. Reyes, M. Camargo, and L. DuPont, “Spaces to foster and sustain innovation: Towards a conceptual framework,” Engineering, Technology and Innovation/International Technology Management Conference (ICE/ITMC), IEEE, p. 1-7, 2015.

[7] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, “A pattern language: towns, buildings, construction,” New York: Oxford University Press, 1977.

[8] C. Alexander, “The timeless way of building,” New York: Oxford University Press, 1979.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, ”Design Patterns: Elements of Reusable Object-Oriented Software,” Reading: Addison-Wesley, 1995.

[10] B. Bergin, “Fourteen Pedagogical Patterns,”. In M. Devos and A. Rüping (Eds.), EuroPLoP 2000, Proceedings of the 4th European Conference on Pattern Languages of Programs, pp.1-40, Konstanz: Universitätsverlag Konstanz, 2001.

[11] P. Avgeriou, S. Retalis, and A. Papasalouros, “Patterns for Designing Learning Management Systems,” In D. Schütz and K. Marquardt (Eds.), EuroPLoP 2003, Proceedings of the 9th European Conference on Pattern Languages of Programs, pp.115-132, Konstanz: Universitätsverlag Konstanz, 2004.

[12] C. Köppe, “Towards a Pattern Language for Lecture Design: An inventory and categorization of existing lecture-relevant patterns,” EuroPLoP 2013, Proceedings of the 18th European Conference on Pattern Languages of Programs, Irsee, Germany: ACM, 2013.

[13] C. Kohls and J. Wedekind, “Perspectives on patterns,” C. Kohls and J. Wedekind (Eds.), Investigations of E-Learning Patterns: Context Factors, Problems and Solutions, pp.2-18, Hershey: Information Science Pub, 2011.

[14] P. Goodyear, “Educational design and networked learning: Patterns, pattern languages and design practice,” Australasian Journal of Educational Technology, vol. 21, no. 1, pp. 82-101, 2005.

[15] Y. Dimitriadis, P. Goodyear, and S. Retalis (Eds.), “Design Patterns for Augmenting E-Learning Experiences,” Computers in Human Behavior, Special Issue on Design Patterns for Augmenting E-Learning Experiences, vol. 25, no. 5, 2009.

[16] Y. Harashima, T. Kubota, and T. Iba, “Creative education patterns: designing for learning by creating,” EuroPLoP 2014, 19th European Conference on Pattern Languages of Programs, New York: ACM, 2014.

[17] C. Kohls, and S. Panke, “Is that true? Thoughts on the epistemology of patterns,” PLoP 2009, Proceedings of the 16th Conference on Pattern Languages of Programs, New York: ACM, 2009.

[18] N.A. Streitz, J. Geißler, T. Holmer, S.I. Konomi, C. Müller-Tomfelde, W. Reischl, and R. Steinmetz, ”i-LAND: an interactive landscape for creativity and innovation,” Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 120-127, New York: ACM, 1999.

[19] D. Benyon, and O. Mival, “Blended spaces for collaborative creativity,” Proceedings of Workshop on Designing Collaborative Interactive Spaces, AVI2012, 2012.

[20] B. Johanson, A. Fox, and T. Winograd, “The interactive workspaces project: Experiences with ubiquitous computing rooms,”, IEEE pervasive computing, vol. 1, no. 2, pp. 67-74, 2002

 ISBN: 978-1-61208-612-5

Automatic Construction of Image Data Set Using Query Expansion Based on Tag Information

Yuki Mizoguchi Takahiro Yoshimura Yuji Iwahori

Department of Computer Science
Chubu University
Kasugai, 487-8501 Japan
Mail:mizoguchi@cvl.cs.chubu.ac.jp
Mail:yoshimura@cvl.cs.chubu.ac.jp
Mail:iwahori@cs.chubu.ac.jp

Ryosuke Yamanishi

College of Information Science and Engineering
Ritsumeikan University
Kusatsu, 525-8500 Japan
Mail:ryama@media.ritsumei.ac.jp

***Abstract*—Although some automatic construction methods of image data set have been proposed based on the web data, these methods used tag search for the image collection. The problem is that only images which have a target tag can be collected. This paper focuses on the tag given to the image of the data set constructed by the previous method. Tags which have been added to the image of the data set with high frequency and have high relevance to the target label are selected as new search queries. Furthermore, images collected based on the new search queries are classified by classifiers learned with the data set of the target labels constructed by the previous method. The goal is to construct and to expand image data set including images without the target label.**

Keywords–Image Data Set; Query Expansion.

I. Introduction

In the research of general object recognition, the computer recognizes images without constraints, and researches have focused on feature extraction and machine learning methods. A huge image data set with various unbiased objects is required for the learning. A manual collection of huge image data set causes biased collection and a lot of human costs.

Recent researches for image data set try to automatically or semi-automatically generate image data set from Web image sharing services, such as Flickr, i.e., Web image mining [1]. Web image mining enables makes it possible to obtain a large amount of images including daily scenes with low cost [2]. One of the problems in Web image mining is that the image data set generated with Web image mining naturally includes noise images. The method for noise removal using image features has been reported [3].

However, in collecting images from the web by most previous methods, tag search is used with the target label as a query, and the collection range is limited to images in which target words are included in meta data given to images, such as tags, titles, and descriptions.

Therefore, this paper proposes extending the data set by selecting appropriate queries and collecting images that can not be collected by the previous method.

In Section 2, proposed method is given. In Section 3, peformance is evaluated in the experiments. In Section 4, conclusion and future work are discussed.

II. Proposed method

The proposed method consists of the five steps: 1) Collecting images from web, 2) Noise images removal with visual features, 3) Query expansion based tag of image, 4) Recollecting images from web with new query, and 5) Noise images removal with visual features.

The detailed procedure of the proposed method is as follows.

Step1.	Collect images by tag search using target label as query.
Step2.	Removal of noise image by visual information and set the remaining image as ”correct image”.
Step3-1.	Calculate the appearance frequency from the tag given in ”correct image”.
Step3-2.	Calculate the cosine similarity between the tag given to ”correct image” on fastText and the target label.
Step3-3.	Select tags that matches ”high occurrence frequency” and ”high similarity with target label” among the tags as new queries.
Step4.	Collect images by tag search using query selected by Step3.
Step5.	Removal of noise image by visual information as learning by ”correct image”.

A. Construction of Image Data Set

In this subsection, the Steps 1) and 2) are explained. The original idea of Steps 1) and 2) is from our previous paper [4]. To generate a data set as the basis of query expansion, images are collected according to target labels, and noise removal by visual information.

First of all, images are collected from the web as a query using a target label. Next, among the obtained images, images having minority image features are eliminated as noise images according to automatically determined threshold values.

Image features are feature vectors representing SIFT features as Fisher Vectors. In this paper, SIFT features are extracted by grid sampling, extraction interval is $8px$, extraction window size is $16px$. The number of visual words K when calculating Fisher Vector from the image was set to 512. Get Euclidean distance $FD^l(i)$ between each image feature amount $G(i)$ and the centroid vector M^l of all images. We classify images collected by query l based on $FD^l(i)$ as follows.

$$VI^l = \begin{cases} correct\_image & (FD^l(i) \leq VT^l) \\ noise\_image & (FD^l(i) > VT^l) \end{cases} \quad (1)$$

 ISBN: 978-1-61208-612-5

Here, VI^l is a threshold value, which is the average value of $FD^l(i)$ for each NI images.

$$VI^l = \frac{1}{NI}\sum_{i=1}^{NI} FD^l(i) \tag{2}$$

B. Query Expansion

In this subsection, the Step 3) will be shown. Query expansion based on data set constructed by previous method. We focus on the tag of the image inside the data set as the basis of the query extension.

It is conceivable that the query to be selected is included in the tag inside the data set constructed by the previous method. Therefore, the appearance frequency is used as the first criterion of query expansion.

Here, the appearance frequency $hist$ of a certain label is expressed as in (3) when $count$ is the number of images given the label among the number of images num .

$$hist = \frac{count}{num} \tag{3}$$

In addition, tags have high frequency of appearance, but there are tags with low relevance to target labels. For example, words, such as 'Canon' representing a device that photographed the picture and 'London' representing the point where the picture was taken are often tags that the user gives to the image in Flickr. In order to filter such tags with a general relationship, cos similarity on fastText is used as a second criterion.

Here, the cos similarity sim in a fastText of a label and a target label is expressed by the expression (4) when M is the vector of the target label on fastText and V is the vector of a label on fastText .

$$sim = \frac{M \cdot V}{|M||V|} \tag{4}$$

Among tag given to the image in data set of target label, a tag having "high occurrence frequency" and "high similarity" is set as a new search query. In order to express "high occurrence frequency" and "high similarity degree", the criterion $Score$ of tag selection is assumed to be multiplied by $hist$ and sim, and expressed as in (5) . Here, exclude tags that sim can not calculate because there is no word on fastText.

$$Score = hist * sim \tag{5}$$

Finally, rank tags based on $Score$. However, we exclude the target labels themselves and the conjugative form, the plural forms of target labels.

C. Expansion of Image Data Set

In this subsection, the Steps 4) and 5) will be shown. First, images are collected from the web by the query selected in Step 3. Second, convert the collected images to image features according to step 2. Finally, the image outside the threshold is removed as a noise image by using the threshold of the image feature amount used in step 2. The remaining image is added to the data set as a correct image of the target label, and the data set is expanded.

III. Experiments

A. Query Expansion Experiments

In query expansion experiments, we show that the proposed method can expand the query. We collect 2,500 images from Flickr for each label and experiment on 5 target labels of dog, cat, kick, throw, jump. Queries selected by query expansion on the proposed method are shown in Tables I-V. Although the target label in the table is not $hist = 1$, this is because Flickr's search algorithm gathers capital letters and lowercase letters without distinction.

As shown in Tables I-V, tags that are not suitable as queries like 'Canon' or 'London' have been removed.

TABLE I. TAGS SELECTED BY QUERY EXPANSION TO DOG

tag	hist	sim	Score
dog	0.411302	1.000000	0.411302
Dog	0.435850	0.497419	0.216800
Competicion	0.327466	0.230190	0.075379
Elechas	0.327466	0.201274	0.065910
Agility	0.327466	0.188627	0.061769
can	0.328393	0.175213	0.057539
Agility Cantabria	0.327466	0.174608	0.057178
deporte	0.327466	0.147234	0.048214
dogs	0.072256	0.659097	0.047624
puppy	0.085225	0.402321	0.034288

TABLE II. TAGS SELECTED BY QUERY EXPANSION TO CAT

tag	hist	sim	Score
cat	0.603858	1.000000	0.603858
cats	0.201884	0.707840	0.142902
kitten	0.291611	0.415140	0.121059
Cat	0.165545	0.469826	0.077777
catsagram	0.157470	0.475644	0.074900
catsoftwitter	0.157470	0.474278	0.074684
catsofinstagram	0.159264	0.413597	0.065871
kittens	0.157918	0.396114	0.062554
kitty	0.202333	0.295444	0.059778
cute	0.221175	0.246822	0.054591

TABLE III. TAGS SELECTED BY QUERY EXPANSION TO KICK

tag	hist	sim	Score
kick	0.491503	1.000000	0.491503
Kick	0.495528	0.539903	0.267537
Referee	0.294275	0.421585	0.124062
Tackle	0.294275	0.382568	0.112581
Soccer	0.297853	0.368589	0.109785
Football	0.294723	0.315795	0.093072
Pass	0.294275	0.302313	0.088963
Team	0.294275	0.301781	0.088807
Header	0.288014	0.279592	0.080526
Shot	0.294275	0.254012	0.074750

TABLE IV. TAGS SELECTED BY QUERY EXPANSION TO THROW

tag	hist	sim	Score
throw	0.804951	1.000000	0.804951
Throw	0.161758	0.801584	0.129663
ball	0.163892	0.512881	0.084057
throwout	0.096458	0.840592	0.081081
sports	0.288092	0.259825	0.074854
pitcher	0.170294	0.418618	0.071288
sport	0.183099	0.320428	0.058670
catcher	0.105847	0.489021	0.051761
catch	0.091336	0.548228	0.050073
baseball	0.152796	0.327365	0.050020

 ISBN: 978-1-61208-612-5

TABLE V. TAGS SELECTED BY QUERY EXPANSION TO JUMP

tag	hist	sim	Score
jump	0.853669	1.000000	0.853669
volleyball	0.220582	0.363222	0.080120
Jump	0.110725	0.633634	0.070159
team	0.260096	0.251929	0.065526
jumping	0.095962	0.630887	0.060541
spike	0.196266	0.306962	0.060246
sport	0.123752	0.334476	0.041392
block	0.148502	0.272300	0.040437
School	0.184542	0.205032	0.037837
High	0.225792	0.156843	0.035414

B. Evaluation

It is investigated if the image of the data set expanded by the proposed method matches the target label. Queries are superior ranked Score, except for themselves, the conjugative forms and plural forms of target labels from the Tables I-V. These are shown in Table VI. Based on these queries, 5000 images are collected for each, remove noise and build a data set. 200 images were randomly extracted among the constructed data set and the evaluators evaluated for the question whether the image of the data set constructed matches the target label or not.

In this experiment, the image selected that 70% or more of the evaluators match the target label in the questionnaire is taken as the correct image. Among the images of the data set constructed by the proposed method, the ratio of the correct image by subjective evaluation is taken as accuracy. The results of the evaluation experiment are shown in Table VI. Also, an example of the image of the data set constructed by the proposed method is shown in Figures 1-5.

TABLE VI. TARGET LABEL, SELECTED TAG AND ACCURACY

target label	selected tag	accuracy[%]	*1	*2
dog	Agility	93.2	187	174
cat	kitten	91.8	182	167
kick	soccer	28.8	46	13
throw	picher	69.9	97	66
jump	volleyball	14.7	34	5

*1 : Number of correct images by subjective evaluation
*2 : Among the images judged as correct by the proposed method, the number of correct images by subjective evaluation

In dog-Agility, cat-kitten, these accuracy are as high as 90% or more. Most images collected by the query selected by the proposed method are that match the target labels. Thus, proposed method selected useful queries.

Although some other kind of images were obtained from ”pitcher”, noise removel could remove the images except baseball pitcher and its accuracy was around 70%.

The accuracy for some images with kick-soccer and jump-volleyball was low. This is because the number of correct answer images were also small. On Frickr, when collecting images using selected tags as queries, the images clearly matching target labels were significantly smaller than other labels.

Figure 1. Ex.:dog-Agility

Figure 2. Ex.:cat-kitten

Figure 3. Ex.:kick-soccer

Figure 4. Ex.:throw-pichar

Figure 5. Ex.:jump-volleyball

IV. CONCLUSION AND FUTURE WORK

This paper proposes an image data set expansion method that selecting appropriate queries and collecting images that can not be collected in previous methods.

In the experiment, it is possible to select a useful query for data set extension by query expansion from the image tag of the data set constructed by the previous method.

As a future task, in order to select more useful queries, we consider not only combinations of frequencies and relevance, but also combinations of ”occurrence frequency of tags of images removed as noise” or TF-IDF.

ACKNOWLEDGEMENT

Iwahori’s research is supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid Scientific Research(C)(#17K00252) and Chubu University Grant.

REFERENCES

[1] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “Nus-wide: a real-world web image database from national university of singapore,” in Proceedings of the ACM international conference on image and video retrieval. ACM, 2009, p. 48.

[2] X. Li, C. G. Snoek, and M. Worring, “Unsupervised multi-feature tag relevance learning for social image retrieval,” in Proceedings of the ACM International Conference on Image and Video Retrieval. ACM, 2010, pp. 10–17.

[3] R. Fergus, P. Perona, and A. Zisserman, “A visual category filter for google images,” Computer Vision-ECCV 2004, 2004, pp. 242–256.

[4] S. Otani, R. Yamanishi, and Y. Iwahori, “Generation of web image database based on hybrid noise removal method of visual and semantic feature,” Journal of Japanese Society for Artificial Intelligence, vol. 32, no. 1, 2017, pp. WII–N_1–10.

Automatic Construction of Large Scale Image Data Set from Web Using Ontology and Deep Learning Model

Takahiro Yoshimura Yuki Mizoguchi Yuji Iwahori

Department of Computer Science
Chubu University
Kasugai 487-8501 Japan
Email: {yoshimura|mizoguchi}@cvl.cs.chubu.ac.jp
Email: iwahori@cs.chubu.ac.jp

Ryosuke Yamanishi

College of Information Science and Engineering
Ritsumeikan University
Kusatsu 525-8500, Japan
Email: ryama@media.ritsumei.ac.jp

***Abstract*—General object recognition requires a large scale image data set and recognition accuracy depends on the image data set used for the learning. This paper proposes a method to collect only the target images using Ontology and features obtained from obtained from Convolutional Neural Network (CNN). The proposed approach constructs large scale data set automatically by expanding the range of collected images. It is shown that the proposed approach is effective by collecting the image data set and a mean accuracy 88.6% was obtained from the subjective evaluation in experiments.**

Keywords–Image data set; Web image mining; Image processing; Web intelligence.

I. INTRODUCTION

General object recognition is the recognition of unconstrained image existing in the real world using computer system and this is one of the representative tasks in the computer vision. Since Convolutional Neural Network (CNN) [1] was proposed in 2012, the recognition ratio was improved dramatically. However, recognition requires a large scale image data set and the recognition accuracy depends on the image data set. Construction of large scale image data set by human requires a lot of time and human costs. Automatic or semi-automati construction of image data set using Web image mining searching images on the Web has been reported recently [2][3] to avoid the construction by human manually. Web image mining makes it possible to obtain large scale images taken under usual conditions by various humans with low cost since searching operation is available via posting service of a large scale images such as Flicker, Bing Image Search or Google Image Search and so on.

Only the Meta information added to the image, such as title, explanation sentence or tag is stil difficult to collect the target image data set. Image data collected automatically from the Web includes non-target images (noise images) and it is stil difficult to apply these approach directly to the general object recognition. This paper uses the low level concept of ontology and expands to increase the number of image data set and to exclude the noise images simultaneously, then how to perform the automatic construction of a large scale image data set is proposed.

II. PROPOSED APPROACH

The proposed approach constructs a large scale image data set as follows.

- Step 1 Collecting image data from Web
- Step 2 Recollecting image data using the low level concept of ontology
- Step 3 Integrating image data set by excluding the collected duplication images
- Step 4 Removeing the colleted noise images

The detail is shown below.

A. *Collecting Images from Web*

In this section, the Step 1) will be shown. Flickr is used to collect images as a collection method. Flickr is available with keyword for searching images. Related tag is added to the uploaded image by user. Tag consists of words related to the contents of image. The proposed approach searches the related images for the added tag by using the label of constructed data set as the searching query.

B. *Recorrecting Images by Low Level Concept of Ontology*

In this section, the Step 2) will be shown.

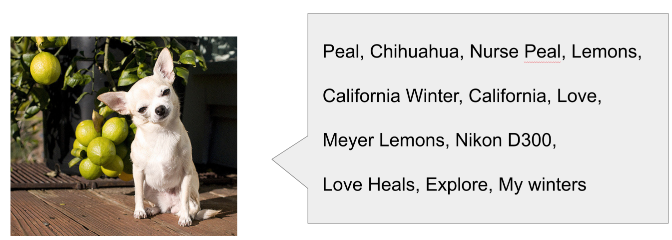

Figure 1. Tags added to Image

Image search by Flickr is applied according to the tags addef by user. When "dog" is given as the label of image data set to construct, there are sometimes no hits even if "dog" is the target label as shown in Figure 1. However, "Chihuahua" is sometimes the tag added to the image, which is the low level concept of "dog". Recorrecting images using this low level concept of the label is used for this purpose to expand the range of correcting images. This paper uses DBpedia[4] as ontology and perform this strategy.

C. *Excluding Duplication Images and Integrating Image set*

In this section, the Step 3) will be shown. Integrating the corrected images is done by Step 1 and Step2. Corrected images sometimes include the case that both of the target label and the target low level concept label are added to the image. In this case, images are corrected with duplication and processing to exclude the duplication images is necessary.

D. Removing Noise Images

In this section, the Step 4) will be shown. There is the case that noise images are included in the image corrected from Web. Noise images are inappropriate for the search query and removing noise images are necessary for the general object recognition.

Feature extration is applied to the image i which was corrected from the Web by the search query q. Here, let the number of the corrected images by the search query q be NI. CNN is used as the image feature extractor, and VGG16 [5] is used as the CNN model. 4096 dimensional feature vector obtained from fc2 layer of VGG16 was used as image features $G(i)$. Euclidian distance $FD^q(i)$ was obtained between each image feature vector $G(i)$ and those centroid vector M^q of all images.

$$\boldsymbol{VI}^q = \begin{cases} \text{Appropriate Image} & FD^q(i) <= VT^q \\ \text{Noise Image} & FD^q(i) > VT^q \end{cases} \tag{1}$$

$$VT^q = \frac{1}{NI}\sum_{i=1}^{NI} FD^q(i) \tag{2}$$

where threshold value VT^q is given by the mean value of $FD^q(i)$ of each image feature vector and entroid vector.

III. Experiment

Subjective evaluation was applied for whether the proposed approach adds the appropriate label for the corrected image data set or not. Image data set was constructed by correcting from Flickr and labels used for the evaluation were ”cat”, ”crab”, ”elephant”, ”fox”, ”giraffe”, and ”lion” as the general words. 100 images are randomly extracted from the constructed image data set and 6 evaluators evaluated whether images in the constructed data set are matched for the object labels or not.

Constructed image data set is shown in Figure 2 and evaluation is shown in Table I.

TABLE I. RESULT BY SUBJECTIVE EVALUATION [%]

Search Keyword	Precision
cat	93.3
crab	94.0
elephant	88.3
fox	79.6
giraffe	79.5
lion	96.7
AVG	88.6

Table I suggests that ”cat”, ”crab”, and ”lion” gave more than 90 % accuracy. ”crab” included some cuisine images but noise images were removed correctly. ”fox” corrected some inappropriate images including building or humans. ”giraffe” was judged as zebra or kangaroo and this decreased accuracy.

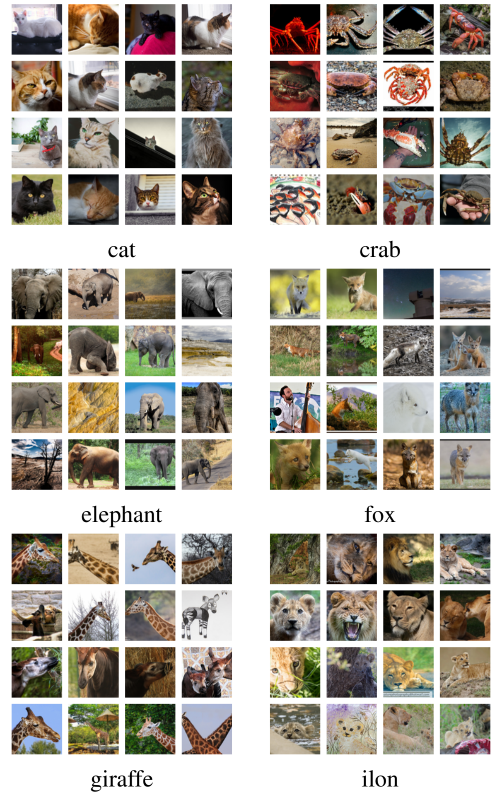

Figure 2. Example of Constructed Image Data Set

IV. Conclusion and Future Work

This paper proposed an automatic construction of Web image dataset by removing noise images using ontology and CNN features. Low level concept of ontorogy made it possible to recorrecting images and expand the range of correcting images. Removing noise images was also applied using the image features obtained by CNN for the corrected images. It was confirmed that image data set constructed by the proposed approach is available to the general object recognition. Evaluation using this generated dataset for the general object recognition is our future task.

Acknowledgment

Iwahori’s research is supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid Scientific Research(C)(#17K00252) and Chubu University Grant.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”Imagenet classification with deep convolutional neural networks”, Advances in neural information processing systems, pp. 1097-1105, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR), 2009.

[3] S.Otani, R.Yamanishi, Y.Iwahori, ”Generation of Web Image Database Based on Hybrid Noise Removal Method of Visual and Semantic Features”, Journal of JSAI (Japanese Society for Artificial Intelligence), vol. 32, no. 1, WII-N 1-10, 2017.

[4] J. Lehmann, et al. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal, 2014.

[5] K.Simonyan,A.Zisserman, ”Very deep convolutional networks for large-scale image recognition”, arXiv 2014.

 ISBN: 978-1-61208-612-5

Colon Blood Vessel Detection Based on U-net

Hiroyasu Usami Yuji Iwahori

Department of Computer Science
Chubu University
Kasugai, 487-8501 Japan
Email: usami@cvl.cs.chubu.ac.jp
Email: iwahori@cs.chubu.ac.jp

Naotaka Ogasawara Kunio Kasugai

Department of Gastroenterology
Aichi Medical University
Nagakute, 480-1195 Japan
Email: nogasa@aichi-med-u.ac.jp
Email: kuku3487@aichi-med-u.ac.jp

***Abstract*— Detection and analysis of colon blood vessels play a significant role in medical diagnosis. This paper proposed the colon blood vessel detection method based on the U-net architecture with a few training images. The proposes method performs better colon blood vessel detection with a few training images from experiments result.**

***Keywords*–Colon Blood Vessel Detection; CNN; U-net; Biomedical Image Processing.**

I. INTRODUCTION

Detection and analysis of colon blood vessels play a significant role in medical diagnosis in a large number of areas like retinopathy, endoscopy, etc. Colon blood vessel lesions are frequently derived from malignant polyps. While some algorithms [1]-[4] are proposed for retina blood vessel extraction, they can not be used in an endoscope environment as well. The algorithms based on convolutional neural network (CNN) [3] [4] perform high precision in retina blood vessel. However, common CNN needs massive datasets. It is difficult to prepare massive datasets with mask images in biomedical image processing. The typical use of convolutional networks is for the classification tasks, where the output of an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. This paper proposes a method based on the U-net architecture [5] for colon blood vessel detection with a few training images. Section Ⅱ explains the proposed method, Section Ⅲ mentions experiments and Section Ⅳ concludes the paper.

II. POPOSED METHOD

The U-net is a network and training strategy that relies on the active use of data augmentation to use the available annotated samples more efficiently, and it is the so-called “ fully convolutional network” [6]. The proposed method based on this architecture works with a few training images to try better colon blood vessel detection.

The network architecture of the proposed method is shown in Fig. 1. The network architecture was constructed with less down sampling layers because the resolution of the targeted endoscope images is not high.

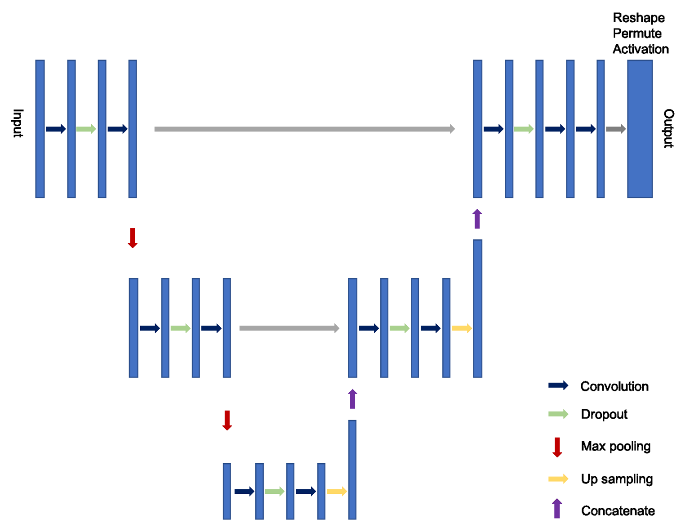

Figure 1. Network Architecture

The network architecture details are as follows. The loss function is the cross-entropy and the neural network employees the stochastic gradient descent for optimization. The activation function after each convolutional layer is the Rectifier Linear Unit (ReLU) and between two consecutive convolutional layers use a dropout of 0.2 empirically set.

The following pre-processing is performed for the training datasets before training.

- Gray-scale conversion
- Normalization
- Contrast-limited adaptive histogram equalization (CLAHE)
- Gamma adjustment

The neural network is trained based on sub-images (patches) of the pre-processed full images. Each patch with dimensions 48x48 is obtained by randomly selecting its center inside the full image. Sample patches are shown in Fig. 2. Also, selected patches partially or entirely outside the Field Of View (FOV), thus the neural network learns how to discriminate the FOV border from the blood vessels.

 ISBN: 978-1-61208-612-5

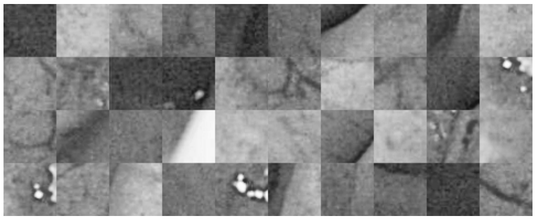

Figure 2. Sample of Patches

III. EXPERIMENTS

An evaluation experiment was conducted to confirm the effectiveness of the proposed method.

A. *Training*

A set of 142,500 patches was obtained by randomly extracting 71,250 patches in each of 15 training images. Although the patches overlap, i.e., different patches may contain a same part of original images, no further data augmentation was performed. The first 90% of the dataset was used for training (128,250 patches), while the remained 10% was used for validation (14,250 patches). Training was performed for 150 epochs, with a mini-batch size of 32 patches. It took about 4 hours using GeForce GTX 1080 GPU × 4 for the training.

B. *Evaluation of the trained model*

The vessel probability of each pixel was obtained by averaging multiple predictions to improve the performance. With a stride of 5 pixels in both height and width, multiple consecutive overlapping patches were extracted in each test image. Then, the vessel probability was obtained for each pixel by averaging probabilities over all the predicted patches covering the pixel.

Fig. 3 shows the results of colon blood vessel detection and overall accuracy of the prediction comparison with correct mask image was 0.8432.

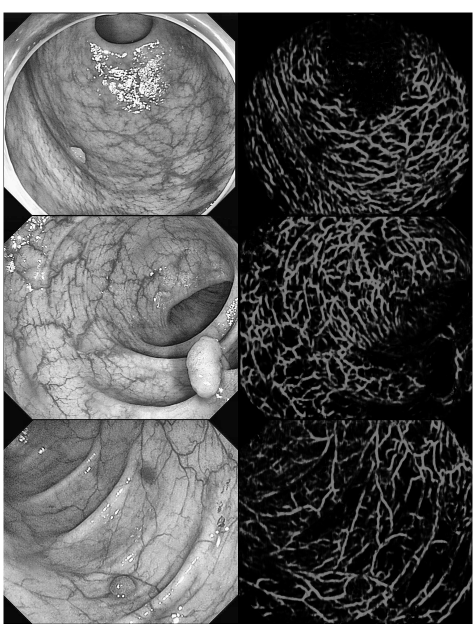

Figure 3. Example of Prediction Results

Fig. 4 shows the Area Under the ROC curve (AUC ROC) of the evaluation, and AUC was 0.9030.

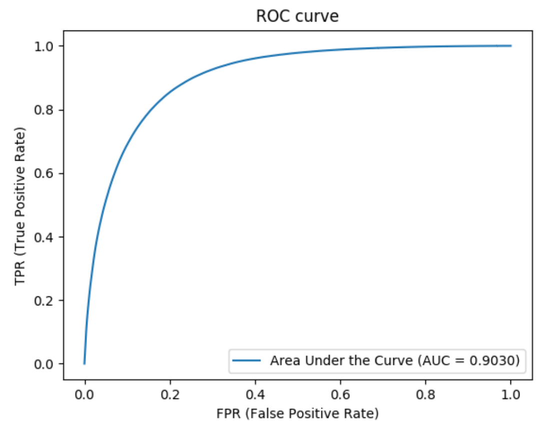

Figure 4. AUC ROC

IV. CONCLUSION

This paper proposed the colon blood vessel detection method based on the U-net architecture with a few training images. It was confirmed that the proposed method performs better colon blood vessel detection with a few training images from experiment results. Applying the proposed method to another kind of endoscope image such as Narrow Band Imaging (NBI) and detecting malignant polyp based on results are future works.

ACKNOWLEDGEMENT

Iwahori's research is supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid Scientific Research(C)(#17K00252) and Chubu University Grant.

REFERENCES

[1] Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998, October). Multiscale vessel enhancement filtering. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 130-137). Springer, Berlin, Heidelberg.

[2] Azzopardi, G., Strisciuglio, N., Vento, M., & Petkov, N. (2015). Trainable COSFIRE filters for vessel delineation with application to retinal images. Medical image analysis, 19(1), 46-57.

[3] Melinšćak, M., Prentašić, P., & Lončarić, S. (2015, January). Retinal vessel segmentation using deep neural networks. In VISAPP 2015 (10th International Conference on Computer Vision Theory and Applications).

[4] Liskowski, P., & Krawiec, K. (2016). Segmenting retinal blood vessels with deep neural networks. IEEE transactions on medical imaging, 35(11), 2369-2380.

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[6] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Control Transition Interfaces in Level 3 Automated Vehicles Four Preliminary Design Patterns

Alexander G. Mirnig and Manfred Tscheligi

Center for Human-Computer Interaction
University of Salzburg
Salzburg, Austria
Email: alexander.mirnig@sbg.ac.at, manfred.tscheligi@sbg.ac.at

***Abstract*—Automated driving technology is rapidly progressing, which causes increasingly complex in-vehicle systems and similarly complex in-vehicle interaction contexts in need of appropriate solutions. One such context is initiating or confirming control transitions between manual and automated modes in semiautomated level 3 vehicles, which is a particularly challenging context due to the combination of potentially inattentive drivers who need to take back control in emergency situations. In this paper, we present some of the results from a review analysis of academic and industry publications as four preliminary patterns for transition designs in level 3 vehicles.**

***Keywords*–Automated vehicles; SAE level 3; Design patterns; Control transitions; Interface design.**

I. INTRODUCTION

Recent years have seen a rapid development and constant increase of automated systems in vehicles. High automation in vehicles is not merely a by-product of advances in automation design but is pursued as a primary goal with the intent to reduce accidents and casualties by reducing the human element and, thus, human error. Currently, the vehicle industry is right in the middle on the road from fully manual to fully automated operation. This so-called *transition phase* bears a number of particular challenges to overcome, caused by a necessity for both human and automated system to interact and conduct parts of the dynamic driving task.

One particularly challenging context is that of SAE (Society of Automotive Engineers) level 3 [1]. At this level of automation, also called "conditional automation" the vehicle can execute all aspects of a dynamic driving tasks while the driver is still required to intervene and take back control, either if he/she chooses to do so or if the system is unable to handle a certain situation and requests a driver intervention. While a level 3 vehicle is driving in an automated mode, the driver is only required to be *receptive* to system output relevant to a request to take back control. The monitoring of the driving environment, which includes assessment of the driving situation, is also performed by the vehicle.

This combination of factors can lead to extended response times in case of emergency transition requests. In such an emergency, the vehicle might decide to transition control back to the driver, either due to difficult traffic conditions or system failure. The driver, however, can not be expected to be ready to take control in the same way a driver of a manual vehicle would, as environment monitoring up to that point was handled by the vehicle. Thus, the driver is likely to require additional time to react to the transition request and then assess the driving situation – time, which he/she might not have in an emergency situation. Some have expressed a desire to skip this level altogether and shorten the transition phase as much as possible [2]. It is doubtful, however, that this skip will be as effective or short as intended and not extended by, e.g., unforeseen technical difficulties or setbacks in development, which are known to happen frequently in virtually any area of research and development. It is sensible to try and provide effective transition interface solutions for this transition phase, long or short as it may be, as these solutions will increase driver safety during that phase.

In order to provide a guide for level 3 transition design, we conducted a review of available literature from academia and industry, with the aim of identifying holes and potentials in automotive design and research. The results of this review are reported in Mirnig et al. 2017 [3]. While analyzing the results of this review, we also identified a number of consistencies among interface designs, which could serve as a further useful basis for designing automated vehicle interfaces. We decided to compile these consistencies in the form of preliminary design patterns. In this paper, we present the first four of these patterns. After this introduction, in Section II, we briefly outline related work regarding design patterns and vehicle automation levels. The patterns themselves are presented in Section III. We discuss these patterns in Section IV and provide a conclusion and future outlook in Section V.

II. RELATED WORK

In the following, we provide an overview of automation levels in Subsection II-A and design patterns in Subsection II-B

A. Levels of Vehicle Automation

The extent to which a vehicle is capable of automated operation is usually expressed via *levels of automation*. The three most common definitions for automation levels come from SAE International, the National Highway Traffic Safety Administration (NHTSA), and the German Federal Highway Research Institute (Bundesanstalt fur Straenwesen BASt). While having differences in level of detail and focus on vehicle functions, all three standards describe automation as incremental steps from a basic, nonautomated, to a fully automated level. On an international level, the SAE standard is the most widely used one, to the point where the NHTSA abandoned their automation level definitions and decided to adopt the SAE's at the end of 2016 [4]. For the sake of consistency and clarity, we will also use these levels in this paper.

According to the SAE J3016 standard, vehicle automation ranges from across six levels: level 0 (no driving automation), level 1 (driver assistance), level 2 (partial driving automation), level 3 (conditional driving automation), level 4 (high driving automation, and level 5 (full driving automation). Level 3 is defined as "The sustained and ODD-specific (ODD=Operational Design Domain) performance by an ADS (Automated Driving System) of the entire DDT (Dynamic Driving Task) with the expectation that the DDT fallback-ready user is receptive to ADS-issued requests to intervene, as well as to DDT performance-relevant system failures in other vehicle systems, and will respond appropriately," and represents a critical breakpoint between levels 2 and 3. On level 2, monitoring of the environment is still on the part of the driver, which means that a transition of control from automated driving mode to an unprepared driver does not occur under normal circumstances. On level 4, fallback performance, i.e., handling of the driving task in case of driver's inability to take control or system failure, is to be handled by the vehicle. This means that from level 4 onwards, transition requests from the vehicle are never mandatory to be responded to. Level 3 combines environment monitoring by the vehicle with mandatory fallback performance by the human driver, which is the cause for the likely scenario of control transition to an unprepared driver, making level 3 a particularly challenging design space.

B. Design Patterns

Christopher Alexander is considered by many as the "founder" of contemporary pattern approaches. He initially developed the concept of patterns for architecture, his home domain. According to him, constructing a building consists of a string of solutions to individual and context-dependent problems. It is these problems and their solutions that make up the conceptual building blocks. In his classic work on patterns [5] [6], he described both the basic pattern approach and developed a collection of patterns for the construction of buildings.

His ideas were later adopted by other disciplines as a problem solution documentation method. One of the most well-known of these adaptations is the work of Gamma et al. [7], who provided a pattern structure and pattern collection for software engineering. Human-Computer Interaction, of which automotive interface design is a subdomain [8], has also adopted patterns as a method to document working interface design solutions as *design patterns*. Patterns, which describe working solutions to reoccurring problems, have a counterpart in the so-called *antipatterns*. Antipatterns describe solutions that might look like they should work at first but do not in practice [9]. There is also a third type of patterns, called *dark patterns*, which describe solutions intended to deceive or trick users. Since the focus of these patterns is on the intention behind a solution and less on the implementation and its reproducibility, they can be seen as outside of the "classic" pattern spectrum inhabited by patterns and antipatterns [10].

Pattern writing is an iterative process, which starts with the initial *pattern mining* [11] [12]. Pattern mining is the act of looking for repeated, working solutions within a pool of available data and deciding, whether or the solution is worth making into a pattern. Once a solution has been mined, the actual writing begins. This writing usually follows a standard pattern template or structure, which ensures that the problem and its surrounding context are captured in enough detail to make the solution reproducible, and allow easier sorting and referencing of a pattern in a later pattern collection. According to the minimal pattern structure described in [10], a successful pattern should at least always contain a means of reference, a description of the problem, a description of the context the problem occurs in, the solution in detail, and at least one (although ideally more than one) example. Initial (and intermediate after the first cycle) pattern versions are then iterated by other pattern authors or domain experts, either on a one-to-one basis or in the form of writer's workshops. Once the patterns have reached a high enough level of quality and equally sufficient number, they are published as a pattern collection, either in print or in an online repository.

III. Control Transition Interface Patterns

This work is based on a literature review conducted for the purpose of identifying holes and/or weaknesses in current control transition interaction designs. The review was based on a total number of 469 scientific publications (via IEEE Xplore, ScienceDirect, and the ACM Digital Library) and 200 industry patents (via Depatisnet and Google Patents). After initial analysis regarding the presence of actual interface implementation descriptions, the papers were reduced down to 35 academic papers and 22 industry patents, which were analysed in detail. The full analysis procedure and results can be found in Mirnig et al. 2017 [3]

In this paper, we focus on some of the design regularities that were found as an additional result of the analysis. We use the minimal pattern structure described in [10] for the pattern format. Each pattern consists, thus, of a unique **name**, a **problem** statement, a **context** description, a **solution** description, and several **examples**. The set of patterns presented are preliminary patterns, which have not undergone a full iteration cycle. Thus, future extensions will require iteration and further validation of the current patterns as well. The pattern content is mostly based on the literature reviewed in Mirnig et al. 2017 [3]. Explicit references are made when other sources are cited or concrete examples are provided in the corresponding Subsection of each pattern.

A. Pattern 1: Interaction Method to respond to or initiate transition requests

Problem: Control transitions require adequate means for the human driver to either initiate a request for the vehicle to drive in autonomous mode or respond to a request by the vehicle to take control. Since control transitions represent a fundamental change in driving task performance and delegation, such controls should not be easily confused with other controls and be difficult to activate by accident.

Context: In accordance with the taxonomy proposed by McCall et al. [13], control transitions can occur in five different configurations: scheduled, non-scheduled system initiated, non-scheduled user initiated, non-scheduled system initiated emergency, and non-scheduled user initiated emergency. Scheduled transitions allow longer preparation times for both driver and vehicle (depending on the transition direction).

Nonscheduled transitions, especially in emergency situations, require quick, efficient and error-free input and should, thus, be the focus when designing transition controls.

Solution: Since a transition means either assuming or relinquishing control, it can be assumed that the driver will be physically connected to the standard physical vehicle controls (steering wheel and pedals) at some point during the transition by necessity. Thus, a sensible position for both initiation and confirmation of transitions is near the physical vehicle controls. Since actuating controls on or near the steering wheel does not inhibit using the wheel's primary control means (turning), proximity to the wheel should be preferred over the pedals. Physical transition controls (switches, knobs, levers, etc.) should, therefore, be placed close to the steering wheel to enable faster execution of driving maneuvers after the transition while (in case of driver to vehicle transitions) simultaneously guiding the driver's attention towards the primary vehicle controls and the road ahead.

Examples: There are numerous implementations for physical transition controls close to the steering wheel. Cullinane et al. [14] describe a button to press on the left side of the steering wheel. Boehringer et al. [15] and Gazit [16] specify actuation of the steering wheel as a transition initiation from vehicle to driver. Coelingh [17] describes actuation of either steering wheel or pedals as transition initiation. Note that such direct approaches increase the likelihood of accidental transitions initiations if no additional confirmatory steps are present. The Tesla S (see Figure 1, which is used, e.g., in a study setup by Dikmen et al. [18], uses the cruise control lever (pull twice towards the driver) for control transitions initiated by the driver.

Figure 1. Tesla S cruise control lever [19]

B. *Pattern 2: Consistent Visual Metaphor to Signal Transition Requests*

Problem: Due to the novelty of automated vehicles, there are few to no common visual concepts that a driver would naturally associate with a control transition. Thus, it is difficult to use appropriate visual indicators that evoke familiarity in the driver and are unambiguous in their meaning.

Context: There is a multitude of indicators in the cockpit the driver needs to monitor and/or be receptive to. While Advanced Driver Assistance Systems (ADAS) of levels 1 and 2 usually have their own indicators to show whether they are active or inactive, level 3 automation constitutes a different degree of autonomy that allows the driver to perform a primary task other than driving. This means that existing ADAS indicators are not, by default, suitable for also displaying control transitions. A suitable visual metaphor must focus on the essential process of resuming or relinquishing the vehicle's controls.

Solution: The primary interaction means when operating a vehicle are the pedals and steering wheel. Of these two, the steering wheel shape is most easily recognizable and, therefore, presents a robust visual metaphor for assuming or relinquishing control. Since the wheel itself represents controllability by a human, a simple way to use this metaphor is displaying a steering wheel when control by the driver is requested and not displaying it when control is to be relinquished.

Examples: In a simulator setup by van den Beukel et al. [20], transition requests are communicated via a steering wheel icon with superimposed hands grasping the wheel (see Figure 2). When no hands and only the steering wheel are displayed, no transition request is taking place. While the display also contains color coded information, the transition requests are only communicated via displaying the steering wheel with hands on or off.

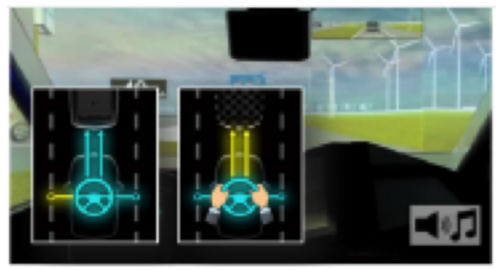

Figure 2. Wheel icon with hands on or off

In a setup by van der Meulen et al. [21], the steering wheel is displayed as a semitransparent icon in the middle of the screen together with a verbal message at the bottom of the screen, whenever a transition is requested by the simulator (see Figure 3). In a real vehicle, displaying an icon in the middle of the screen would require a windshield display or projection, which might not be feasible. In general, the icon is best positioned where the driver's primary visual attention lies while performing driving (manual mode) or nondriving (automated mode) tasks.

Figure 3. Semitransparent steering wheel icon in screen center

The Google patent by Cullinane et al. [14] describes a steering wheel icon with one hand on the wheel (in the same position, where the button to confirm the request is located). The icon is accompanied by a verbal message to push said button when ready to respond to the transition (see Figure 4).

Figure 4. Wheel icon with one hand and verbal message

The Tesla Model S similarly displays transition requests as a verbal message in a box at the bottom of the navigation display with a blue steering wheel icon to the left of the message (see Figure 5).

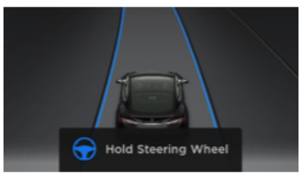

Figure 5. Coloured steering wheel with verbal message

C. Pattern 3: Priority of transitions

Problem: Control transitions can occur in two directions – from driver to vehicle or from vehicle to driver. Depending on which of the two directions is prioritized, different design implications arise for the in-vehicle controls. The problem is deciding which of these (or if neither) should be prioritized and which design implications they entail.

Context: In line with McCall et al. [13], the first distinction to be made is between scheduled and unscheduled transitions. Scheduled transitions put emphasis on planning of the transition, more so on the driver's than the vehicle's side as the initiator. Unscheduled transitions put emphasis on reaction to transition requests, suggesting a focus on the driver as the recipient of the transition request. Depending on whether the nonscheduled transition is an emergency or not, this emphasis might be different.

Solution: *vehicle to driver*: Scheduling a transition from vehicle to driver can depend on traffic or road conditions that the vehicle is not equipped to handle or the vehicle reaching the limits of its operational design domain in any other way. Since the scheduling of these transitions is on the vehicle's side, the driver only needs to be able to respond to such requests in time without necessarily requiring further information or input means. If the scheduled transition is known well ahead in advance, then an additional output with an estimated time indicator would be beneficial. At a basic level, however, an interface design for responding to unscheduled transition requests from driver to vehicle is as also suitable for scheduled ones. Thus, when an interface needs to accommodate scheduled transitions, vehicle-driver transitions should be prioritized during design with designs for unscheduled transitions being usable for scheduled transitions as well.

Driver to vehicle: An effective interface for scheduled transitions from driver to vehicle requires support for the driver to plan the transition ahead of time. This would ideally occur in a navigational interface, where the driver can not only input the planned route but also points (e.g., motorway exits), where they intend to assume or relinquish control. A minimal solution would leave the planning to the driver with no possibility to input the actual transition request ahead of time, which would have to be made manually once the planned point is reached. Such an implementation would be functionally identical to an implementation for unscheduled driver-vehicle transitions, where no planning/scheduling support is needed. Thus, when designing, first priority should be put on a working solution for unscheduled driver-vehicle solutions with possible additional support for more effective scheduled transitions.

Examples: Most implementations, both from academia and industry, focus on transitions from vehicle to driver. Examples for such transitions can be found in Funakawa, Ebina, Hegemann, Forster et al., and Melcher. Examples for implementations focusing on transitions from driver to vehicle can be found in Cullinane [14], Goldman-Shenhar [22], and Albert [23].

D. Pattern 4: Visual Driving Mode Indicators

Problem: In a semiautomated vehicle, the vehicle and driver essentially share the controls. In order to reduce the possibility of involuntary transition of control in either direction, it should at all times be clear who is in control and who is not.

Context: Demonstrators and simulator implementations are often limited in their temporal dimension, mostly focusing on the period briefly before and after the transition but not on a full journey in a vehicle. When the context is extended to a full journey from anywhere to ten minutes to several hours with more than one control transition in-between, an additional complicating factor arises: It might not at all times be clear who is in control of the vehicle. This phenomenon is particularly pronounced in shared or partial control scenarios, where e.g., activated adaptive cruise control can be indistinguishable from full automation. If, in such a scenario, the driver were to falsely assume that both horizontal and lateral automation are engaged, when in reality they are not, and actuates the steering wheel, the vehicle could leave its lane and cause an accident. Generally speaking, the longer a journey is and the more transition occur, the more likely it is that such a *mode confusion* might occur.

Solution: Beyond appropriate output to signal transition requests, a permanent indicator of the current driving mode is required. The driving mode indicator needs to be displayed permanently, so that its information can be accessed on demand. This permanency requirement means that one of the best ways to signal the current driving mode is via a visual indicator, which is less distracting than an auditory or haptic solution. It is not necessary for it to be in the driver's direct field of view while driving and can also be located in the middle console, on a side display, or anywhere else that is either in front or to the side of the driver, so that it can be reached with a quick glance.

The driving mode indicator needs to be expressive enough to meet the functionalities of the vehicle's automation, i.e., if there is only a binary distinction between fully manual or fully automated control, then the interface only needs to be able to display two different modes. If there are several mixed or shared modes, then the display must be able to clearly distinguish between these. Indicators that are effective and easy to implement are icons or color coding (or a combination thereof). In case of a binary system, showing the driving mode via a single icon as being displayed or not displayed is an acceptable solution. Visual indicators can be accompanied by text messages, although should ideally not consist of text only, as this takes longer to process cognitively.

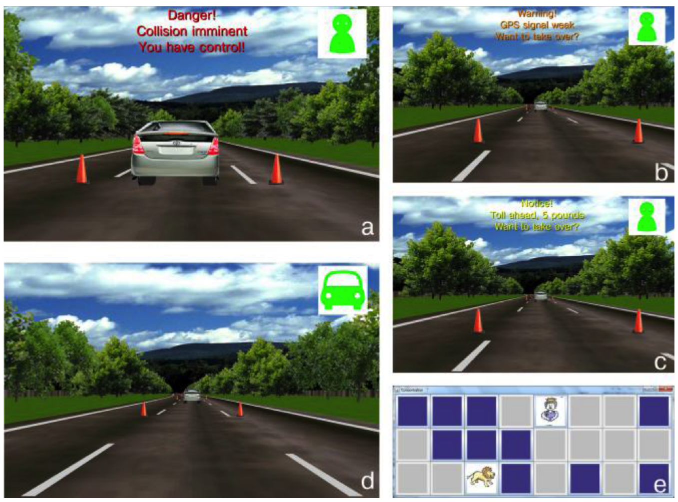

Figure 6. Color coded icon in upper right corner

Examples: Politis et al. [24] have implemented an icon-based driving mode indicator, which displays the current driving mode in the upper right corner (see Figure 6, a-d. The image displayed in e shows the secondary task tablet game and is not part of the mode indicator). In this setup, the transition requests are separate from the driving mode indicators and are displayed in the top center as color coded text messages.

Figure 7. Vehicle being shown as either on or off autodrive lane

The patent by Cullinane et al. [14] divides vehicle lanes into regular/manual and autodrive lanes. The current driving mode is indicated via constantly displaying the lane the vehicle is on and indicating, whether that lane is an autodrive lane or not. This is done via color coding the lanes and an additional text message at the top of the lane indicator (see Figure 7).

IV. Discussion

In the following, we discuss some interesting aspects and limitations of the presented patterns. One concerns the status of the solutions presented and whether or not they could be extended to or with antipatterns, the other the constant improvements in automation technology and associated difficulties of documenting working solutions in an endurable format.

A. Patterns and Antipatterns

All four of the solutions presented in this paper are presented as regular patterns, i.e., *working* solutions. However, while patterns 2 and 4 are clearly presenting working solutions without visible drawbacks, pattern 1 contains a certain antipattern potential. In the case of pattern 1, it is true that most implementations focus on physical input for good reasons and that a dual button setup is one of the more effective solutions to design a safe physical transition interface. The problem, however, lies in the premise of the superiority of physical controls for transition interfaces. As argued in our prior work [3], this is not necessarily the case and might, in fact, be one of the contributing factors to the often perceived impossibility of designing safe and effective control transitions for level 3. Thus, pattern 1 will require further research regarding the suitability of different interaction modes, which could eventually change the part about physical input into an antipattern, if it should turn out that physical input is one of the less suitable modes. Regardless, in its current form as a regular pattern, it properly represents the status quo in driver interaction design.

Pattern 3 is, in itself, presenting working solutions but might be best accompanied by an antipattern in the future. Beyond the priorities based on scheduled versus unscheduled transitions described in pattern 3, there is a strong trend towards putting emphasis and focus on designing vehicle-driver transitions, with driver-vehicle transitions often being an afterthought, if implemented at all. Similar to pattern 1, this limits the exploratory potential in interaction design and might be another contributing factor to the difficulty of designing for level 3 systems. If the design focus is only on in-time notifications for driver-vehicle transitions, this leaves out more nuanced approaches that could, e.g., allow the driver to plan transitions ahead of time and decide (perhaps even before beginning to drive), when to relinquish and assume control and plan their cognitive attention accordingly. Thus, exclusive focus on vehicle-driver transitions might be an antipattern, which is not reflected in the current iteration of pattern 3

B. A Rapidly Evolving Environment

The automotive industry is currently seeing rapid technical developments, which is especially pronounced regarding automated vehicle research and development. It becomes, thus, difficult to consider anything an established or proven solution that goes beyond automation level 2 (which includes advanced driving assistance systems such as adaptive cruise control or lane keeping assistance). Furthermore, even scoping the available technologies themselves is difficult in a comprehensive manner, as automation technologies of levels 3 and above are frequently limited to concept vehicles or industry showcases, which are open to a limited audience within an equally limited timeframe.

 ISBN: 978-1-61208-612-5

Thus, it is difficult to provide time-tested and proven solutions not only for the automotive domain in general but for automated driving technologies in particular. As the reviews and resulting patterns could show, it is still possible to extract commonalities and bring them into a pattern format even in this rapidly evolving subdomain. But it should be expected that such patterns will be outdated faster than patterns from other domains and that the pool of working solutions to draw from will be smaller. This does not mean that a pattern approach is to be considered impossible or unsuitable for interaction design in automated vehicles but the resulting patterns are likely more limited than they would be in a more static domain.

V. Conclusion and Future Work

In this paper, we presented commonalities of control transition interaction designs in level 3 automated vehicles, which had emerged in the process of a literature review and analysis of academic and industry implementations. These commonalities concerned (a) interaction modes to activate transitions or respond to transition requests, (b) a suitable visual metaphor for control transitions, (c) priority of transition direction depending on transition types, and (d) visual indicators to display driving modes. These results were presented in the form of four patterns, with the intention to make the information contained therein easier to access and (re-)apply. The patterns presented are initial versions, which have not yet undergone a full iterative cycle.

Future work will need to focus on refine and iterate the presented patterns further as well as extend the amount of patterns, as there are far more commonalities to be found in existing transition design implementations than what is covered by the patterns presented above. Maintaining an up to date pattern collection in the rapidly evolving automated driving domain will require a joint effort from within the automotive community and this initial pattern set shall serve as a functional basis for the automotive community to collect design best practices for interaction design in automated vehicles.

Acknowledgment

The financial support by the Austrian Science Fund (FWF): I 2126- N15 is gratefully acknowledged. The authors thank Magdalena Gärtner, Arno Laminger, Alexander Meschtscherjakov, Sandra Trösterer, Rod McCall and Fintan McGee for their contributions to the original work this publication is based on.

References

[1] I. SAE, “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems,” Standard J3016, 2016.

[2] Wired, “Ford’s skipping the trickiest thing about self-driving cars,” Online Article, October 2015, retrieved: January, 2018 from https://www.wired.com/2015/11/ford-self-driving-car-plan-google/.

[3] A. G. Mirnig, M. Gärtner, A. Laminger, A. Meschtscherjakov, S. Trösterer, M. Tscheligi, R. McCall, and F. McGee, “Control transition interfaces in semiautonomous vehicles: A categorization framework and literature analysis,” in Proc. of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI ’17). New York, NY, USA: ACM, 2017, pp. 209–220.

[4] N. H. T. S. Administration, “Federal automated vehicles policy,” Official Policy, 2016.

[5] C. Alexander, The Timeless Way of Building. New York, USA: Oxford University Press, 1979.

[6] C. Alexander, S. Ishikawa, and M. Silverstein, A Pattern Language: Towns, Buildings, Construction. New York, USA: Oxford University Press, 1997.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Pearson, 1994.

[8] A. Mirnig, T. Kaiser, A. Lupp, N. Perterer, A. Meschtscherjakov, T. Grah, and M. Tscheligi, “Automotive user experience design patterns: An approach and pattern examples,” International Journal On Advances in Intelligent Systems, vol. 9, 2016, pp. 275–286.

[9] J. O. Coplien, Software Patterns. New York, USA: SIGS Books, 1996.

[10] A. Mirnig and M. Tscheligi, “(dont) join the dark side: An initial analysis and classification of regular, anti-, and dark patterns,” in 9th Int. Conf. Pervasive Patterns and Applications (PATTERNS 2017). IARIA, 2017, pp. 65–71.

[11] S. Köhne, A Didactical Aproach towards Blended Learning: Conception and Application of Educational Patterns. *O*rig. title: Didaktischer Ansatz für das Blended Learning: Konzeption und Anwendung von Educational Patterns. Hohenheim, Germany: University of Hohenheim, 1995.

[12] K. Quibeldey-Cirkel, Design Patterns in Object Oriented Software Engineering. *O*rig. title: Entwurfsmuster: Design Patterns in der objektorientierten Softwaretechnik. Berlin, Germany: Springer, 1999.

[13] R. McCall, F. McGee, A. Meschtscherjakov, N. Louveton, and T. Engel, “Towards a taxonomy of autonomous vehicle handover situations,” in 8th Int. Conf. Automotive User Interfaces and Interactive Vehicular Applications. ACM, 2016, pp. 193–200.

[14] B. Cullinane, P. Nemec, M. Clement, R. Mariet, and L. Jonsson, “Engaging and disengaging for autonomous driving,” 2017, retrieved: January, 2018 from https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat\&docid=EP000002906453B1.

[15] M. Boehringer, S. Dubiella-Zimmermann, S. Jansen, J. Lockhart, E. Sengewald, and F. Stiebert, “Method and device for automatically operating a vehicle in an autonomous driving mode requiring no user action,” 2012, retrieved: January, 2018 from https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat\&docid=US020120277947A1.

[16] R. Gazit, “Steering wheels for vehicle control in manual and autonomous driving,” 2013, retrieved: January, 2018 from https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat\&docid=US020130002416A1.

[17] E. Coelingh and J. Ekmark, “Method and arrangement for allowing secondary tasks during semi-automated driving,” 2016, retrieved: January, 2018 from https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat\&docid=EP000003109119A1.

[18] M. Dikmen and C. M. Burns, “Autonomous driving in the real world: Experiences with tesla autopilot and summon,” in Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM, 2016, pp. 225–228.

[19] “Tesla S Official Press Materials,” retrieved: January, 2018 from https://teslamotors.box.com/pressfiles.

[20] A. P. van den Beukel, M. C. van der Voort, and A. O. Eger, “Supporting the changing driver’s task: Exploration of interface designs for supervision and intervention in automated driving,” Transportation research part F: traffic psychology and behaviour, vol. 43, 2016, pp. 279–301.

[21] H. van der Meulen, A. L. Kun, and C. P. Janssen, “Switching back to manual driving: How does it compare to simply driving away after parking?” in Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM, 2016, pp. 229–236.

[22] C. Goldman-Shenhar and E. Raphael, “Fahrzeug-benutzerinteraktionssystem,” 2017, retrieved: January, 2018 from https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat\&docid=DE102016117428A1.

[23] M. Albert, A. Lange, A. Schmidt, M. Wimmer, and K. Bengler, “Automated driving–assessment of interaction concepts under real driving conditions,” Procedia Manufacturing, vol. 3, 2015, pp. 2832–2839.

[24] I. Politis, S. Brewster, and F. Pollick, “Language-based multimodal displays for the handover of control in autonomous cars,” in Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM, 2015, pp. 3–10.

ISBN: 978-1-61208-612-5

Massively Parallel Optical Flow using Distributed Local Search

Abdelkhalek Mansouri *, Jean-charles Creput *, Fabrice lauri * and Hongjian Wang ‡
*Le2i FRE2005, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté
F-90010 BELFORT Cedex
France
Email: {abdelkhalek.mansouri, jean-charles.creput, fabrice.lauri}@utbm.fr
‡Experimental Radiation Oncology, Medical Faculty Mannheim - Heidelberg University
Theodor-Kutzer-Ufer 1-3, D-68167 Mannheim,
Germany
Email: hongjian.wang@medma.uni-heidelberg.de

***Abstract*—The design of many tasks in computer vision field requires addressing difficult NP-hard energy optimization problems. An example of application is the visual correspondence problem of optical flow, which can be formulated as an elastic pattern matching optimization problem. Pixels of a first image have to be matched to pixels in a second image while preserving elastic smoothness constraint on the first image deformation. In this paper, we present a parallel approach to address optical flow problem following the concept of distributed local search. Distributed local search consists in the parallel execution of many standard local search processes operating on a partition of the data. Each process performs local search on its own part of the data such that the overall energy is minimized. The approach is implemented on graphics processing unit (GPU) platform and evaluated on standard Middlebury benchmarks to gauge the substantial acceleration factors that can be achieved in the task of energy minimization.**

Keywords–Optical flow; Parallel and distributed computing; Variable neighborhood search; Graphics processing unit.

I. Introduction

Many image processing and computer vision problems can be formulated as optimization problems. Optical flow is a fundamental problem in this field. Despite the big progress since the first works of Horn-schunk [1] [2], optical flow remains very challenging [3].

Such optimization problem can be stated in a generic framework of image matching, including optical flow and stereo matching problems. It is worth noting that large classes of applications can be formulated as pattern matching operation between two graphs, such as locations of services, routing problems, and also the traveling salesman problem [4]. Because metaheuristic algorithms are time-consuming we deal with minimizing the corresponding energy functions in a massively parallel way. In order to address optical flow, we propose a distributed local search (DLS) algorithm following the idea of Verhoeven and the approach proposed in [5] [6] for stereo matching. The approach is a parallel formulation of a local search procedure in a partition of the data following standard local search metaheuristic.

Starting from a given initial solution, local search continually improves that solution at each iteration by searching it's neighborhood solution with lower cost, and then replacing it. The search stops when all solution neighbors are worse than the current solution, meaning a local optimum is reached. In the distributed local search approach proposed in this paper, image data is partitioned based on a regular decomposition of the plane between cells. We implement a parallel local search algorithm on GPU with the compute unified device architecture (CUDA) programming interface, under the concept of one cell to one thread. DLS runs many local search operations on different parts of the data in a distributed way to build a single solution that we obtain by gathering all the partial solutions from the different threads/cells. We apply the proposed DLS algorithm to resolve optical flow problem by minimizing the corresponding energy function, and we evaluate on the standard Middlebury benchmarks [8].

The rest of this paper is organized as follows. In section II, we present some of previous works on optical flow. In Section III, we formulate a general energy function to be equivalent to optical flow problem. In Section IV, we present the DLS algorithm in detail, providing basic data structures and operations, explaining local evaluation, and giving the details of GPU implementation. Experimental results are reported in Section V, before some conclusions are drawn in Section VI.

II. Related Work

Optical flow estimation has been improving steadily, since the work of Horn and Schunck [1]. We can see a progress clearly in the increasing quality of works on the Middlebury site for optical flow benchmark [8]. Most existing approaches follow the same spirit of the Horn and Schunck [1] formulation, which is based on differential form. An objective function is optimized which combines a data term that models matching between pixels and smoothness term that models how the flow is expected. In this part, we review some of the work related to optical flow estimation.

For better understand to the problem Simon et al. [9] provided a set of database and evaluation methods for optical flow algorithms. They contributed several types of data to inspect different types of optical flow algorithms. Yuri Boykov [10] showed that combinatorial optimization techniques are powerful tools for solving the problem, by developing algorithms based on graph cuts, that efficiently find a local minimum. His method works two to five times faster than any of the other methods, making near real-time performance possible. Tom et al. [11] proposes a global technique for minimizing non-convex, vector valued energy functions defined on Markov random fields using a functional lifting approach to resolve the problem. In [12], Lempitsky et al. put forward a new energy minimization approach for optical flow estimation called FusionFlow that combines the advantages of discrete and continuous optimization. Thomas et al. [13] describes an approach to estimate a dense optical flow field with almost the same high accuracy as known from variation optical flow by integrating rich descriptors into the variation optical flow setting.

Using cost volume filtering techniques become more and more interesting, since it efficiently achieves high-quality solutions for general multi-label problems. An interesting approaches is the Patch-Match filter for visual correspondence proposed by Lu et al. [14], due to its accurate result, computational efficiency and its ability to handle large textureless regions. In [15] the authors demonstrate a simple and powerful filtering approach for solving discrete labeling problems, that handles both fine motion structure and large displacements. For large displacement motion, we find the work of Bao et al. [16], who invested in the successes of local methods in visual correspondence searching as well as approximate nearest neighbor field algorithms to provide a fast approach that can handle the issues of fast running time with preserving the quality of the result. Classical coarse to fine framework [17] works also well for large displacement motions estimation, but it has some limitations with fine scale image structures, which may disappear in coarse scales. All these approaches are sequential approaches most often related on energy minimization framework. In this paper, we also address energy minimization providing a full parallel implementation of standard local search technique. We did not found yet such a direct application of combinatorial optimization local search technique to the optical flow problem.

III. Energy Formulation

Given a pair of images I and I', the goal of image matching is to assign each pixel $p = (x_p, y_p)$ a label l from the label set $L = 0, 1, ..., L-1$. A label l_p in visual correspondence represents a pixel moving from its regular position into the direction of its homologous pixel [18]. In the following sections, we will directly use the notations of labels as relative displacements, as usual with such problems. For optical flow problems considered here, $l = (u, v)$, where u and v correspond to the displacement in x and y directions.

We model the problem of optical flow estimation as an elastic grid matching problem. The two images are respectively represented by two graphs where edges correspond to the 4-connected image neighborhood. One graph is called matcher grid $G1 = (V_1, E_1)$ where a vertex is a pixel (from image I) with a variable location in the plane, the second is called the matched grid $G2 = (V_2, E_2)$ where vertices are pixels located in a regular grid (from image I'). The goal of elastic grid matching is to find the matcher vertex locations in the plane, so that the following energy function is minimized:

$$E(G_1, G_2) = \sum_{p \in V_1} D_p(p, c_p^{V_2}) + \lambda \cdot \sum_{\{p,q\} \in E_1} V_{p,q}(p - p_0, q - q_0) \quad (1)$$

where p_0 and q_0 are the initial locations of p and q respectively in a regular grid. Here, the first term of the energy function so called data term D_p expresses the data energy that measures how much assigning label f_p to pixel p disagrees with the data. In particular, it models how well the corresponding pixels of I and I' match. Traditionally, it is modeled based on the brightness (or color) constancy assumption. The second term $V_{p,q}$ is the smoothness term which favors certain flow fields over others. It expresses smoothness constraints on the labeling enforcing spatial coherence [18] [19] [20]. The energy function is commonly used for visual correspondence problems, and it can be justified in terms of maximum a posteriori estimation of a *Markov random field* (MRF) [21] [22].

IV. Energy Minimization

We use Distributed Local Search to minimize the energy function.

A. Distributed Local Search

There are many parallelization models for local search in literature. Following the original idea of Verhoven et al. [5] for the 2-opt local search in TSP, we adopt the method for image processing field and hence change the variable neighborhood search strategy. *Distributed local search* is a parallel formulation for local search algorithms [23] based on cellular decomposition of the Euclidean plane, in attempt to keep the principles of standard local search metaheuristic.

By the partition of a solution into a number of partial solutions, we implement several local search operations in a parallel way. Each operation acts on a part of the data to reach one partial solution. The data is partitioned according to a cellular decomposition. We assign one process to each cell in order to achieve local evaluation, perform neighborhood search, and select local improvement moves to execute. The many processes locally evolve the current solution into an improved one. The solution results from the many independent local search operations simultaneously performed on the distributed data in the plane. To exploit more potontial solutions, and escape local minima obtained by local search in most cases, we design different operators in a similar way to the *variable neighborhood search* (VNS).

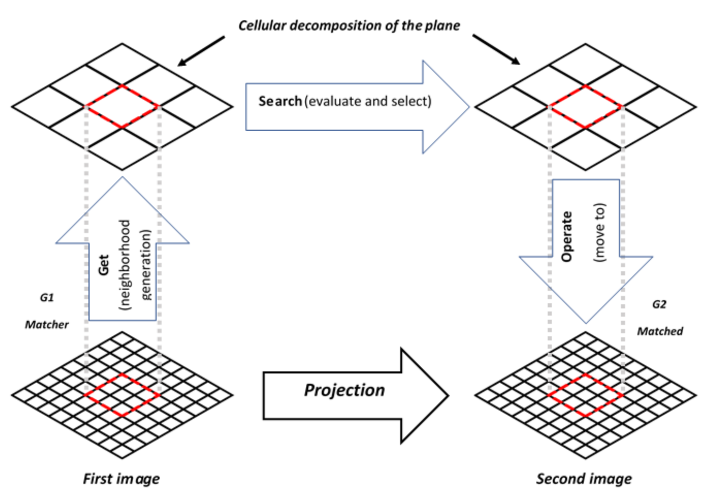

Figure 1. Distributed Local Search projection for visual correspondence problems through the cellular decomposition of the plane.

B. Data Structures and Basic Operations

The data structures and direction of operations for DLS algorithms are illustrated in Figure 1. As we mention before, we represent an image by a regular grid. The input data set is deployed on the pixel level. The image is partitioned between regular cells, such that a given cell corresponds to a zooming out of the image at a given level. The figure shows a zoom of level 1. Each cell is a basic processor that handles a local search processing iteration with the three following steps: neighborhood generation step (**get**), where we assign to each partial solution a set of partial local neighbors that can be reached within a single local search iteration; neighbor solution evaluation and selection of the best neighbor(**search**); then moving the matcher pixels toward the selected matched pixels (**operate**). Here, the neighborhood structure is directly defined by the cell structure. Each processor is responsible for moving the pixels inside its cell and only these pixels. The nature and size of specific moves and neighborhoods depend on the type of operator used and the size of the decomposition. The higher is the size, the larger is the local cell/neighborhood. The final solution is composed by the many partial solutions from the cells.

C. Local Evaluation with Mutual Exclusion

During the parallel execution of local search operators, many conflicts should occur, which affect the coherence of local evaluations when two neighbor threads evaluate and move the same pixel or two neighboring pixels at the same time. According to the cellular partition of the image, conflict operations could only happen on the cells frontiers.

In order manage cell frontiers during DLS , we propose a strategy called *dynamic change of cell frontiers* (DCCF), by which we fix the pixels on the cell frontiers, and only evaluate and move the pixels inside a cell. As the cell frontiers are fixed we guarantee the exclusive execution of the thread on the pixels inside its cell. But we still have to deal with the cell frontier pixels and make them participating in the optimization process. To do that we make the cellular decomposition of the plane dynamically changeable between each round of DLS operations. At different moments, the cellular decomposition slightly shifts on the input image in order to change the cell frontiers and consequently the fixed pixels.

D. Neighborhood Operators

In order to exploit more solution space, and escape the local minima that reached by local search, we design different neighborhood operators. Mainly we have two kinds of operators, small operators to move only a single pixel from the cell at each iteration, and large operators to simultaneously move a set of pixels inside a cell. Moving a pixel in a given neighborhood structure corresponds to changing its label in the corresponding labeling space.

Small Move Operators. In a move operation, if only one pixel moves, meaning that only one pixel's label is changed, this kind of move operation is called small move operation. We design two small move operators.

- *Local move operator*: It applies an increment/decrement to the current label of the considered pixel.
- *Propagation operator*: It takes the labels of the considered pixel's neighboring pixels, as the candidate labels, and it replaces the current label with the best one found in the propagation window.

Large Move Operators. They consider multiple pixels. We design several large move operators.

- *Random pixels move operator*: It randomly picks several pixels in the considered cell, and it assigns a same candidate label to these pixels. A parameter *pickedNumber* is set to control the number of pixels the operator randomly picks.
- *Random pixels jump operator*: It randomly picks several pixels in the considered cell, and it applies a same increment/decrement to the current labels of the considered pixels.
- *Random pixels expansion operator*: It randomly picks two groups of pixels, where pixels in the same group have the same label. Then, it "expands" the label of one group to the other, setting the labels of all the pixels in the second group with the same label as the first group. A parameter *maxPickedNumber* is set to control the max number of pixels the operator is allowed to randomly pick for each group.
- *Random pixels swap operator*: It picks pixels in the same way as the random pixels expansion operator does. Then, it "swaps" the labels of the two groups, setting the labels of all the pixels in the second group with the label of the first group, meanwhile setting the labels of all the pixels in the first group with the label of the second group.
- *Random window move operator*: It picks a fixed-sized window of pixels at a random position within the considered cell, and it assigns a same candidate label to all the pixels in this picked window. A parameter *pickedWindowRadius* is set to control the radius of the randomly picked window of considered pixels.
- *Random window jump operator*: It picks pixels in the same way as the random window move operator does, and it applies a same increment/decrement to the current labels of all the pixels in this picked window.

E. GPU Implementation Under VNS Framework

We use Compute Unified Device Architecture (CUDA) to implement the DLS algorithm on GPU platforms. The CUDA kernel calling sequence from the CPU side allows applications of different operators in the spirit of VNS and manages dynamic changes of cellular decomposition frontiers. According to our previous experiments, the repartition of tasks between host (CPU) and device (GPU) is actually the best compromise we found to exploit the GPU CUDA platform at a reasonable level of computation granularity. Data transfer between CPU side and GPU side only occurs at the beginning and the end of the algorithm. It is the CPU side that controls DLS kernel calls with different operators executed within the DCCF pattern for frontier cells management. With several neighborhood operators in hand, we use them under the VNS framework in order to enhance the solution diversification.

 ISBN: 978-1-61208-612-5

V. EXPERIMENTAL STUDY

We use DLS algorithm to compute Optical flow, viewing the task as an energy minimization problem, by using a simple energy function, applied to benchmark images from the widely used Middlebury optical flow data set [9]. The labels are the displacement, and the data costs are the absolute color differences between corresponding pixels. For the smoothness term in energy function, we use a truncated linear cost as the piecewise smooth prior defined in [19].

In Figure 2, there are reported the experimental results of optical flow using DLS on the Middlebury optical flow benchmark [9], with a set of choice of operators.

We test the *random window move* operator, the *random pixels move* operator, and the *random pixels expansion swap* operator respectively. The *random pixels expansion swap* operator combines the *random pixels expansion* operator and the *random pixels swap* operator together, selecting the best move of these two operators to act.

We tested also different combinations of operators, listed on the Figure 2

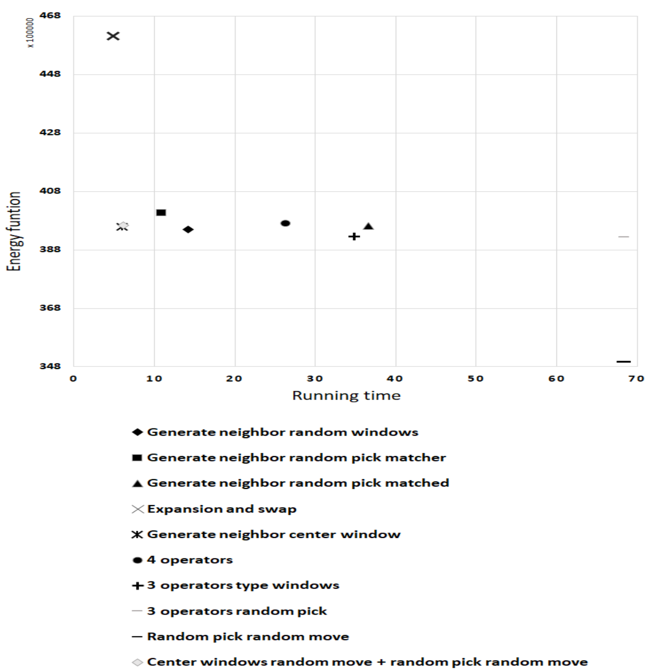

Figure 2. DLS with different operator combinations, on the Middlebury set of benchmark. The x-axis shows the running time and the y-axis shows the energy value. All the results are mean values over 10 runs.

Results show that the number of picked pixel has an essential role in the energy minimization process, since random large move operators lead to a faster convergences toward low energies, while small move operators lead to slower convergences. *Random pixels expansion swap operator* has more impact on running time than on the energy function. The *random pick random move operator* produces lowest energy with the longest execution time among all operators. Best compromises between energy and execution time are provided when we use a combination of operators.

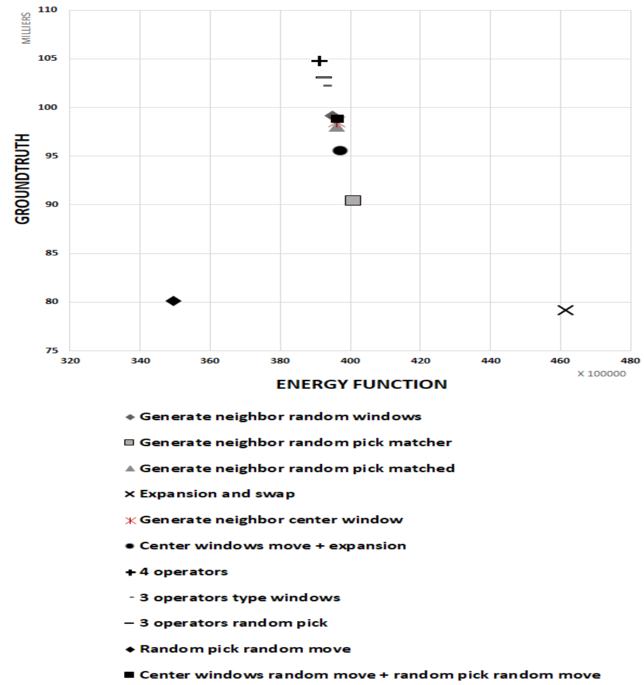

Figure 3. DLS with different operator combinations, on the Middlebury set of benchmark. The x-axis shows the energy function and the y-axis shows the ground truth error value. All the results are mean values over 10 runs.

In principle, more operators tend to lead to better diversification of solutions, hence lead to lower energies. This is supported by the result of the most combination. Another observation is that the combinations, which include the *random pixels expansion swap operator* are more likely to find faster result than other combinations. We think this is due to the highly stochastic nature of the random pixels expansion swap operator, where two randomly picked groups of pixels are considered.

In Figure 3, with the same choices of operators, results show the relation between the ground truth error and the energy function. The results show that random pixel operators yield to better quality than other operators, which make them a good choice, since they provide relatively lower energy with a short running time. In our experiments, the *random pixels expansion swap operator* gives good ground truth result, and the least execution time but with a higher energy value. At the same time the *random move random pick operator* gives a result as good as the *random pixels expansion swap operator*, but with the most execution time.

In Figure 4 are displayed the flow results for some of the Middlebury benchmarks. In the first and the second columns we present the input image and the ground truth respectively, in the third column is presented the DLS with *random pixel expansion and swap operator* which seems to be the best choice as an operator. we are also shown the results of the *random pick random move operator*, and the result of three operators combined in similar way to the VNS algorithm. Note that during our experiments, we deal with optical flow only as an energy minimization problem, just focusing on minimizing energies. The flow maps obtained from all the tested operators are the raw results after energy minimization, without any

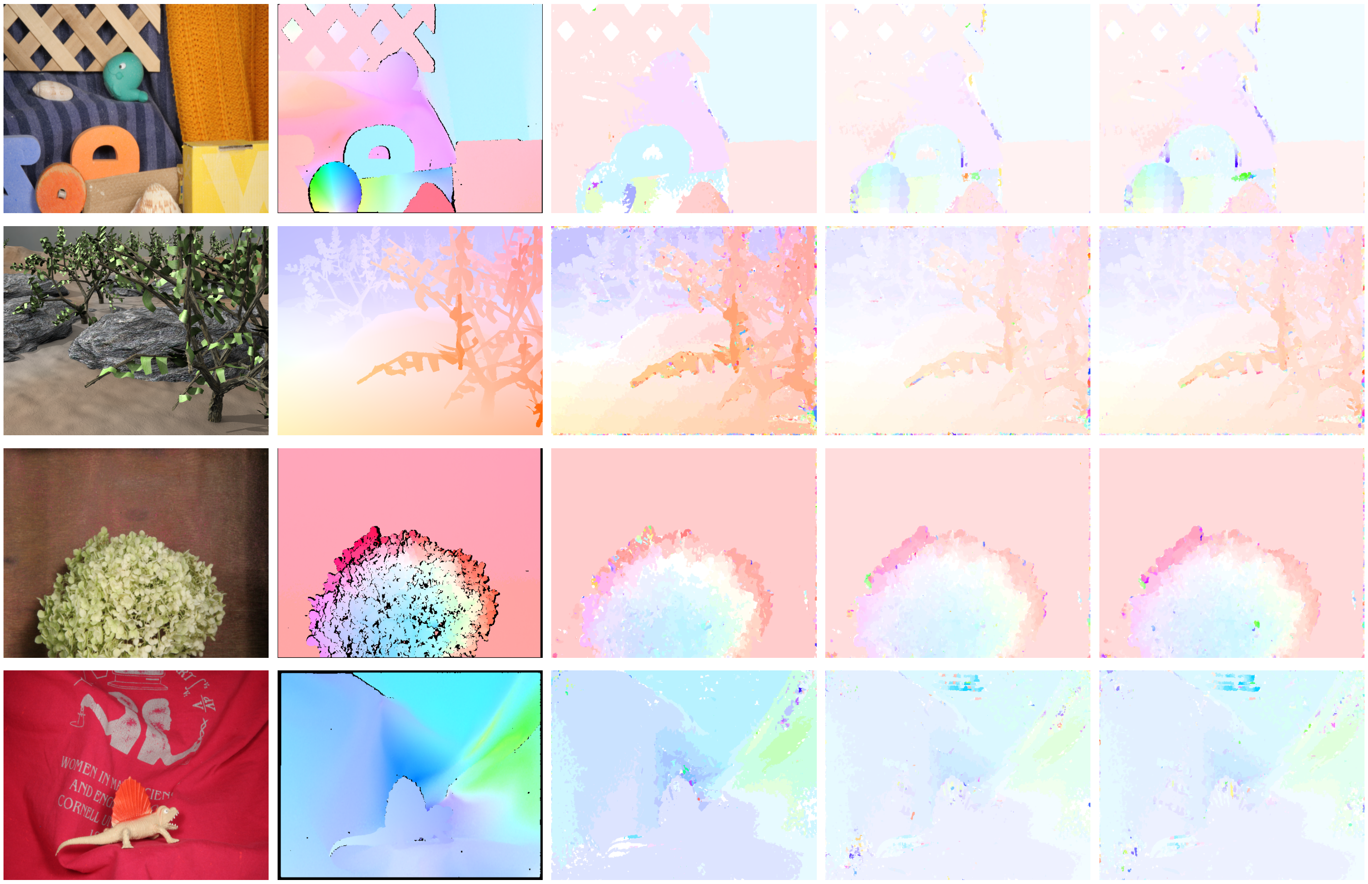

Figure 4. An example of flow result on Middlebury benchmark. (1st row) RubberWhale; (2nd row) Grove3; (3rd row) Hydrangea; (4th row) Dimetrodon; First column: input image ; second column: ground truth; third column: flow result with random pixels swap and expansion operator; fourth column: flow result with random pick random move operator; last column flow result with 3 operators (VNS)

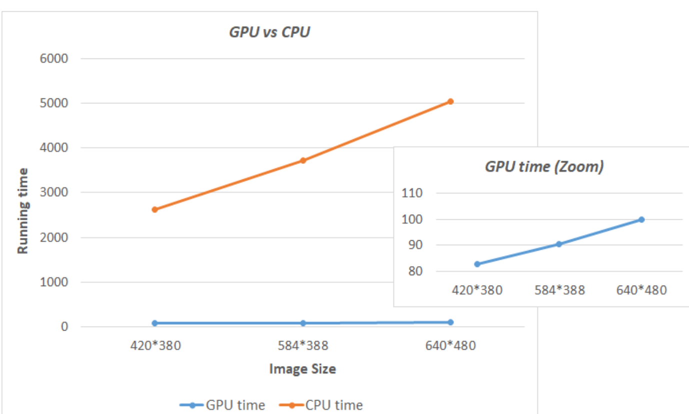

Figure 5. Comparison between DLS on GPU and DLS on CPU regarding input image size

additional post-treatments such as left-right consistency check, occlusion detection, which are all treatments specific to optical flow in order to minimize the errors compared with ground truth maps. Moreover, as pointed out in [18], the ground truth solution may not always be strictly related to the lowest energy.

In Figure 5, we focus on the performance of DLS when input size augments. We experiment on three images from Middlebury data set with different sizes. We can see that DLS-GPU has an acceleration factor which increases according to the augmentation of input size. This means that further improvement could be carried out only by the use of multi-processor platform with more effective cores. Figure 6 is a caption of the quantitative flow evaluation measured with average endpoint error(AEE) [9] in the Middlebury site.

VI. CONCLUSION

In this work, we introduced a new approach for optical flow estimation. By using combinatorial optimization, we have proposed a parallel local search procedure, called distributed local search. We have applied the algorithm to the well-known Middlebury optical flow benchmark. The result of the GPU implementation of DLS on optical flow is encouraging since we used a very simple energy function and without any post processing operations.

The main encouraging result is that the GPU implementation of DLS to optical flow provides an increasing acceleration factor as the instance size augments while allowing substantial minimization of the energy. That is why we hope for further improvements or improved accelerations factors with the availability of new multi-processor platforms with more and more independent cores.

 ISBN: 978-1-61208-612-5

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive flow [45]	106.0	0.36 112	0.59 105	0.37 113	1.21 111	1.60 105	1.23 111	1.21 110	1.77 105	1.18 112	0.94 105	2.03 105	0.97 104
FFV1MT [107]	106.4	0.33 110	0.64 109	0.24 106	0.79 103	1.90 113	0.64 101	1.33 114	1.90 111	1.23 113	1.38 113	2.98 114	1.29 112
SLK [47]	106.5	0.30 108	0.70 113	0.36 112	1.09 109	1.77 109	1.21 110	1.25 112	1.98 114	1.03 110	1.56 116	2.26 110	1.71 116
DLS-OF [117]	108.0	0.27 105	0.68 112	0.24 106	1.64 115	1.75 108	2.05 116	1.05 102	1.49 94	1.60 116	0.77 102	2.00 103	0.80 103
Periodicity [78]	108.6	0.31 109	0.78 117	0.20 100	1.54 114	2.62 117	1.71 113	1.86 117	2.00 115	1.66 117	1.15 110	3.05 116	1.07 107
PGAM+LK [55]	110.1	0.37 113	0.70 113	0.59 116	1.08 108	1.89 111	1.15 108	0.94 98	1.59 99	0.88 104	1.40 114	3.28 117	1.33 113
FOLKI [16]	111.5	0.29 107	0.73 115	0.33 111	1.52 113	1.96 114	1.80 114	1.23 111	2.04 116	0.95 107	0.99 106	2.20 107	1.08 109
Pyramid LK [2]	114.3	0.39 115	0.61 106	0.61 117	1.67 116	1.78 110	2.00 115	1.50 116	1.97 113	1.38 114	1.57 117	2.39 111	1.78 117

Average endpoint error	Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive flow [45]	1.20 103	1.57 103	1.08 88	1.73 103	1.90 89	1.12 106	0.59 116	0.37 116	1.37 116	1.37 103	2.16 101	1.81 101
FFV1MT [107]	1.76 115	1.99 115	2.45 115	2.33 114	3.64 116	1.72 115	0.16 69	0.18 90	0.27 61	1.81 109	2.64 109	2.27 107
SLK [47]	1.54 113	1.82 112	2.14 114	2.02 109	2.79 112	1.36 109	0.17 78	0.16 72	0.26 55	2.43 114	3.18 114	3.31 114
DLS-OF [117]	1.15 100	1.57 103	1.07 87	2.45 115	2.68 110	1.47 112	0.46 114	0.32 115	1.21 114	2.26 113	2.86 113	4.28 115
Periodicity [78]	5.17 117	6.79 117	4.19 117	3.79 117	5.26 117	2.93 117	0.12 25	0.18 90	0.36 93	2.67 115	5.01 116	3.18 113
PGAM+LK [55]	1.37 111	1.70 110	1.67 110	2.10 110	2.53 106	1.39 111	0.36 113	0.28 114	0.65 109	1.89 112	2.72 112	2.71 111
FOLKI [16]	1.53 112	1.85 113	2.07 113	2.14 111	3.23 115	1.60 113	0.26 109	0.21 105	0.68 110	2.67 115	3.27 115	4.32 116
Pyramid LK [2]	2.94 116	3.72 116	2.98 116	3.33 116	2.74 111	2.43 116	0.30 110	0.24 112	0.73 112	3.80 117	5.08 117	4.88 117

Figure 6. DLS optical flow performance on the Middlebury data set [9].

It is a well-known fact that the minimum energy level does not necessarily correlate to the best flow field. Here, we only address energy minimization discarding too much complex post-treatments necessary for the "true" ground truth flow field. It should follow that many tricks are certainly not yet implemented to make energy minimization coincide to ground truth evaluation. In order to improve the flow quality in terms of minimizing the errors to ground truth only, specially designed terms for addressing typical situations in vision, such as occlusion, slanted surfaces, and the aperture problem, need to be added in the formulation of energy function. Furthermore, more complex post-treatments should also be considered to complete the parallel energy minimization method.

References

[1] B. K. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, vol. 17, no. 1, 1981, pp. 185 – 203, URL: http://www.sciencedirect.com/science/article/pii/0004370281900242/. [accessed: 2018-01-15].

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679, URL: http://dl.acm.org/citation.cfm?id=1623264.1623280 [accessed: 2018-01-10].

[3] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," International Journal of Computer Vision, vol. 106, no. 2, Jan 2014, pp. 115–137, URL: https://doi.org/10.1007/s11263-013-0644-x [accessed: 2018-01-15].

[4] H. Wang, N. Zhang, and J.-C. Crput, "A massively parallel neural network approach to large-scale euclidean traveling salesman problems," Neurocomputing, vol. 240, no. Supplement C, 2017, pp. 137 – 151, URL: http://www.sciencedirect.com/science/article/pii/S0925231217303326 [accessed: 2018-01-10].

[5] M. G. A. Verhoeven, E. H. L. Aarts, and P. C. J. Swinkels, "A parallel 2-opt algorithm for the traveling salesman problem," Future Gener. Comput. Syst., vol. 11, no. 2, Mar. 1995, pp. 175–182, URL: http://dx.doi.org/10.1016/0167-739X(94)00059-N [accessed: 2018-01-10].

[6] H. Wang, A. Mansouri, and J.-C. Crput, "Cellular matrix model for parallel combinatorial optimization algorithms in euclidean plane," Applied Soft Computing, vol. 61, no. Supplement C, 2017, pp. 642 – 660, URL: http://www.sciencedirect.com/science/article/pii/S1568494617304970 [accessed: 2018-01-12].

[7] C. R. Reeves, Ed., Modern Heuristic Techniques for Combinatorial Problems. New York, NY, USA: John Wiley & Sons, Inc., 1993.

[8] R. S. Daniel Scharstein. The middlebury computer vision pages. URL: http://vision.middlebury.edu/flow [accessed: 2018-01-15].

[9] B. S. et al, "A database and evaluation methodology for optical flow," International Journal of Computer Vision, vol. 92, no. 1, Mar 2011, pp. 1–31, URL: https://doi.org/10.1007/s11263-010-0390-2 [accessed: 2018-01-10].

[10] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, Sep. 2004, pp. 1124–1137, URL: http://dx.doi.org/10.1109/TPAMI.2004.60 [accessed: 2018-01-15].

[11] T. Goldstein, X. Bresson, and S. Osher, "Global minimization of markov random fields with applications to optical flow," Inverse Problems and Imaging, vol. 6, no. 4, 2012, pp. 623–644, URL: http://aimsciences.org/journals/displayArticlesnew.jsp?paperID=7942 [accessed: 2018-01-12].

[12] V. Lempitsky, S. Roth, and C. Rother, "Fusionflow: Discrete-continuous optimization for optical flow estimation," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, June 2008, pp. 1–8.

[13] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, March 2011, pp. 500–513.

[14] J. Lu, H. Yang, D. Min, and M. N. Do, "Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in 2013 IEEE Conference on Computer Vision and Pattern Recognition, June 2013, pp. 1854–1861.

[15] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 3017–3024, URL: http://dx.doi.org/10.1109/CVPR.2011.5995372 [accessed: 2018-01-12].

[16] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 3534–3541, URL: http://dx.doi.org/10.1109/CVPR.2014.452 [accessed: 2018-01-12].

[17] L. Alvarez, J. Weickert, and J. Sánchez, "Reliable estimation of dense optical flow fields with large displacements," International Journal of Computer Vision, vol. 39, no. 1, Aug 2000, pp. 41–56, URL: https://doi.org/10.1023/A:1008170101536 [accessed: 2018-01-12].

[18] R. S. et al, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 6, 2008, pp. 1068–1080.

[19] O. Veksler, "Efficient graph-based energy minimization methods in computer vision," Ph.D. dissertation, Cornell University, 1999.

[20] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 11, 2001, pp. 1222–1239.

[21] J. Besag, "On the statistical analysis of dirty pictures," Journal of the Royal Statistical Society. Series B (Methodological), 1986, pp. 259–302.

[22] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," Pattern Analysis and Machine Intelligence, IEEE Transactions on, no. 6, 1984, pp. 721–741.

[23] H. Wang, "Cellular Matrix for Parallel K-means and Local Search to Euclidean Grid Matching," Theses, Université de Technologie de Belfort-Montbéliard, Dec. 2015, URL: https://tel.archives-ouvertes.fr/tel-01265951/ [accessed: 2018-01-10].

Improved Gas Detection Algorithm for FTIR-Based Hyperspectral Imaging System Using Normalized Matched Filter

Hyeong-Geun Yu, Jae-Hoon Lee, Dong-Jo Park
School of Electrical Engineering
Korea Advanced Institute Science and Technology
291 Daehak-Ro, Yuseung-Gu, Daejeon, Republic of Korea
elloss@kaist.ac.kr, ljhinlove@kaist.ac.kr,
djpark@kaist.ac.kr

Hyeon-Woo Nam, Byeong-Hwang Park
CRB Defense Technology Directorate,
Agency for Defense Development,
P. O. Box 35, Yuseung-Gu, Daejeon, Republic of Korea
hyunwoonam@add.re.kr, pbh1348@add.re.kr

***Abstract*— A Hyperspectral Imaging System (HIS) can be used to detect a harmful gaseous plume from a long distance. Each pixel in the HIS can obtain a radiance spectrum. The hazardous gaseous plume has a unique pattern expressed in the radiance spectrum. A Normalized Matched Filter (NMF) is an algorithm extracting the absorption pattern from the radiance spectrum. We propose to apply a NMF algorithm into the brightness temperature spectrum in order to detect chemical warfare agents (CWAs). Simulation results demonstrate that the proposed algorithm significantly improves the detection performance.**

Keywords- Gas detection, Hyperspectral Image.

I. INTRODUCTION

Passive remote sensing using a Hyperspectral Imaging System (HIS) based on Fourier Transform InfraRed (FTIR) spectrometers is known as a key technology to detect hazardous gases in the atmosphere. The HIS, which consists of an FTIR spectrometer and a focal plane array detector, can provide not only spectral information but also spatial information. Each pixel in the HIS can obtain the radiance spectrum for the corresponding field of view (FOV) from a standoff distance.

Generally, the hazardous gaseous plume present in the atmosphere has a specific radiance spectrum pattern. The spectrum measured by the HIS is a combined spectrum of the gaseous plume and the background. The presence of the gaseous plume can be determined by extracting the absorption pattern of the target gas from the radiance spectrum measured by the HIS instrument [1]. A Normalized Matched Filter (NMF) can be used to extract the absorption pattern from the measured radiance spectrum and detecting the gaseous plume [2].

In this paper, we propose an algorithm that transforms the measured radiance spectrum into brightness temperature and adapts the NMF to the brightness temperature spectrum. Since in the brightness temperature domain, the background spectrum is constant, it is easier to extract the target pattern.

The rest of this paper is organized as follows. Section II describes transforming a radiance spectrum into a brightness temperature spectrum. Section III addresses the proposed normalized matched filter. Section IV presents the experimental results. The acknowledgement and conclusions close the article.

II. BRIGHTNESS TEMPERATURE SPECTRUM

The measured spectrum for each pixel obtained from the HIS equipment is the radiance spectrum. The envelopes of radiance spectra are different according to the temperature of the gaseous plume and the background. The radiance spectrum generally does not have constant baseline. It is difficult to extract the pattern of gases from the radiance spectrum by these reasons. On the other hand, the brightness temperature spectrum of the background is nearly constant in the range of 925 – 1440 cm^{-1} where most Chemical Warfare Agents absorb [1]. For these reasons, we transform the radiance spectrum into the brightness temperature spectrum $T_{meas}(v)$ using Plank's function, as follows:

$$T_{meas}(v) = \frac{hcv}{\ln\left(\frac{2hc^2v^3}{L_{meas}(v)}+1\right)k}, \quad (1)$$

where h is Plank's constant, c is the speed of light, k is Boltzmann's constant, v is the wavenumber, and $L_{meas}(v)$ is the measured spectral radiance for the wavenumber.

III. NORMALIZED MATCHED FILTER

Let H_0 and H_1 denote the absence and presence of a target gas, respectively. The measured brightness temperature spectrum $\mathbf{x}$ is described as the following two hypotheses:

$$\mathbf{x} = \begin{cases} \mathbf{v}, & H_0, \\ \mathbf{s}g + \mathbf{v}, & H_1, \end{cases} \quad (2)$$

where $\mathbf{s}$ is the target gaseous absorption spectrum, $\mathbf{v}$ is the background clutter, i.e., $\mathbf{v} \sim N(\mathbf{m}_b, \mathbf{C}_b)$ and g is the amount of a gas. $\mathbf{m}_b$ and $\mathbf{C}_b$ are denoted as mean and covariance of background clutter. Using the Generalized

Likelihood Ratio Test (GLRT) approach [2], we obtain the NMF detector

$$T_{NMF} = \frac{\left(\tilde{\mathbf{s}}^T\tilde{\mathbf{x}}\right)^2}{\left(\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}\right)\left(\tilde{\mathbf{x}}^T\tilde{\mathbf{x}}\right)} \underset{H_1}{\overset{H_0}{\gtrless}} \lambda \tag{3}$$

where $\tilde{\mathbf{s}} = \mathbf{C}_b^{-1/2}(\mathbf{s}-\mathbf{m}_b)$ and $\tilde{\mathbf{x}} = \mathbf{C}_b^{-1/2}(\mathbf{x}-\mathbf{m}_b)$. If T_{NMF} is larger than a detection threshold λ, the decision is H_1, otherwise it is H_0.

IV. SIMULATION RESULT

The data used for the experiments were obtained with a FTIR passive remote-sensing equipment, HI-90 by Bruker Optics. It can provide a datacube with a spectral resolution of 4 cm^{-1} in the spectral range of 900 ~ 1260 cm^{-1} and a spatial resolution of 128×128 pixels at a high frame rate.

Figure 1 shows radiance spectra (a) and the brightness temperature spectra (b) of each pixel of the hyperspectral image at the background and a sulfur hexafluoride (SF_6) gaseous plume filled in a gas cell. The figure shows that a particular absorption pattern of SF_6 at a spectral range near 950 cm^{-1}. the absorption pattern appears more prominently in the brightness temperature spectrum than in the radiance spectrum.

We compare our proposed algorithm with the conventional gas detection algorithm, which uses the NMF to the radiance spectrum. Figure 2 depicts the Receiver Operating Characteristic (ROC) curve, which presents the detection performance of the algorithm. The proposed algorithm has better performance than the conventional algorithm because the brightness temperature spectrum is better than the radiance spectrum for finding the absorption pattern of a gas by applying the NMF.

Figure 3 shows the detection results of the proposed algorithm and the conventional algorithm. We set the detection thresholds λ of two algorithms so that the probability of false alarm P_{fa} is 0.005.

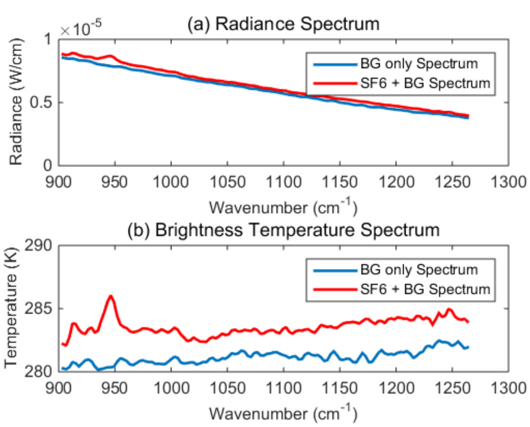

Figure 1. Radiance spectra (a) and brightness temperature spectra (b) of a gas plume and the background

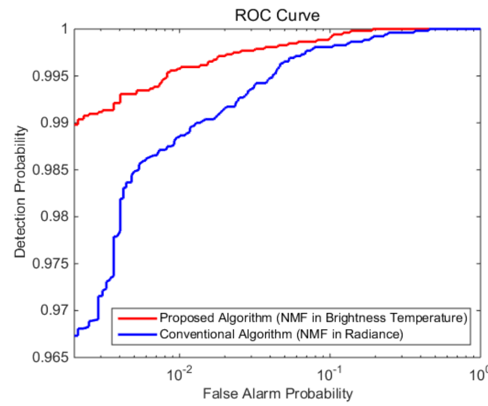

Figure 2. ROC curve of proposed algorithm and conventional algorithm

In case that the proposed algorithm detects gas at the pixel, the green color is mapped. In case that the both algorithms detect gas at the pixel, it is expressed in red color. We can see that the NMF in the brightness temperature spectrum is better than the NMF in the radiance spectrum.

V. CONCLUSIONS

We proposed a hazardous-gaseous plume detection algorithm for the FTIR-based HIS. First, the measured radiance spectra are transformed into the brightness temperature spectrum using Plank's function. Gas detection is performed by applying the NMF to the brightness temperature spectrum. The proposed algorithm outperforms the existing algorithm which applies the NMF to the radiance spectrum.

ACKNOWLEDGMENT

This work was supported by the Agency for Defense Development of the Republic of Korea.

REFERENCES

[1] R. Harig and G. Matz, "Toxic Cloud Imaging by Infrared Spectrometry: A Scanning FTIR System for Identification and Visualization," Field Anal. Chem. Technol., vol. 5, no. 1–2, pp. 75–90, 2001.

[2] D. G. Manolakis, S. E. Golowich, and R. S. DiPietro, "Long-Wave Infrared Hyperspectral Remote Sensing of Chemical Clouds: A Focus on Signal Processing Approaches," IEEE Signal Process. Mag., vol. 31, no. 4, pp. 120–141, Jul. 2014.

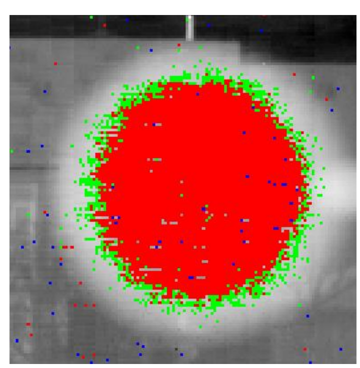

Figure 3. Detection results of the proposed algorithm

Evolvable Documents – an Initial Conceptualization

Marek Suchánek and Robert Pergl

Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
Email: marek.suchanek,robert.pergl@fit.cvut.cz

***Abstract*—We may say that documents are one of the cornerstones of our civilization. Information technologies enabled unparalleled flexibility and power for retrieving, storing, and sharing documents. However, in a daily documents-intensive job, one needs to deal with severe complications of documents evolvability and reusability of their parts. Maintaining consistency across several documents and their versions is typically a tedious and error-prone task. Similar evolvability challenges have been dealt with in software engineering and principles such as modularity, loose coupling, and separation of concerns have been studied and applied. There is a hypothesis that they may help in the domain of evolvable documents, as well. We perceive devising a conceptualization of documents as the first step in this endeavor. In this paper, we present a generic conceptualization leading to evolvable documents applicable in any documentation domain, and we propose next steps.**

Keywords–Electronic documents; Evolvability; Modularity; Conceptualization; Separation of Concerns.

I. Introduction

Documents are a vital carrier for storing and distributing knowledge - the precious result of various human activities. The amount of documents grows rapidly primarily thanks to their "cheapness" in the digital era. However, an interesting observation may be made: In spite of various means of storing, retrieving, and sharing documents in electronic forms, the foundations did not change, and the documents are the same hard-to-maintain and evolve structures as they always were. Imagine for example a document capturing regulations of a study program enrollment. Such a document is issued and maintained by the Dean of a faculty. However, it must be compliant with the university's regulations document, which in turn must be compliant with the regulations of the Ministry of Education. We have three levels of documents where the more specific ones contain parts of the more general ones, take them as-is or elaborate more specific versions, add further regulations, etc. Now, imagine that there is a change in the Ministry's regulations, which must be appropriately dealt with in the referring documents. This situation affects at least tens of Faculty's agendas which results in inefficiency, inconsistency, and other related problems.

The first observation is that documents are seen as monolithic wholes or wholes composed of highly coupled parts which cannot be separated or even reused. If we would be able to decouple parts of documents, make them loosely coupled just by higher concerns, and design them as reusable, it would significantly help in many domains, such as teaching materials, corporate documents, manuals, regulations, etc. The practice of software engineering suggests that if done properly, evolvability may be significantly improved, the efficiency of document management gained, and error rate decreased [1].

In Section II, we first briefly introduce a wide variety of related work affecting documents domain in terms of the modularity and evolvability. Section III is divided into three steps of our approach to creating a generic conceptualization, i.e., independent on a type of enterprise or domain involved. We apply concepts from theories used in computer science and software engineering verified by practice. Furthermore, we build our approach on the Normalized Systems (NS) theory [1], which is dealing with evolvability of information systems and it has been reported to be successfully applied in other domains than software development including documents [2]-[4]. In Subsection III-A, we split the domain into key parts and then, in Subsection III-B, we introduce conceptual models for them using the ontologically well-founded conceptual modeling language OntoUML [5]. After this exploratory and inductive part, Subsection III-C contains deduced possible and potentially suitable next steps and future work.

II. Related Work

Over the years of Information and Communication Technologies (ICT) field development, many solutions for working with documents and documentation emerged [6]. In this section, we briefly discuss some key areas and approaches related to electronic documents. This review of the current state-of-the-art provides a foundation for our conceptualization of the documents problem domain in general.

Nowadays, there are many different text processing tools, syntaxes and complex systems for dealing with documents within their whole life-cycle [6]. The goal of this part is not to describe particular existing solutions, but to briefly emphasize essential and interesting approaches or ideas that should be considered before developing new solutions. All of the mentioned approaches strife to make dealing with documents simpler and more effective.

A. Formats and Syntax

There is a plethora of markup languages and ways how to encode a document providing different advantages, some are focused to be easily readable in plain text, and others provide ways to encode complex document elements [6]. An interesting concept of versatility and evolvability is represented by the Pillar markup language for the Pharo environment [7]. It consists of a document model which is easily extensible by implementing new classes and visitors defining syntactic constructs meaning and handling. Furthermore, the provided tool allows export in many other formats and markups.

 ISBN: 978-1-61208-612-5

Converting between formats is also important to mention. A great example of a markup converter is Pandoc, which enables to convert from over 20 formats to more than 30 formats [8]. The lower number of input formats illustrates the fact that some of them are harder to process. At the same time, an output format of a document should be expressive and extensible. For example, LaTeX has mechanisms of custom packages and commands, environments, and macros. It is then a considerable challenge to convert it to another format lacking these extensions [9].

B. Templates and Styles

Separation of a graphical design and content is the first notion of separation of concerns in documents. A document, or any piece of data in general, can be rendered using an independent template associated with one or various styles. This approach can be seen in many documentation systems and languages, such as Extensible Markup Language (XML), HyperText Markup Language (HTML) and Cascading Style Sheets (CSS), LaTeX or even in various *What you see is what you get* (WYSIWYG) Office suites. This separation leads to good evolvability of document structure and style without touching the content itself. [6][9]

Using templates with styles to easily form and design complex structures is well observable in the field of web development. Many web frameworks are supplied with one of many template engines, namely, Twig, Jinja, JavaServer Pages, Mustache, or other. Template engine takes structured data and a template as input and then produces a rendered document, e.g., query result in the form of HTML document with table or JavaScript Object Notation (JSON) array based on the request. Moreover, it is usually possible to extend and compose templates together, and to create reusable components and macros. [10]

C. Sharing and Collaboration

Documents are often written by more than one person. Collaboration possibilities are related to the format used. If the document files are in plain text, then one of the solutions is to use Git or other version control system (VCS) [11]. There are also many cloud services allowing users to create and edit documents collaboratively, for instance, Google Documents, Overleaf, or Microsoft Office Online.

When mentioning Git and other VCSs, it is important to emphasize that it already provides a lot of functions that a powerful document system needs [11][12]. Such features are among others:

- tracking of history and comparing changes of version,
- tagging a specific version,
- signing and verifying changes,
- looking up who changed a particular line of text,
- working with multiple sources/targets and linking other projects submodules,
- logging and advanced textual or binary search within the changes,
- allowing changes in multiple branches,
- merging or combining changes.

Moreover, services like GitLab, GitHub, or BitBucket provide more collaborative tools for issues, change reviews, project management, other services integrations. One of important related services type is continuous integration (CI), which allows to build, check, and distribute results seamlessly. It can be used for example to compile the LaTeX document and send the Portable Document Format (PDF) to a file server or email address. [12][13]

D. Document Management Systems and Wikis

A document management system (DMS), as explained in [6] and [14], is an information system that is able to manage and store documents. Most of them are capable of keeping a record of the various versions created and modified by different users. The term has some overlap with the notion of content management systems. It is often viewed as a component of enterprise content management (ECM) systems and related to digital asset management, document imaging, workflow systems and records management systems.

One of the leading current DMS is an open source system Alfresco that provides functionality, such as storing, backing up, archiving, but also ISO standardization, workflows, advanced searching, signatures and many others [15]. From our perspective, the problem is that DMSs are mainly focused just on working with a document as a whole – documents stored as files with rich metadata, which doesn't contribute to an evolvability itself.

Knowledge can be gathered, formatted, and maintained in a wiki – a website allowing users collaboratively modify content and structure directly from the web browser [16]. Wikis are extensible and simple-to-use sets of pages that can be edited via a WYSIWYG field or manually with some simple or custom syntax, e.g., Markdown, reStructuredText, or DokuWiki. The system keeps track of changes within pages as well as the attachments, so it enables to compare differences and see who and when changed the document. Common extensions of wikis are tools for exporting to various formats or extending syntax and other user-friendly functionality [17]. There are many diverse commercial and open-source solutions with slightly different functionality. Commercial solutions are often called enterprise content management and consist of a wiki system and a DMS to manage documents in a better way than just with a plain DMS [14].

E. The Normalized Systems Theory

The Normalized Systems theory [1] deals with modularity and evolvability of systems and information systems specifically. It introduces four principles in order to identify and eliminate combinatorial effects (i.e., dependencies that are increasing with the system size): Separation of Concerns, Data Version Transparency, Action Version Transparency, and Separation of States. Applying the principles leads to evolvable systems composed of fine-grained and reusable modules. In the documents domain, only the first two principles are applicable [4], because actions and states are workflow-related.

The principles and concepts of the theory have been reported to be used in other domains, such as study programs [2] and documents [3]. In the paper [4], it is shown in a form of the prototype, how the theory can be used (especially the separation of concerns and creating modular structures) in the domain of documents for study programs. The prototype is able to combine selected fine-grained independent modules and to generate a resulting LaTeX document.

 ISBN: 978-1-61208-612-5

F. Source Code Documentation

Basically, for every widely-used programming language, there are one or more systems for building a documentation from annotations and comments that are placed directly in a source code. Such systems are, e.g., Javadoc for Java, Doxygen for C/C++, Sphinx for Python (see Listing 1), or Haddock for Haskell.

The fundamental idea is to place parts of documentation directly into a documented artifact (a variable, a function, a class, a module, a source file, etc.). The resulting documentation is as modular and evolvable as the writer creates it according to guidelines and with *Don't Repeat Yourself* (DRY) principle. It is then indeed easy to edit just a part of documentation related to one concern if the concern is separated in the source code. Another observation is that such documentation is composed of reusable parts. Linking the source file to different project results in its inclusion to a documentation of a different project, too. [18]

Listing 1. Documentation of Python source code

```
class Person:
    """This is simple example Person class

    You can create new person like this:

    .. code::

       bd = datetime.datetime(1902, 1, 1)
       p = Person("Peter Pan", bd)

    :ivar name: Full name of the person
    :vartype name: str
    :ivar birthdate: Birthdate of the person
    :vartype birthdate: datetime
    """

    #: Number of people instantiated
    people = 0
    ...

    @property
    def age(self):
        """Age of the person (birthdate-based)"""
        t = date.today()
        b = self.birthdate
        return self._age_diff(t, b)
```

III. Our Approach

Our approach to investigate and understand the problem domain of evolvable documents is to split it into *four separate key areas* and to build conceptual models of the domain in ontologically rich language OntoUML based on them. Next, we suggest possible solutions which can be based on them and could lead to improvement of documents evolvability.

A. Key Document Viewpoints

After the brief overview of current approaches in the ICT support for documents, this section introduces various key viewpoints that are not ICT-related but are typical for documents in any form. Each of them is shortly described, and a possible implication in the computer science domain follows. The viewpoints are defined with respect to the semiotic ladder that introduces several steps from social world to physical world: pragmatics, semantics, syntactics, and empirics [19].

Pragmatics and semantics, that are related to the meaning and intentions, are covered within first three subsections. Syntactics is related to the last subsection called Structure. Encoding the document in the physical world, as other parts of empirics, is out of the this work's scope.

1) Meaning: Apparently, the meaning is the key part of a document, as the purpose of the document is to store and carry a piece of information that can be retrieved in the future [20]. As the triangle of reference says, the meaning is encoded in symbols of some language via concepts [21]. The common problem is that in case of documents, the language is a natural language. Because of that, documents are hard to be understood by computers effectively in the sense of their true meaning. Advanced methods in data mining and text processing disciplines are trying to address this [22]; however sometimes the meaning is hard to be decoded even by human beings themselves...

Meaning, purpose, concern and other content information may be provided as metadata of the document. Considering such metadata, there should be a simple, single and flexible model for such description of documents for an easy automated processing. [23][24]

If a meaning of a text is captured in a machine-readable way, then it is possible to extract desired information, compare the meaning of different documents, find logical dependencies, and many others with an automated processing. The most basic form of captured meaning are *triplets* that consist of subject, predicate, and object [24]. Such an assertion is very simple but powerful. For specific languages, it is possible to derive them more easily than from the others (e.g., English with its stable sentence structure vs. Slavic languages); text mining may also be used for derivation [22]. The assertions can naturally have relations between themselves and form a swarm of assertions, which is helpful for comparing different sources of information. The information storing based on triplets is typical especially for bioinformatics.

2) Concerns: Writing a document happens with concern in mind, and typically there are multiple concerns across a document. We can understand a concern in a document as a principle that binds sentences in a paragraph, paragraphs in a section, and sections in a document together. The whole document then speaks about the highest-level concern that is then split into parts recursively, until we reach some atomic level such as paragraphs containing a set of statements. Lower-level concerns can act as a separator of document modules, and higher level concerns are then composed by multiple submodules. It indicates that splitting the concerns further is not intended by the author.

For example, considering manual for a product, the top-level concern is about the product in general with sub-concerns installation, usage, license, and warranty. The usage can be then again split into concerns related to usage of specific parts of the product. On the other hand, the warranty might not have any further sub-concerns.

3) Variants: Apart from the primary concerns in a document, there are also cross-cutting concerns that are not related to meaning and information inside a document, but rather to its usage. Such cross-cutting concerns are an intended audience, a language, a form of document (slides, handout, book, etc.), and so on. Those represent variants of a single document. They are a source of possible combinatorial effects.

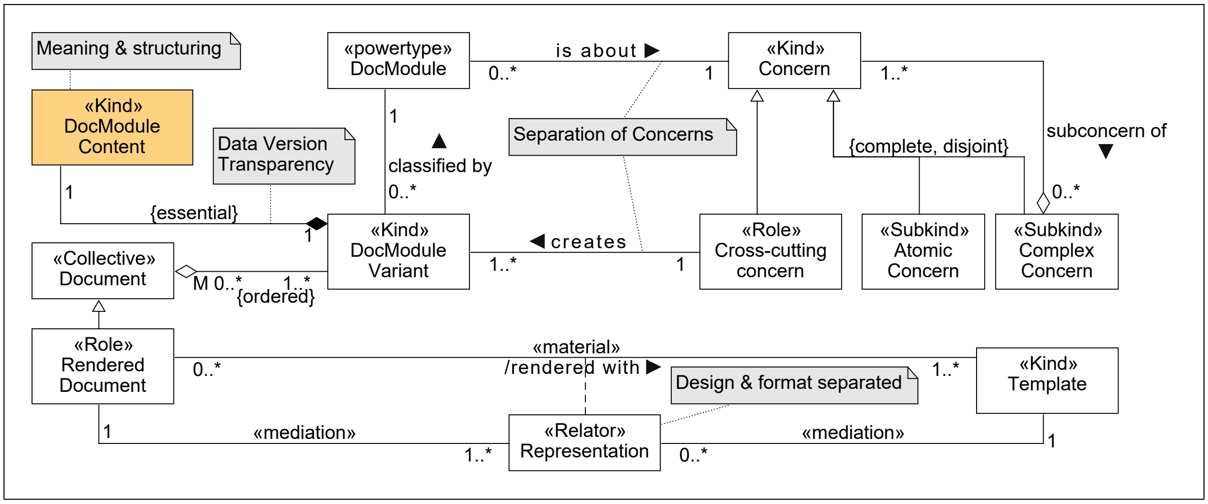

Figure 1. Conceptualization of concern-based document modularization in the OntoUML ontologically well-founded language.

For example, teaching a course requires a textbook and lecture slides which are, of course, very closely related. When you do some update in the textbook, you need to update the affected slides. Now, imagine teaching the course in two languages and some classes for seniors and some for juniors. So, you have 8 different variants and adding one more language would lead to another 4. Apparently, it is becoming hard to manage these separate documents correctly. This is the core challenge, where combinatorial effect-free documents should help.

4) Structure: A structure of a document is essentially a hierarchy of the document composition: chapters, sections, subsections in various levels, paragraphs, and parts of paragraphs. Then there are also other *block elements*, such as lists, tables, figures, code examples, equations and similar. Next, we distinguish so-called *inline elements*, which are parts of text inside a block to capture the different meaning of words (e.g., a link, important, math, a quote, a superscript, etc.) or to provide additional information, for example, a reference or a footnote. Notice that we don't state anything about the style here.

The flexibility of a document structure is an enabler of evolvability. Aligned with the notion of modules in programming, every modular unit should be loosely coupled with remaining parts and allow to be moved to a different place even in a different document. A heading level represents a typical problem: there is a level of the unit involved, and it gets more complicated with cross-references. It goes even deeper when we consider that its position in a document may form a list of prerequisites that the reader should know beforehand.

Finally, we would expect a possibility to define a new custom element, based on those already specified in the structure, to increase usability and flexibility. That indicates the need for multilevel modeling in the document structure. For example, a table with predefined rows and columns can be used for invoices, or a special type of paragraph can indicate the higher importance of content for readers.

B. Conceptualization of Documents

Based on the previous considerations, we can now assemble the conceptual models. We use already mentioned language OntoUML which uses high-level and well-defined terms from the Unified Foundational Ontology (UFO) as stereotypes and significantly enhances semantics and expressiveness of basic Unified Modeling Language (UML). Details about the language and the ontology are fully explained in [5]. The connector of all the introduced models is the *document content*, the carrier of information. All models are connected, compatible, and describe different viewpoints introduced in the previous section. Moreover, NS patterns and modularization are well observable in the following models.

1) Concern-based Document Modularization: Figure 1 shows the diagram of the conceptual model with the separation of concerns pattern for documents. A document is a modular structure composed of module variants. Concerns as the drivers of modularization are naturally binding elements of documents to groups. Cross-cutting concerns are then the special case of general concerns in case they produce variants of document modules. Documents can be rendered using many templates, while the content is still the same. That separates a used style and typography from the actual content.

For example, in a manual for a software product, there are the following concerns: installation, usage, warranties, etc. Some of those have sub-concerns, which creates submodules, e.g., installation for various platforms. A cross-cutting concern, in this case, can be the language. Variants of "installation" are formed by using different natural languages. The manual is an ordered collection of various variants. Thus it is possible to have a multi-language manual, but also language specific manual, or just installation manual in English and then reuse these module variants easily. Finally, the manual can be then rendered with a template for printing, website, annotated XML, eBook, and so on.

Language is a typical cross-cutting concern in documents, but it can also be a case of general concern for creating modules. Consider a document about some ancient language. Probably some top-level module will be about the language concern with sub-concerns related to different parts of the language. Such document can be published in many languages as a cross-cutting concern as well.

 ISBN: 978-1-61208-612-5

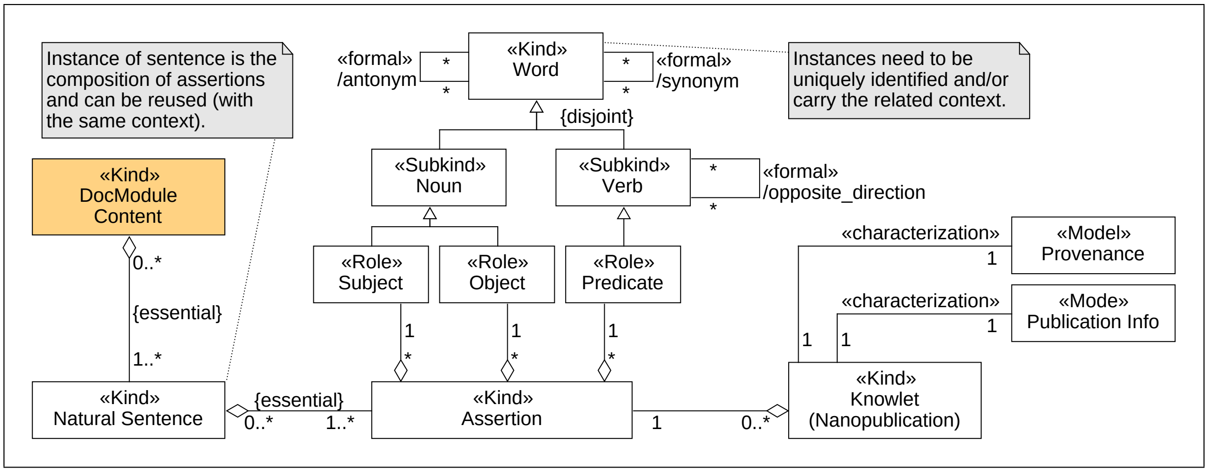

Figure 2. Conceptualization of meaning encoded in nanopublications.

2) Meaning in Nanopublications: The way a meaning is encoded within a document module content is shown in Figure 2. A content is formed by natural sentences, which are essential for the content as a whole. In a sentence, there can be one or more assertions, which are triplets in a simplified view: subject, predicate, and object. It is possible to form multiple assertions with the same meaning by using synonyms, and by switching subject with objects while using predicate for opposite direction.

Knowlet, or so-called nanopublication, is such an assertion with additional information and provenance as characterizations. Nanopublications are widely used within semantic webs and Resource Description Framework (RDF) in general, as described in [22] and [25]. Each instance of a word should be uniquely identifiable, in semantic web this problem is solved by the use of Uniform Resource Identifier (URI). For example, even with a simple assertion like *cat is white*, we need to know which cat the assertion is about, or if it is about all cats. The context is crucial for assertions, but it is hard to be adequately captured [22].

This expression of meaning could allow machines to read and understand the content in a more efficient way than is possible with text mining. Moreover, a semantic search, comparison, or reasoning can be built in a more straightforward way. It could lead to easier work with the documents, their parts and changes, and significant resource savings.

3) Document Content Structuring: The task of document content structuring has been addressed many times through syntax for composing documents and systems like the already-mentioned Pillar. For our purpose, the conceptualization is designed on a higher abstraction level, as shown in Figure 3. A content of the module is composed of block elements that contain text and inline elements that decorate part of text within a block element. Those types of elements are essentially powertypes [26] in the conceptualization, and their instances are particular usages of them. For example, the most common instance of a block element is a paragraph, and an instance of a paragraph is a particular paragraph containing a particular text, which is in our model covered by the atomic content kind not further subdivided. It works similarly for figures, pieces of data, file imports, and so on.

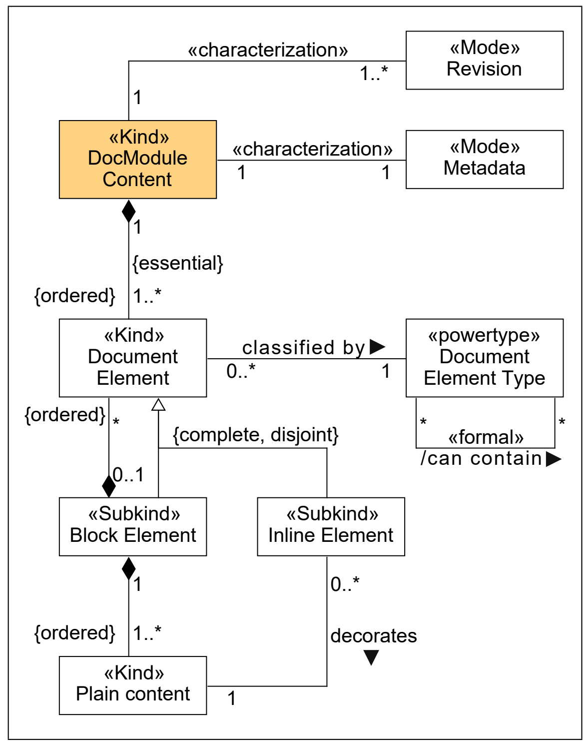

Figure 3. Conceptualization of structuring document module content.

Element type instances can be the well-known unordered or ordered lists, tables, definition lists, links, forms, cross-references, figures, quotes, external references, etc. On top of that, using powertypes allows defining new structural elements with different semantics, e.g., an important paragraph, specific table combined with a form, or external file. Metadata for each module content and document element can be provided. Content may be maintained as revisions that allow keeping track of changes.

C. Next Steps Towards Evolvable Documents

The last part of our approach is about the next steps which are suggested to be done in the near future as a consequence

 ISBN: 978-1-61208-612-5

based on the introduced conceptualization. Of course, the domain of documents is changing rapidly as well as the computer science which affects it significantly. Therefore is not just single possible way how to achieve evolvable documents and many options need to be explored, evaluated and then found out if there is some more suitable way. Mentioned steps seem to us very promising based on our own experience.

1) A Prototype of Evolvable Documents System: One of the possible next steps is to design and develop a simple prototype which further elaborates ideas from this paper and implements them. The result should be easy to use in any domain. The prototype would serve to find proof(s) of concept and to uncover new challenges.

The process of prototype development would be simplified by using the provided conceptualization and could explore missing, incorrect or unnecessary concepts by induction. Very important is to develop a system which is evolvable itself regarding Normalized Systems theory and it is not discouraging users with a complicated user interface.

2) A Methodology for Evolvable Writing: The prototype itself is not something that can be used directly and widely. However, during the process of further elaboration, some form of generic guidelines for creating evolvable documents can emerge. The possibility to write evolvable documents is highly affected by selected tools and formats.

Instead of developing a new solution, there is an alternative to build a more generic methodology and try to implement it as an integration of solutions currently available. Many of such current tools have been already mentioned: Git, Pandoc, LaTeX, Markdown, XML, GitHub, Pillar, etc.

IV. Conclusion

In this paper, we presented our initial approach to evolvable documents based on the principles of Normalized Systems theory but, compared to the related work, applicable in generic. The presented conceptualization is the basis of this genericity. By incorporating modularization based on the semiotic ladder and the NS concepts together with the ontology-driven conceptual modeling language OntoUML, we uncovered different aspects and challenges in the documents domain. The presented tightly-related conceptual models demonstrate the power of modularization and they can become a foundation for further discussion and building of a methodology or a system prototype using the model-driven development (MDD) methods. During the future research, it is likely that the models will need to be extended both in scope and detail.

Acknowledgments

This research was supported by the CTU grant No. SGS17/211/OHK3/3T/18, and was partially done as part of Summer Camp '17 organized by the FIT CTU in Prague, and also of Normalized Systems Summer School '17 organized by the University of Antwerp. This work also contributes to the CTU's ELIXIR CZ Service provision plan.

References

[1] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Kermt (Belgium): Koppa, 2016.

[2] G. Oorts, H. Mannaert, P. De Bruyn, and I. Franquet, "On the evolvable and traceable design of (under) graduate education programs," in Enterprise Engineering Working Conference. Springer, 2016, pp. 86–100.

[3] G. Oorts, H. Mannaert, and P. De Bruyn, "Exploring design aspects of modular and evolvable document management," in Enterprise Engineering Working Conference. Springer, 2017, pp. 126–140.

[4] G. Oorts, H. Mannaert, and I. Franquet, "Toward evolvable document management for study programs based on modular aggregation patterns," in PATTERNS 2017: the Ninth International Conferences on Pervasive Patterns and Applications, February 19-23, 2017, Athens, Greece/Mannaert, Herwig [edit.]; et al., 2017, pp. 34–39.

[5] G. Guizzardi, Ontological foundations for structural conceptual models. CTIT, Centre for Telematics and Information Technology, 2005.

[6] B. Duyshart, The Digital Document. Taylor & Francis, 2013.

[7] T. Arloing, Y. Dubois, S. Ducasse, and D. Cassou, "Pillar: A versatile and extensible lightweight markup language," in Proceedings of the 11th edition of the International Workshop on Smalltalk Technologies. ACM, 2016, p. 25.

[8] M. Dominici, "An overview of pandoc," TUGboat, vol. 35, no. 1, 2014, pp. 44–50.

[9] S. Kottwitz, LaTeX Cookbook. Packt Publishing, 2015.

[10] Wikipedia, Template Engines: JavaServer Pages, WebMacro, ASP. NET, Template Engine, Web Template System, Web Template Hook Styles, Haml, Template Processor. General Books, 2011.

[11] K. Ram, "Git can facilitate greater reproducibility and increased transparency in science," Source code for biology and medicine, vol. 8, no. 1, 2013, p. 7.

[12] S. Chacon and B. Straub, Pro Git, ser. The expert's voice. Apress, 2014.

[13] E. Westby, Git for Teams: A User-Centered Approach to Creating Efficient Workflows in Git. O'Reilly Media, 2015.

[14] K. Roebuck, Document Management System (DMS): High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. Lightning Source, 2011.

[15] V. Pal, Alfresco for Administrators. Packt Publishing Ltd, 2016.

[16] B. Leuf and W. Cunningham, The Wiki Way: Quick Collaboration on the Web. Addison-Wesley, 2001.

[17] A. Porter, WIKI: Grow Your Own for Fun and Profit. XML Press, 2013.

[18] C. Bunch, Automated Generation of Documentation from Source Code. University of Leeds, School of Computer Studies, 2003.

[19] R. K. Stamper, "Applied semiotics," in Proceedings of the Joint ICL/University of Newcastle Seminar on the Teaching of Computer Science, Part IX: Information, B. Randell, Ed., 9 1993, pp. 37–56.

[20] B. Frohmann, "Revisiting "what is a document?"," Journal of Documentation, vol. 65, no. 2, 2009, pp. 291–303.

[21] C. K. Ogden and I. A. Richards, "The meaning of meaning: A study of the influence of thought and of the science of symbolism," 1923.

[22] B. Mons, H. van Haagen, C. Chichester, J. T. den Dunnen, G. van Ommen, R. Hooft et al., "The value of data," Nature genetics, vol. 43, no. 4, 2011, pp. 281–283.

[23] E. Duval, W. Hodgins, S. Sutton, and S. L. Weibel, "Metadata principles and practicalities," D-lib Magazine, vol. 8, no. 4, 2002. [Online]. Available: http://www.dlib.org/dlib/april02/weibel/04weibel.html

[24] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1 concepts and abstract syntax. W3C Recommendation," 2014. [Online]. Available: https://www.w3.org/TR/rdf11-concepts/

[25] T. Kuhn, P. E. Barbano, M. L. Nagy, and M. Krauthammer, "Broadening the scope of nanopublications," in Extended Semantic Web Conference. Springer, 2013, pp. 487–501.

[26] G. Guizzardi, J. P. A. Almeida, N. Guarino, and V. A. de Carvalho, "Towards an ontological analysis of powertypes," in JOWO@IJCAI, 2015. [Online]. Available: http://ceur-ws.org/Vol-1517/JOWO-15_FOfAI_paper_7.pdf

On the Variability Dimensions of Normalized Systems Applications: Experiences from an Educational Case Study

Peter De Bruyn, Herwig Mannaert and Philip Huysmans

Department of Management Information Systems
Faculty of Applied Economics
University of Antwerp, Belgium
Email: {peter.debruyn,herwig.mannaert,philip.huysmans}@uantwerp.be

***Abstract*—Normalized Systems Theory (NST) aims to create software systems exhibiting a proven degree of evolvability. While its theorems have been formally proven and several applications have been used in practice, no real overview of the typical types or dimensions along which such NST compliant applications can evolve is present. Therefore, this paper presents an NST case within an educational context in which its different variability dimensions are illustrated. Based on this case, a more general overview of 4 variability dimensions for NST applications is proposed: changes regarding the application model, expanders, craftings and technological options.**

Keywords–Evolvability; Normalized Systems; Variability dimensions; Case Study

I. INTRODUCTION

The evolvability of information systems (IS) is considered as an important attribute determining the survival chances of organizations, although it has not yet received much attention within the IS research area [1]. Normalized Systems Theory (NST) was proposed as one theory to provide an ex-ante proven approach to build evolvable software by leveraging concepts from systems theory and statistical thermodynamics [2]–[4]. The main dimensions of evolvability or variability facilitated by the theory have nevertheless not yet been thoroughly documented. Additionally, while some NST cases have been documented in extant literature [5]–[9], the overall number of cases is still fairly limited. This paper attempts to tackle both mentioned gaps by discussing an NST application, which was built and used for the management of process evaluations of master dissertations at the faculty of the authors. Based on this case, we discuss the different dimensions along which variations in an NST application can arise. These dimensions are consequently also important indications with respect to the main areas in which an NST application can evolve throughout time.

The remainder of this paper is structured as follows. In Section II, we briefly present NST as the theoretical basis on which the considered software application was built. Section III provides some general context regarding the case: why and how the application was used. Next, the case is further analyzed in Section IV. We offer a discussion in Section V and our conclusion in Section VI.

II. NORMALIZED SYSTEMS THEORY

The case application we will present and analyze in the following sections, is based on NST. This theory has been previously formulated with the aim of creating software applications exhibiting a proven amount of evolvability [2]–[4]. More specifically, the goal is to eliminate the generally experienced phenomenon in which software systems become more difficult to maintain and adapt, as they become bigger and evolve throughout time [10].

NST is theoretically founded on the concept of *stability* from systems theory. Here, stability is considered as an essential property of systems. Stability means that a bounded input should result in a bounded output, even if an unlimited time period $T \rightarrow \infty$ is considered. In the context of information systems, this implies that a bounded set of changes should only result in a bounded impact to the system, even in cases where $T \rightarrow \infty$ (i.e., considering an unlimited systems evolution). Put differently, it is demanded that the impact of changes to an information system should not be dependent on the size of the system to which they are applied, but only on the size and property of the changes to be performed. Changes dependent on the size of the system are called *combinatorial effect*. It has been formally proven that any violation of any of the following *theorems* will result in combinatorial effects (thereby hampering evolvability) [2]–[4]:

- *Separation of Concerns*, stating that each concern (i.e., each change driver) needs to be separated from other concerns in its own construct;
- *Action Version Transparency*, stating that an action entity should be able to be updated without impacting the action entities it is called by;
- *Data Version Transparency*, stating that a data entity should be updateable without impacting the action entities it is called by;
- *Separation of States*, stating that all actions in a workflow should be separated by state (i.e., being called in a stateful way).

The application of the theorems in practice has shown to result in very fine-grained modular structures, which are generally considered to be difficult to achieve by manual programming. Therefore, NST proposes five *elements* (action, data, workflow, connector and trigger) that serve as design patterns [3], [4]:

- *data element*: a set of software constructs encapsulating a data construct (including a set of convenience methods, such as get- and set-methods, and providing

 ISBN: 978-1-61208-612-5

remote access and persistence), allowing data storage and usage within an NST application;

- *action element*: a set of software constructs encapsulating an action construct (providing remote access, logging and access control), allowing the execution of units of processing functionality within an NST application;
- *workflow element*: a set of software constructs allowing the execution of a sequence of action elements (on a specific data element) within an NST application;
- *connector element*: a set of software constructs enabling the interaction of an NST application with external systems and users in a stateful way;
- *trigger element*: a set of software constructs enabling the triggering of action elements within an NST application, based on error and non-error states.

Based on these elements, NST software is generated in a relatively straightforward way through the use of the *NST expansion mechanism*. First, a model of the considered universe of discussion is defined in terms of a set of data, action and workflow elements. Next, NST expanders generate parameterized copies of the general element design patterns into boiler plate source code. Several layers can be discerned in this code: a shared layer (not containing any reference to external technologies), data layer (taking care of data services), logic layer (taking care of business logic and transactions), remote or proxy layer (taking care of remote access), control layer (taking care of the routing of incoming requests to the appropriate method in the appropriate class in the proxy layer) and view layer (taking care of presenting the view to be rendered by the user interface, such as a web browser). This generated code can, if preferred, be complemented with *craftings* (custom code) to add non-standard functionality that is not provided by the expanders themselves at well specified places (anchors) within the boiler plate code. The boiler plate code together with the optional craftings are then compiled (built) so that the application can be deployed.

III. CASE INTRODUCTION

The case we present is concerned with the master thesis evaluations at the Faculty of Applied Economics of the University of Antwerp. At the university, master students writing their dissertation are not only evaluated with regard to the end result (i.e., the thesis itself) but also (for a minor part) with regard to the process they make in order to arrive at that end result (e.g., their communication and reporting skills, problem-solving attitude, etcetera during the project). This "*process evaluation*" is built around a set of specific evaluation criteria for students of this faculty, based upon the pedagogic vision the faculty. More specifically, depending on the trajectory a student is following, the thesis advisor(s) need(s) to assess a student two or three times on 4 skill dimensions (each comprising of a set of specific skills to be rated from insufficient until very good) during the completion of his or her master thesis.

In this context, the *procesEval* application, based on NST, was created around 2013. Up to that moment, the process evaluation was either performed on paper or had to be registered via a customized part of the university's online learning and course management system. While the paper based evaluation was considered as generating administrative overhead (the results had to be manually copied into the university's database systems by the administration) and providing little overview for the thesis advisors (e.g., when performing the second process evaluation they could not easily consult the first process evaluation in order to make a more objective comparison), the electronic variant in the online learning and course management system was considered cumbersome from a usability perspective (e.g., users complaining about the amount of clicks required to perform "simple" actions or experiencing difficulties in order to find the information they are looking for).

The faculty management decided to develop an NST application to manage the process evaluations. This choice was made for several reasons. First, the expertise on how to build NST applications was present within the faculty itself as the theory (and the adjoining code expanders) was the output of research projects of faculty members. Second, as the software system would be developed by members of the faculty itself as well, the developers were highly knowledgeable about the inner working of the faculty (administration) and the associated (functional) requirements. And third, evolvability and maintainability were considered to be import quality aspects of the software system to be developed as the process evaluation was anticipated to remain an important part of the student evaluations for several years to come (but could be subject to some further fine-tuning or redirection in the future). Given the situation of the project as sketched above, it was expected that the application could be developed in a rather short development trajectory without too many hurdles (i.e., no significant risk related to the technology was present and the application domain was well known and understood).

The application itself was developed in the beginning of 2013. In the academic years 2013–2014 and 2014–2015, a first pilot test with a set of key users (technological proactive faculty members) was conducted. In the academic years 2015–2016 and 2016–2017, the set of test users was gradually expanded up to the level were all thesis supervisors could use the procesEval application if they wanted, but could still use the paper version if preferred. As of the academic year 2017–2018, all faculty members were expected to use the NST procesEval application for the administration of the master thesis process evaluations. Apart from minor (usability) adjustments, the project has been completed without major problems. Currently, on a yearly basis, about 45 faculty members manage the process evaluation of roughly 500 students via the procesEval application.

In Figure 1, a screenshot of the procesEval application is shown (the names of the students are blurred out to assure anonymity, the names of the label being Dutch as this is the administrative language of the organization). Here, one can see that a supervisor can get an overview of all the students he or she is supervising in the current academic year. By selecting a particular student, a set of tabs appears below the first table providing further details regarding his (earlier) evaluations or working sessions (e.g., meetings) and documents (e.g., preliminary thesis version). Figure 2 shows a screenshot of one particular process evaluation. The application therefore manages all process evaluations (typically 2-3) of all master dissertations (as of 2017–2018) of multiple academic

 ISBN: 978-1-61208-612-5

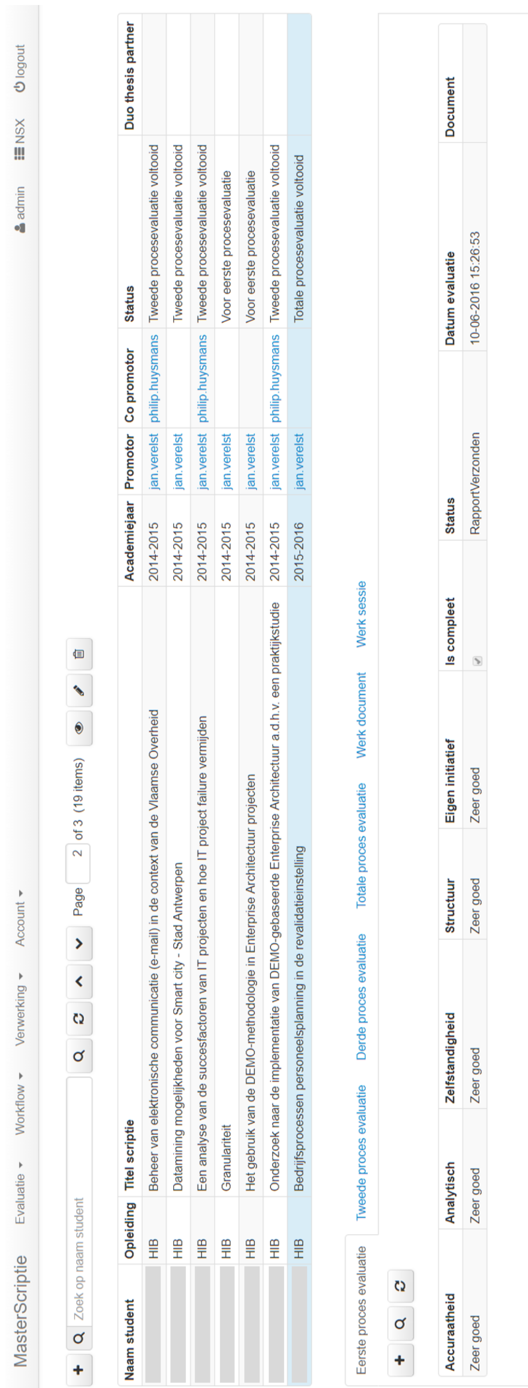

Figure 1. A general screenshot of the procesEval application.

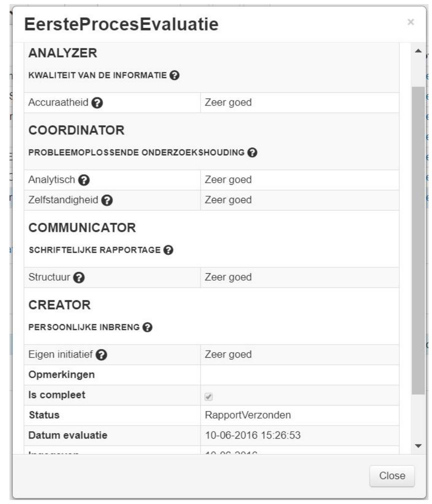

Figure 2. A screenshot of a specific process evaluation within the procesEval application.

years. Based on the provided information, the application automatically generates overview reports of the evaluations, sends an email with information regarding their evaluations to students and supervisors as well as reminders (e.g., when a particular process evaluation is due).

IV. CASE ANALYSIS

A. General overview

An NST application typically consists of a set of base components (which are reused in several applications), as well as one or multiple non-base components (typically specific for the application under consideration). The base components used within the procesEval application consisted of 29 data elements, 7 task elements and 1 flow element. The non-base component used within the procesEval application consisted of 14 data elements, 8 task elements and 4 flow elements. As a consequence, relatively speaking, the NST application was still rather small comprising about 63 NST elements.

B. Model variations

By using the NST approach, the procesEval application could be extended and adapted at the level of the model (i.e., the definition of the different element instances for the considered application domain). For instance, additional elements could be added: next to the registration of three possible process evaluations for each student, some working documents and information regarding working sessions (e.g., what was agreed upon by the student and his supervisor during a meeting) could be added to the model. After re-generating the application based on this model, this functionality becomes available in the new version of the application. Similarly, existing (i.e., earlier created) components could be added to the model. For example, a notification component was added to the procesEval as this contained the functionality to automatically trigger emails and could be leveraged to enable the automatic report delivery (of the process evaluations to the students, supervisors and administration). While the model could be changed in terms of data elements and components, this also holds for all other possible changes within the model.

More specifically, the following types of adaptations can be performed to create different variations of the application:

- addition, update or deletion of a component (i.e., a set of data, task and flow elements);
- addition, update or deletion of a data element definition (its fields with its types and options, finders, options, child elements);
- addition or deletion of a task element definition (the specific implementation of a task is a crafting, see below);
- addition update or deletion of a flow element definition and its accompanying default state transitions.

It should be remarked that the determination and evolutions of such model is completely technology-agnostic (i.e., does not require any specification in programming language specific terminology). For instance, the specification of the model (in terms of elements and their properties) is currently stored in an XML format, not containing any references to the (background) technology of the current reference implementation (i.e., Java). Based on this model, boiler plate source code for each of the layers can be created.

C. Crafting variations

Once the model is converted into boiler plate source code, additional code (so-called "craftings", which are custom made for an application) could be added between predefined anchors (insertions) or in additional classes (extensions). This way, non-standard functionality can be incorporated within the application as well. In total, the procesEval contained 22 classes with insertions and 29 additional classes. For instance, in the procesEval application, specific coding had to be added to make sure that a supervisor logging into the application can only view those master dissertations that he is supporting in the concerning year (i.e., dissertations supported by other supervisors or those of the previous year should not be visible). For this purpose, a few lines of code were added in the MasterThesisFinderBean determining the fetching of the results viewable for a particular user. These FinderBeans are expanded as part of the data layer: enforcing the filter of master dissertations at the level of the data layer ensures that no data from other users can be retrieved by the currently logged in user. Consequently, this crafting only impacts the data layer, while the remaining layers have no impact from this change: they perform their functionality handling the (filtered) data offered by the MasterThesisFinderBean.

Additionally, a set of screentips was added to assist the user in filling-in the process evaluation (e.g., summarizing the meaning of each of the evaluation criteria in case of a mouse-over). The expanded NST code base supports this functionality by providing a helpInfo Knockout binding. Specific screentips can be added by including a crafting using this Knockout binding, and referring to a certain key. At run-time, the specific values for the required keys can be added in instances of HelpInfo data elements. This enables the configuration of the screentips even when the application has already been deployed. Note that only the view layer is customized for this functionality. This makes sense, since it is purely a useability concern, not impacting actual business logic. However, it is dependent on the specific technology used in the view layer (i.e., Knockout), and should be remade when a different technology is used.

Next, as mentioned before, the procesEval also needed to create and send reports summarizing the content of the process evaluation. The definition of these reports (the items to be included and the corresponding layout) is considered to be a separate functionality, and should therefore be contained in a task element. The expanders provide all boilerplate code needed to execute this task in the NST application, and only the specific report generating functionality needs to be added as a crafting. The actual implementation of the execution of a task element is clearly separated, allowing versions and variations of the task implementation to co-exist. Currently, reports are generated using Jasper Reports. This requires the addition of a Jasper template file to the code base, and some code to fill the parameters to be inserted into this template. The additional processing logic is completely contained in the logic layer.

These craftings were added in a gradual and iterative way to the application: each time a particular additional functionality was added or improved, a new version of the overall application could be built and deployed. Each of these craftings were situated at another layer (i.e., data, view and logic).

D. Infrastructural technology variations

The procesEval application could be generated by using various different underlying infrastructural technologies. For instance, whereas a prototype of the application is typically demonstrated by using an HSQL database, most production systems are deployed while using a PostgreSQL database. Nevertheless, one can choose for SQLServer and MySQL databases as well. Further, the procesEval can be built by using different build automation frameworks (i.e., Ant and Maven). And finally, the procesEval could also be generated by using different controlling (Cocoon, Struts2, or combination Struts2-Knockout) and styling frameworks (plain style or using Bootstrap). In practice, the Struts2-Knockout and Bootstrap were used in the production environment. Changing the choice of a particular infrastructural technology in the procesEval only impacts those layers depending on the purpose of the technology (e.g., the database selection impact the data layer, whereas the GUI framework selection impacts the view layer).

E. Expander version variations

The expanders (i.e., the programming logic used to convert the model into boiler plate source code according to the infrastructural technologies chosen) evolves throughout time as well. This way, when considered the current procesEval project duration (2013–present), 8 different production versions were deployed while using the same model and craftings (as the expanders provide backwards version compatibility). In each of these production versions, the new or improved possibilities of the expanders could be used. For instance, in one particular version of the expanders, information regarding a Date field did no longer have to be entered manually but could be selected by using a more advanced date picker. And, more relevant in the context of the procesEval, another particular version of the expanders allowed the automatic creation of summarizing graphs on certain fields. For example, it would now be possible to inspect the number of master dissertations who did not yet receive a first process evaluation versus those who did in a visual way. In order to use the date picker, no changes in the model or the craftings are required. In order to use the status graphs, only one additional specification in the model (i.e., an option indicating that a graph for a particular field should be created) needs to be added. Clearly, the precise set of layers that is impacted due to an expander update depends on the type of modifications performed in that particular version update (logic related, view related, etcetera).

V. DISCUSSION

While the above analysis is only based on one case study, we anticipate that the proposed categorization can be generalized to a large extend as it also aligns with the general "degrees of freedom" available during the development and maintenance of an NST application. This overall approach, together with 4 variability dimensions, is visualized in Figure 3.

First, as represented at the top of the figure, the modeler should select the *model* he or she wants to expand. Such a model is technology agnostic (i.e., defined without without any reference to a particular technology that should be used) and represented by a blue puzzle (i.e., each puzzle piece represents a defined element, with the columns corresponding to data, task, flow, trigger and connector elements). Such a model can have multiple versions throughout time (e.g., being updated or complemented) or concurrently (e.g., choosing between a more extensive or summarized version). As a consequence, the figure contains multiple blue puzzles that are put behind each other and the chosen model represents a variability dimension (represented by the green bidirectional arrow).

Second, the *expanders* (represented by the trapezoid in the figure) generate (boiler plate) source code by taking the specifications in the chosen model as its *arguments*. For instance, for a data element Person, a set of java classes PersonBean, PersonLocal, PersonRemote, PersonDetails, etcetera will be generated. This code can be called boiler plate code as it provides a set of standard functionalities for each of the elements within the model. Nevertheless, one could argue that this set of standard functionalities is already quite decent as it contains the possibilities to provide standard finders, master-detail (waterfall) screens, certain display options, document upload/download functionality, child relations, etcetera. The expanders themselves evolve throughout time. Typically, in each new version, a set of bugs of the previous one are solved and additional features (e.g., creation of a status graph) are provided. It should be remarked that, given the fact that the application model is completely technology agnostic and can be used as argument for any version of the expanders, these bug fixes and additional features become available for all versions of all application models (only a re-expansion or "rejuvenation" is required). As a consequence, the figure contains multiple trapezoids that are put behind each other and the expander version represents a variability dimension (represented by the green bidirectional arrow).

Third, in the middle left of the figure, a set of *infrastructural options* are displayed by means of different rectangular blocks. These consist of global options (e.g., determining the build automation framework), presentation settings (determining the graphical user framework), business logic settings (determining the database used) and technical infrastructure (e.g., determining the background technology). For each of these infrastructural options, the modeler can choose out of a set of possibilities (e.g., different user interface frameworks for which the associated code can be generated), which will be used by the expanders as their *parameters*. That is, given a chosen application model version and expander version, different variants of boiler plate code can be generated, depending on the choices regarding the infrastructural options. As a consequence, the figure contains multiple infrastructural option block sets that are put behind each other and the infrastructural options represent a variability dimension (represented by the green bidirectional arrow).

Fourth, *craftings* ("custom code") can be applied to the generated source code. These craftings are represented in the lower left of the figure by means of red clouds as they enrich (are put upon) the earlier generated boiler plate code and can be harvested into a separate repository before regenerating the software application (after which they can again be applied). This includes extensions (e.g., additional classes added to the generated code base) as well as insertions (i.e., additional lines of code added between the foreseen anchors within the code). Craftings can have multiple versions throughout time (e.g., being updated or complemented) or concurrently (e.g., choosing between a more advanced or simplified version). These craftings should contain as little technological specific statements within their source code (apart from the chosen background technology). Indeed, craftings referring to (for instance) a specific GUI framework will only be reusable as long as this particular GUI framework is selected during the generation of the application. In contrast, craftings performing certain validations but not containing any EJB specific statements will be able to be reused when applying other versions or choices regarding such framework. Craftings not dependent on the technology framework of a specific layer can be included in the "common" directory structure, whereas technology-dependent craftings need to reside in the directory structure specified for that technology (e.g., EJB for the logic layer, JPA for the data layer, Struts2 for the control layer). As a consequence, the figure contains multiple crafting planes that are put behind each other and the chosen set of craftings represents a variability dimension (represented by the green bidirectional arrow).

In summary, each part in Figure 3 with green bidirectional arrows is a variability dimension in an NST context. It is clear that talking about *the* "version" of an NST application (as is traditionally for software systems) in such context becomes rather pointless. Indeed, the eventual software application (the grey puzzle at the bottom of the figure) is the result of a specific version of an application model, expander version, infrastructural options and set of craftings. Put differently, with M, E, I and C referring to the number of available application model versions, the number of expander versions, the number of infrastructural option combinations and crafting sets respectively, the total set of possible versions V of a particular NST application becomes equal to:

$$V = M \times E \times I \times C$$

Remark that the number of infrastructural option combinations is equally a product:

$$I = G \times P \times B \times T$$

Where G represents the number of available global option settings, P the number of presentation settings, B the number of business logic settings and T the number of technical infrastructure settings. This general idea in terms of combinatorics corresponds to the overall goal of NST: enabling evolvability

 ISBN: 978-1-61208-612-5

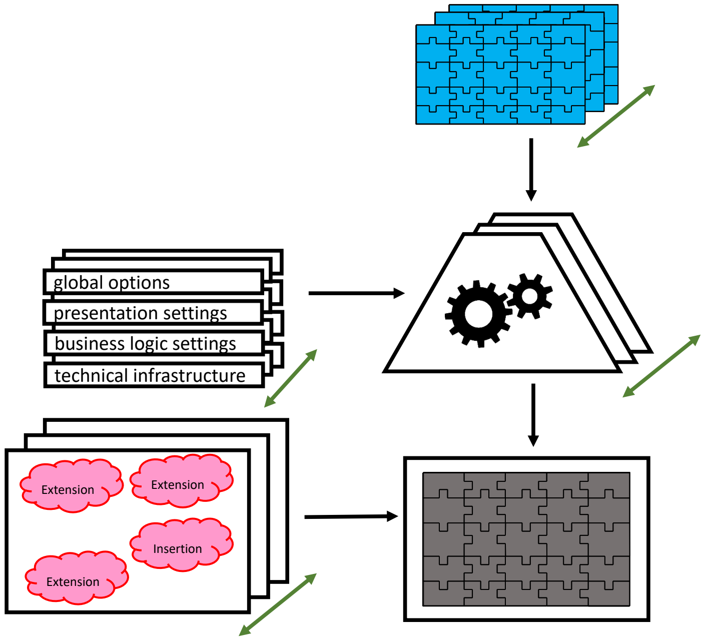

Figure 3. A graphical representation of four variability dimensions within a Normalized Systems application.

and variability by *leveraging the law of exponential variation gains* by means of the thorough decoupling of concerns and the facilitation of their recombination potential [4].

VI. CONCLUSION

This paper presented a case study of an NST software application in an educational context and analyzed the different dimensions in which it could evolve. Based on this, 4 general variability dimensions were proposed.

This paper is believed to make several contributions. From a theoretical side, inductive reasoning based on our case allowed the formulation and illustration of 4 variability dimensions, might be the (or at least a subset of the) orthogonal dimensions along which a typical NST application can evolve. At the same time, these variability dimensions clarifies that the concept of an overall application "version" is not applicable for NST applications as a specifically deployed application is the result of a combination of choices for each of the variability dimensions. For practitioners, this paper contributes to the set of case studies available on NST, which might provide them with a better insight regarding the application potential of the theory in practice.

Next to these contributions, it is clear that this paper is also subject to a set of limitations. That is, we proposed the set of variability dimensions based on one case study, which was limited in size and complexity. This limits the generalizability of our findings. Therefore, future research should be directed towards the analysis of additional cases, including information systems being larger, more complex and executed within other application areas than the educational industry. These additional cases might confirm, and possibly extend, the variability dimensions proposed in this paper.

REFERENCES

[1] R. Agarwal and A. Tiwana, "Editorialevolvable systems: Through the looking glass of is," Information Systems Research, vol. 26, no. 3, 2015, pp. 473–479.

[2] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, special Issue on Software Evolution, Adaptability and Variability.

[3] ——, "Towards evolvable software architectures based on systems theoretic stability," Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116.

[4] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[5] M. Op't Land, M. Krouwel, E. Van Dipten, and J. Verelst, "Exploring normalized systems potential for dutch mods agility: A proof of concept on flexibility, time-to-market, productivity and quality," in Proceedings of the 3rd Practice-driven Research on Enterprise Transformation (PRET) working conference, Luxemburg, Luxemburg, September 2011, pp. 110–121.

[6] G. Oorts, P. Huysmans, P. De Bruyn, H. Mannaert, J. Verelst, and A. Oost, "Building evolvable software using normalized systems theory : a case study," in Proceedings of the 47th annual Hawaii international conference on system sciences (HICSS), Waikoloa, Hawaii, USA, 2014, pp. 4760–4769.

[7] P. Huysmans, P. De Bruyn, G. Oorts, J. Verelst, D. van der Linden, and H. Mannaert, "Analyzing the evolvability of modular structures : a longitudinal normalized systems case study," in Proceedings of the Tenth International Conference on Software Engineering Advances (ICSEA), Barcelona, Spain, November 2015, pp. 319–325.

[8] P. Huysmans, J. Verelst, H. Mannaert, and A. Oost, "Integrating information systems using normalized systems theory : four case studies," in Proceedings of the 17th IEEE Conference on Business Informatics (CBI), Lisbon, Portugal, July 2015, pp. 173–180.

[9] P. De Bruyn, P. Huysmans, and J. Verelst, "Tailoring an analysis approach for developing evolvable software systems : experiences from three case studies," in Proceedings of the 18th IEEE Conference on Business Informatics (CBI), Paris, France, August-September 2016, pp. 208–217.

[10] M. Lehman, "Programs, life cycles, and laws of software evolution," in Proceedings of the IEEE, vol. 68, 1980, pp. 1060–1076.

 ISBN: 978-1-61208-612-5

On the Evolvability of Code Generation Patterns: The Case of the Normalized Systems Workflow Element

Herwig Mannaert and Peter De Bruyn

Department of Management Information Systems
Faculty of Applied Economics
University of Antwerp, Belgium
Email: {herwig.mannaert,peter.debruyn}@uantwerp.be

Koen De Cock and Tim Van Waes

Research and Development Department
Normalized Systems eXpanders
Antwerp Science Park, Belgium
Email: {koen.de.cock,tim.van.waes}@nsx.normalizedsystems.org

***Abstract*—Normalized Systems Theory (NST) aims to create software systems exhibiting a proven degree of evolvability. The theory contains a set of formally proven theorems and proposes a set of elements (patterns) to realize the adherence to the theorems in practice. While the development and evolution of several information systems in practice (based on the theory) have been documented, the evolution or enhancement regarding one of the fundamental element patterns has not yet been presented. Therefore, this paper discusses the evolution of one of these patterns, the flow element, and what this means for the software applications it is used in. This way, additional insight is provided on how the theory enables the enhancement and simultaneous evolution of large sets of information systems.**

***Keywords*–Evolvability; Normalized Systems; Design Patterns**

I. INTRODUCTION

Having evolvable information systems (IS) is important for the survival chances of organizations, although the topic has not yet received much attention within the IS research area [1]. Normalized Systems Theory (NST) precisely focuses on providing an ex-ante proven approach to build evolvable software systems by using concepts from systems theory and statistical thermodynamics [2]–[4]. In order to apply the theory in practice, a code generation framework (NS expanders) was developed, compliant with NST. During the last five years, about 50 information systems have been developed using this code generation framework, from which some cases have been documented in previous publications [5]–[9].

While these cases describe the evolution of various versions of specific information systems developed using NST and the NS expanders, no specific evolution within the code generation framework itself has currently been documented. Therefore, this paper will discuss the evolution (the incorporation of additional and improved functionality) of a specific part of the NS expanders, i.e., the flow element. This way, we aim to provide additional insight into how the theory (and more specifically, the systematic improvement of the NS expanders) enables the enhancement and simultaneous evolution of a large set of information systems.

The remainder of this paper is structured as follows. In Section II, we discuss some related work with an emphasis on NST, which is the theoretical basis of both the evolutionary approach and the code generation framework. Section III explains the structure of an initial version of the flow element. Next, we discuss some additional developments and improvements regarding the considered element in Section IV and some empirical test results in the context of this element evolution in Section V. Finally, our discussion is offered in Section VI.

II. RELATED WORK

In Sections II-A and II-B, we introduce and summarize the theoretical underpinnings of NST and the associated expansion and regeneration mechanisms, respectively. In Section II-C, we briefly discuss the contrast with some existing approaches.

A. Normalized Systems Theory

The software applications and the patterns they make use of, as we will discuss and analyze in the following section, are based on NST. NST was proposed with the purpose of allowing the design of software applications exhibiting ex-ante evolvability [2]–[4]. In particular, the theory focuses on the ripple effects (due to all kinds of coupling) occurring in software systems when changes are applied and proposes some ways to eliminate them. It is believed that such kind of ripple effects may be one of the main causes of Lehman's Law of increasing complexity [10], which states that software systems become more difficult to maintain and adapt over time due to its deteriorating structure. Indeed, the more coupling and the more ripple effects, the more difficult software applications can be adapted.

NST starts from the concept of stability as defined in systems theory as its theoretical basis. A system is considered to be stable when a bounded input only results in a bounded output, even for those cases where an unlimited time period is considered. In the context of software applications, this would require that a bounded set of elementary functional changes should only result in a bounded impact to the software system, even when considering an unlimited time period (and therefore, an unlimited system size with an unlimited amount of software construct instances). This reasoning, based on stability, therefore also implies that the impact of changes to a software system (i.e., the number of construct instances that need to be created or adapted) cannot be dependent on the size of that information system. The impact should only be dependent on the size and property of the changes that are applied and not the number of construct instances within the system. Changes of which the impact is dependent on the size of the software systems are called *combinatorial effect*,

 ISBN: 978-1-61208-612-5

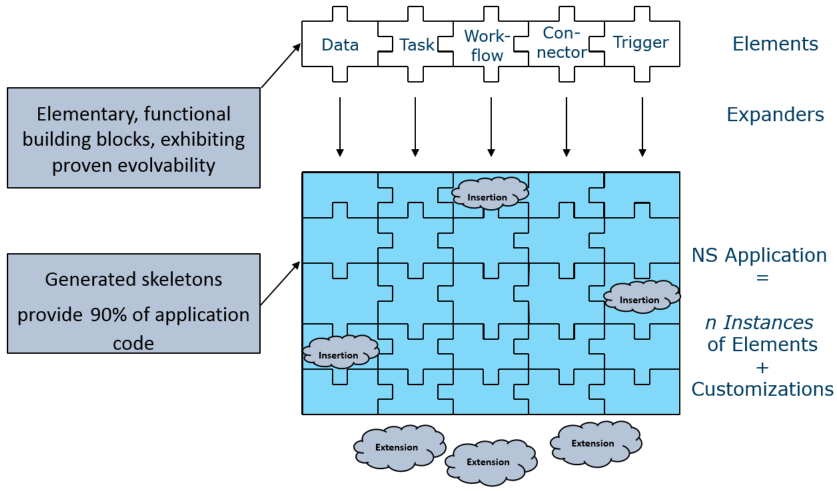

Figure 1. A graphical representation of code generation and additional custom coding within a Normalized Systems application.

and are to be avoided. NST has proposed and proved that a set of theorems should be complied with in order to avoid combinatorial effects (as their presence negatively impacts the evolvability of a software system) [2]–[4]:

- *Separation of Concerns*, which states that each concern (i.e., each change driver) needs to be encapsulated in an element, separated from other concerns;
- *Action Version Transparency*, which declares that an action entity should be updateable without impacting the action entities it is called by;
- *Data Version Transparency*, which indicates that a data entity should be updateable without impacting the action entities it is called by;
- *Separation of States*, which states that all actions in a workflow should be separated by state (and called in a stateful way).

B. NS Expansion and Regeneration

Applying the NST theorems in a systematic way results in a software system having many (but small) modules. The design of such fine-grained structure is far from trivial. Moreover, as any theorem violation during the development process results in combinatorial effects, it is very hard to create a perfectly NST compliant application by manual coding. Consequently, NST advocates the use of elements: design patterns which are re-used (and parametrized) over and over again to build a software system. Therefore, a software system is said to be generated or "expanded" to a large extent. In particular, the following set of five elements is currently proposed [3], [4]:

- *data element*: used to enter, update and retrieve data, such as *invoices* or *customers*.
- *task element*: used to perform specific tasks, such as creating and rendering an invoice.
- *flow element*: used to sequence and execute various tasks on the instances of the data.
- *connector element*: used to enable input/output by human users or external systems.
- *trigger element*: used to activate flows and tasks in a periodic way.

A code generation framework, called the NS expanders and built by a spin-off company of the University of Antwerp (i.e., the Normalized Systems eXpanders factory or NSX), has been created in order to further refine the elements. This framework allows the creation of NST software in a relatively straightforward way and has been used to generate several applications in different types of industries. As schematically represented in Figure 1, the generated code base or skeleton is later on augmented with additional —manually written— custom code. This custom code is divided into *insertions* (code fragments embedded within the classes of the elements) and *extensions* (separate software classes).

Insertions and extensions are harvested and stored in a separate source code repository, enabling the possibility to reinject this custom code base into future generations of skeletons consisting of expanded elements. Such future versions could provide additional or improved functionality in the element patterns (e.g., providing more advanced security features, or allow the use of new (versions of) frameworks for specific concerns like persistency or access control). This process of regenerating the code skeleton using a new version of the NS expanders, and reinjecting the existing custom code into a new version of the information system, is called *regeneration* or *rejuvenation*.

C. Related approaches

Other approaches than NST have obviously also already advocated the use of (software) design patters. Typically, the idea is to document and provide access to high quality solutions for frequently occurring problems so that the same problem does not need to be solved by every single developer and he or she can immediately make use of a mature solution that has proven its value in the past. A seminal work in this respect was for instance the work of the Gang of Four [11]. While these design patterns represent the core reasoning of the solution to a particular problem, most of them (such as those from Gamma et al. [11]) still require a certain amount of interpretation before they can be converted into actual working code (i.e., they cannot be mapped one-to-one to software code). The NS expanders, however, do not need this additional interpretation and result directly in operational software code.

 ISBN: 978-1-61208-612-5

Recently, some work on Model-Driven Software Development (MDSD) [12] has adopted a similar approach in which models are created and can then be converted into working code. Our approach differs in the sense that we specifically focus on generating software with a high degree of evolvability.

III. Initial Flow Element Pattern

This section focuses on the initial design of the *flow element* pattern and the additional requirements which arose during its use in practice. In the next section, we will discuss how these additional requirements were incorporated in an improved version of the element.

A. The Flow Element Pattern

Figure 2 presents a sequence diagram documenting an initial version of the inner pattern of the flow element (i.e., before the target development as discussed in Section IV was applied). For every flow element called `<Flow>`, a number of Java classes are generated or *expanded*, to implement the automated processing of a state machine operating on a data element called `<Data>`. The various operations or tasks of the state machine are specified in the various entries of a configuration data element called `StateTask`. The individual operations or tasks are implemented in task elements called `<Task>`, and every execution of a task on an instance of the data element `<Data>` is logged in an entry of an history data element called `<Data>TaskStatus`. In order to be able to start/stop the processing of the flow element and set some other parameters (such as time windows or time intervals), a control data element `EngineService` is provided.

Let us focus on the `orchestrate` method of the central class `<Flow>EngineBean` of the flow element `<Flow>` as shown in Figure 2. First, based on the name of the flow, the control data is retrieved from the appropriate entry of the *EngineService* data element. If the flow engine is not stopped, the various entries of the *StateTask* data element (configuring the state machine) are retrieved for this workflow. Then, the engine loops through the various state tasks or transitions. For every state transition specified for this workflow element, all instances of the target data element are retrieved whose status corresponds to the begin state specified in the state transition. In a second (embedded) iteration, the flow engine loops through every instance of the data element and invokes the task element (`<Task>`) as specified in the state transition entry. In accordance with the internal structure of the task element, this corresponds to an invocation of the `perform` method of the `<Task>Bean` who will delegate the actual implementation to a delegation class. For every execution of a task on an instance of the data element, an entry is created in a `<Data>TaskStatus` data element for the purpose of logging and history tracking. As can be observed in the sequence diagram, the entry is created before the task is executed, and updated after execution (e.g., adding the timestamp at completion, and specifying whether the result was a success or failure).

B. Additional Requirements

Certain requirements for highly demanding back-end processes, related to data integrity and high performance, were not provided out-of-the-box by the above described initial flow element. However, such requirements are necessary when, for example, processing millions of income or VAT tax declarations. Consider for instance the need for locking or claiming of instances of the target data element, the need for the transactional encapsulation of the task execution and the setting of the result state, and the need for the parallel and simultaneous execution of a task on multiple instances of the target data element. An improved version of the flow element, incorporating these functionalities, should however still allow the regeneration of all existing (i.e., previously developed) applications, and therefore the regeneration of all existing instances of flow elements across all these applications. Therefore, backward compatibility is considered crucial: the default behavior of the improved flow element needs to correspond to the behavior of the original flow elements that are being regenerated or rejuvenated.

IV. Improved Flow Element Pattern

The flow element pattern was extended and improved in order to accommodate the additional functional requirements for high throughput processing as discussed Section III-B. In this section, we will discuss how the structure of the flow element pattern evolved, gradually introducing the additional functionality that satisfies the various functional requirements related to this high throughput processing.

A. Serial Instance Processing

After retrieving and applying the control data for the workflow engine, and following the retrieval of the various state transition entries for the workflow engine, the first additional piece of functionality is represented in the sequence diagram of Figure 3. In some cases, it is desired to process all tasks (or a number of them) in a serial or consecutive way on a single instance of the target data element. Consider for instance the consecutive processing of various tasks on a single invoice (e.g., computing the invoice, entering the in an accounting system, rendering the invoice document, and mail the invoice), instead of performing every first, second, etc. individual task on all invoices before processing the next task. For this purpose, a *StateTaskChainBuilder* is used to group the various state task transitions in consecutive chains. A parameter specifying the chain building strategy (which might for instance specify a maximal length for the chains) allows us to perfectly emulate the old behavior by indicating a maximal chain length equal to 1. The iteration to retrieve all instances of the target data element based on a specified begin state now loops over the various begin states of the state task chains, instead of using all begin states of all individual state tasks.

B. Claiming for Data Integrity

The second additional piece of functionality is the need for a data claiming mechanism. Before the actual processing of tasks on instances of the target data element, these instances are claimed using a claiming table in the database. Though it was already impossible for two different engines running in parallel to process the same instance of a target element simultaneously (through the use of intermediate states), processing time could be lost by using several parallel engines. Indeed, various parallel engines retrieving the same data instances frequently experienced during processing that data instances were already being processed by another engine in the initial version of the flow pattern. The mechanism of claiming data instances before processing them, avoids this.

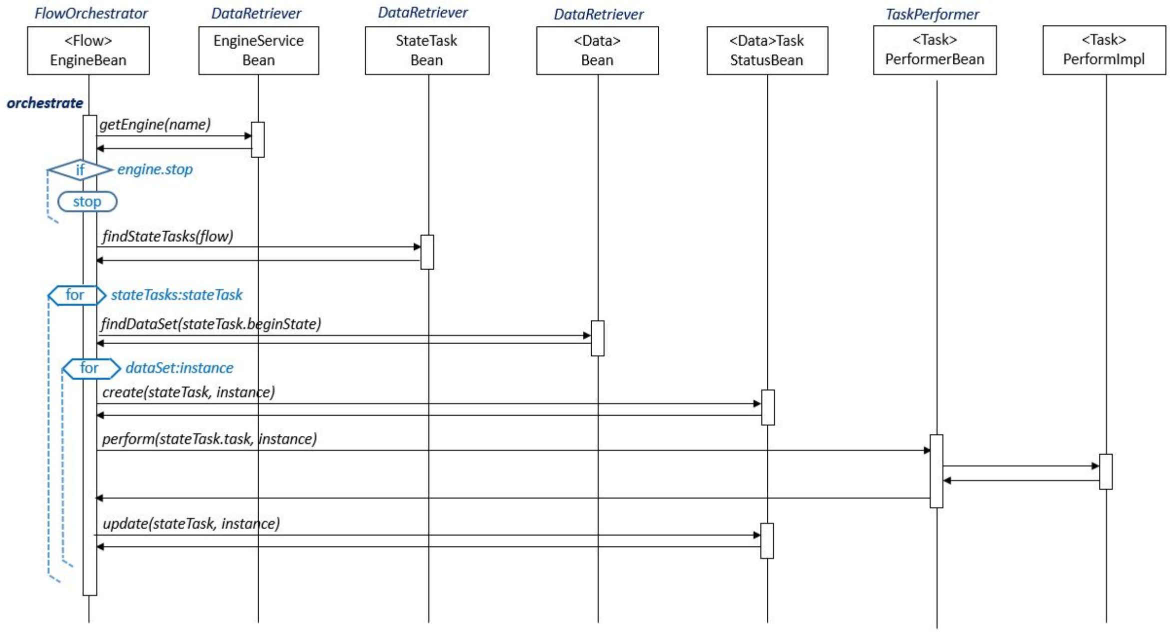

Figure 2. A sequence diagram representing the control flow of the initial flow element pattern.

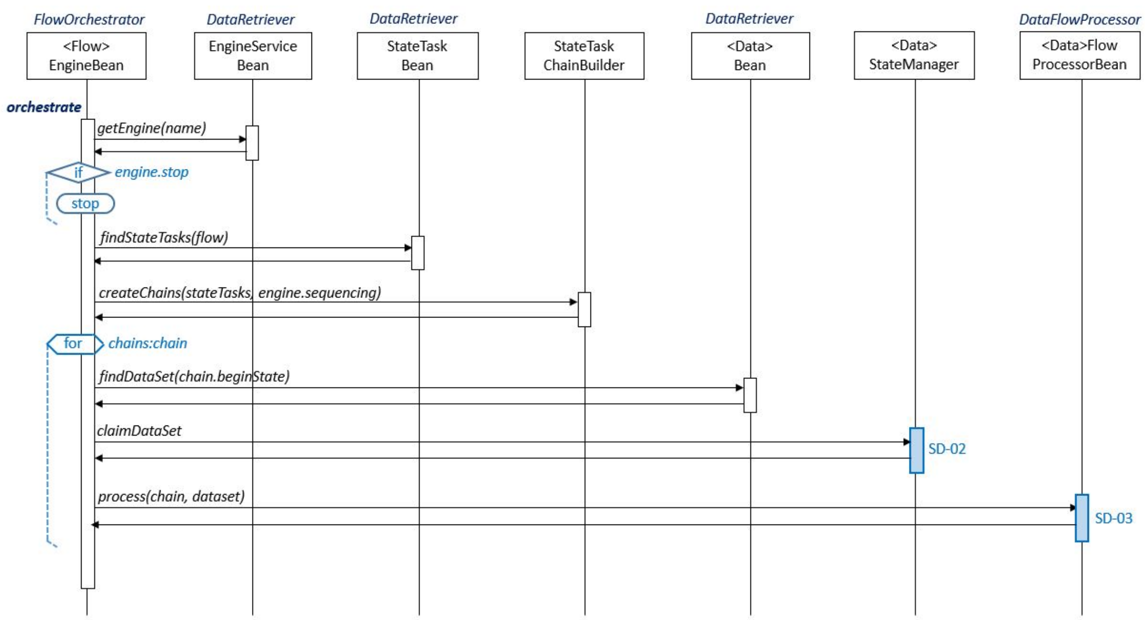

Figure 3. A sequence diagram containing the creation of state task chains and the claiming of instances in the improved flow element pattern.

C. Parallelized Task Processing

Now that a specific set of data instances has been claimed, they can be processed. Instead of looping straight away through all instances, the whole data set is passed to a `<Flow>Processor`, which is able to perform a single task —or a chain of consecutive tasks— in parallel threads simultaneously on multiple instances of the target data element. Note that backward compatibility can be achieved by using a parameter to specify the maximum amount of concurrent threads. This enables the emulation of the old behavior, by simply setting the value of the parameter to 1. Or alternatively, by using another `<Flow>Processor` implementation, which might even not allow parallel processing.

The control flow for this parallel processing mechanism is represented schematically in Figure 4. The parallel processor retrieves first the above mentioned parameter for the task(s) that are being processed, and starts the iteration through the various data element instances. For every instance, it is checked whether this instance has indeed been claimed by this engine, and a local hash map is retrieved to verify the maximum amount of instances that are currently being processed. If this maximum amount is reached, the flow engine will wait until a

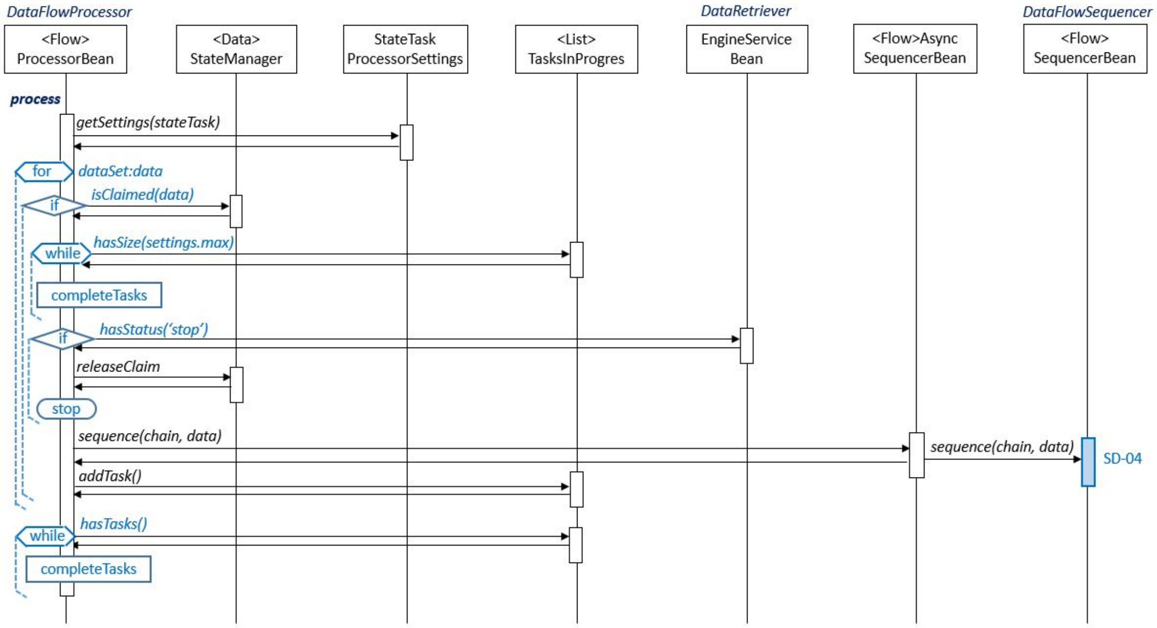

Figure 4. A sequence diagram containing the simultaneous and parallel processing of instances in the improved flow element pattern.

processing slot becomes available. If a slot becomes available, and the flow engine has not been ordered to stop in the control table in the meantime, a `<Flow>AsyncSequencer` is invoked to create a new processing thread. Within this newly created thread, which is being added to the hash map of running processing task threads, a `<Flow>Sequencer` is invoked to perform the (chain of) task(s) on the data instance. When processing threads have been launched for all instances of the data set, the `<Flow>Processor` simply waits until all processing threads have been completed.

D. Transactional State Transitions

Within the `sequence` method of the `<Flow>Sequencer` class, as represented in Figure 5, the data element instance is processed by the different consecutive tasks of the state task chain. For every task of the state task chain, three transactional steps ensure the transactional integrity of the statuses. First, the status of the data instance is set to an interim state, and an entry in the `<Data>TaskStatus` data element is created. Second, the target data instance is processed through an invocation of the actual task element, the status of the data instance is set to an end state, and the entry in the `<Data>TaskStatus` data element is updated (all within a single transaction). Finally, in case of failure, the setting of the end state and the updating of the entry in the history table, is combined with the creation of an additional entry in a failure history data element.

Also here it is crucial that the previously existing behavior of the flow element can be emulated by default. This is done by adding a transaction attribute to the specification of the data element. If not specified, a default value of *no_transaction* is selected, which omits the whole transactional behavior.

V. Empirical Test Results

The initial flow element has been generated by the NS expanders hundreds of times and was included in tens of software applications. Nearly all these applications —and therefore nearly all the corresponding flow elements— have been regenerated or rejuvenated multiple times during the last few years, with limited changes to the flow element pattern.

The new enhanced architecture of the flow element as described in this paper, and its implementation in the expanders, has been developed using a specific reference (test) application. Here, it was verified that the proposed solutions for transactional integrity performed as desired during system crashes. Moreover, it was validated via the reference application that a default set of parameters actually resulted in a behavior identical to the previous implementation of the flow element.

After this testing and validation 8 additional applications (containing a total of 31 flow elements) were regenerated using the new version of the NS expanders (containing the improved flow element pattern). Therefore, from that point in time, all 8 applications could make use of the additional requirements incorporated in the flow element, if preferred. These applications range from administrative applications (e.g., supporting the master thesis assessment of students or processing VAT tax declarations), to more industrial applications (e.g., supporting data hubs for energy providers or monitoring photovoltaic solar panels). The existing default behavior was tested in all 8 applications, whereas the new enhanced functionality was applied within two applications. Already one of these applications has been put into production (i.e., is being used) while running the improved flow element pattern.

VI. Conclusion

By discussing a specific NS element pattern and its evolution (resulting in the evolutionary enhancement of sets of software applications), this paper is believed to make several contributions. With regard to theory, we showed how the NST approach can be used for the simultaneous introduction of enhanced capabilities in (large) sets of information systems.

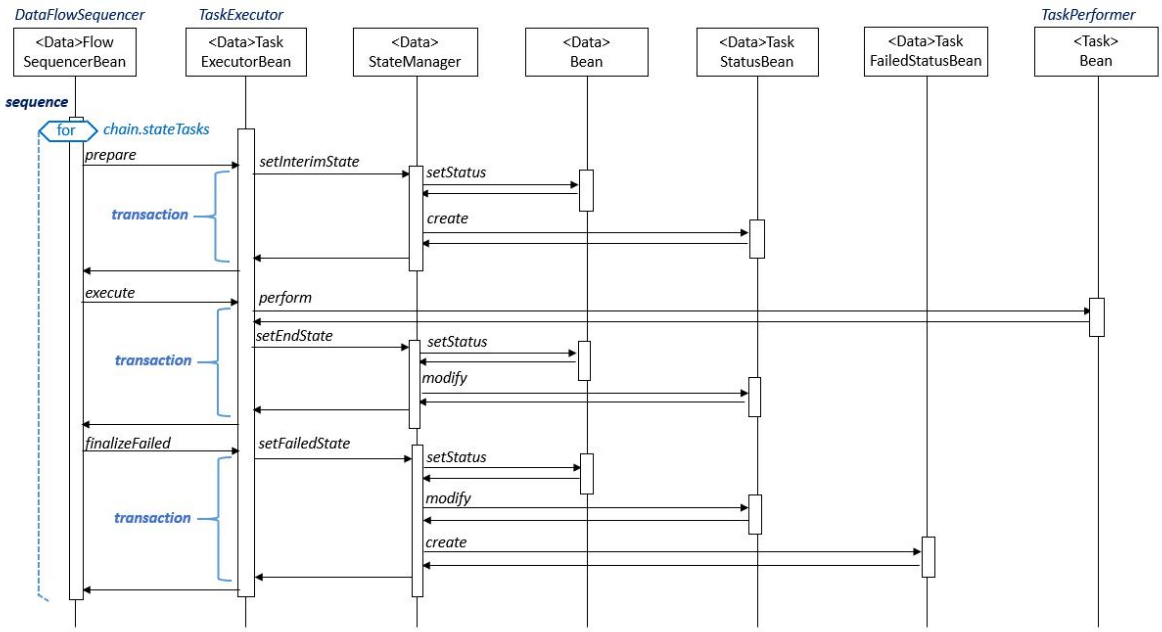

Figure 5. A sequence diagram containing the sequencing and transactional task processing in the improved flow element pattern.

This proves the feasibility of regenerating or rejuvenating (large) sets of information systems, and therefore enhancing their capabilities and/or modernizing the technologies used, while at the same time preserving existing functionality. For practitioners, this paper contributes to the design and documentation of actual workflow engines, processing state machines in a demanding high throughput environment.

Our paper has some limitations as well. First, the described set of information systems that has been regenerated and tested, is still somewhat limited, and only one regenerated application has been put into production using the enhanced pattern. Second, the proposed pattern for high throughput processing by state machine workflow engines has been designed and tested by several (but a limited amount of) experienced software developers, but has not yet been validated by a large amount of experts. A more advanced validation is however the ultimate goal of the NST approach: to validate and improve various software patterns through the collaborative efforts of many experts, and regenerate thousands of applications using these improved patterns. Third, our current discussion was mainly performed by means of a high-level overview (e.g. by using and describing sequence diagrams). A formal representation of the precise impact of using the (updated) workflow element, and the degree to which it is able to avoid the occurrence of combinatorial effects, is not provided in this work. Such further validations, improvements and formal representations are therefore considered as part of future research.

REFERENCES

[1] R. Agarwal and A. Tiwana, "Editorialevolvable systems: Through the looking glass of is," Information Systems Research, vol. 26, no. 3, 2015, pp. 473–479.

[2] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, special Issue on Software Evolution, Adaptability and Variability.

[3] ——, "Towards evolvable software architectures based on systems theoretic stability," Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116.

[4] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[5] M. Op't Land, M. Krouwel, E. Van Dipten, and J. Verelst, "Exploring normalized systems potential for dutch mods agility: A proof of concept on flexibility, time-to-market, productivity and quality," in Proceedings of the 3rd Practice-driven Research on Enterprise Transformation (PRET) working conference, Luxemburg, Luxemburg, September 2011, pp. 110–121.

[6] G. Oorts, P. Huysmans, P. De Bruyn, H. Mannaert, J. Verelst, and A. Oost, "Building evolvable software using normalized systems theory : a case study," in Proceedings of the 47th annual Hawaii international conference on system sciences (HICSS), Waikoloa, Hawaii, USA, 2014, pp. 4760–4769.

[7] P. Huysmans, P. De Bruyn, G. Oorts, J. Verelst, D. van der Linden, and H. Mannaert, "Analyzing the evolvability of modular structures : a longitudinal normalized systems case study," in Proceedings of the Tenth International Conference on Software Engineering Advances (ICSEA), Barcelona, Spain, November 2015, pp. 319–325.

[8] P. Huysmans, J. Verelst, H. Mannaert, and A. Oost, "Integrating information systems using normalized systems theory : four case studies," in Proceedings of the 17th IEEE Conference on Business Informatics (CBI), Lisbon, Portugal, July 2015, pp. 173–180.

[9] P. De Bruyn, P. Huysmans, and J. Verelst, "Tailoring an analysis approach for developing evolvable software systems : experiences from three case studies," in Proceedings of the 18th IEEE Conference on Business Informatics (CBI), Paris, France, August-September 2016, pp. 208–217.

[10] M. Lehman, "Programs, life cycles, and laws of software evolution," in Proceedings of the IEEE, vol. 68, 1980, pp. 1060–1076.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley Professional, 1994.

[12] T. Stahl, M. Völter, J. Bettin, A. Haase, and S. Helsen, Model-Driven Software Development. Wiley, 2006.