Generic Self-Learning Context Sensitive Solution for Adaptive Manufacturing and Decision Making Systems

Dragan Stokic, Ph.D., Sebastian Scholze, Oliver Kotte
Institute for Applied System Technology, ATB-Bremen,
Bremen, Germany
dragan@atb-bremen.de

***Abstract*— The paper investigates applications of context sensitivity to achieve high adaptivity of systems. A generic approach for context sensitivity, based on self-learning algorithms, is proposed aiming at a wide scope of systems. The approach is applied for high adaptation and integration of complex, flexible manufacturing systems as well as complex decision making systems, e.g., in software engineering. The proposed solution includes context extractor, adapter and self-learning modules allowing for adapting of the process and/or decision support systems depending on the extracted context. Both context extraction and adapter are continuously learning and improving their performance. Service Oriented Architecture (SOA) principles are used to implement these modules. The generic solution and specific applications in various manufacturing environments and decision making processes in software engineering are presented. The paper is one of first attempts to develop holistic context sensitive solution applicable to various systems.**

Keywords - context sensitivity;self-learning systems; adaptive manufacturing systems; context model; descision making systems; software engineering

I. INTRODUCTION

The objective of the work presented in this paper is to investigate how context sensitivity can be used to achieve a high adaptivity of various systems. In order to facilitate applicability of the approach to wide scope of systems, self-learning features are introduced. The idea is investigate application of a new generic context sensitive solution to wide scope of complex systems and decision support systems (DSS). The key assumption of the work is that the proposed generic self-learning context sensitive solution can be easily adjusted to allow for adaptation of wide scopes of systems. The context sensitivity allows for observation of changes in circumstances in which a system is operating, which in turn allows for a dynamic adaptation of the system to these varying conditions. In order to explore applicability of the proposed generic solution to a wide spectrum of systems, in this paper the emphasis is on its application for the systems in two very different domains: complex manufacturing systems and complex DSS in engineering domain.

Implementation of complex models and algorithms for such context sensitive systems require powerful Information and Communication Technology (ICT) infrastructures. A service oriented approach opens entire new perspectives for self-adapting systems. It is likely that approaches based on SOA principles, using distributed networked embedded services in device space (sensors, controllers, etc.), are the most appropriate for implementation of self-learning adaptive systems in general and specifically complex manufacturing systems and DSS.

The paper is organized in the following way. Section II provides a brief overview of the state-of-the art. In Section III, the overall concept is elaborated. In Section IV, the applications of the proposed concept are briefly described, while in Section V the future plans are indicated.

II. SURVEY OF THE STATE-OF-THE ART

Context Awareness is a concept propagated in the domains of Ambience Intelligence (AmI) and ubiquitous computing. It is the idea that computers can be both sensitive and reactive, based on their environment. As context integrates different knowledge sources and binds knowledge to the user to guarantee that the understanding is consistent, context modeling is extensively investigated within Knowledge Management (KM) research [1]. The current research on knowledge context is primarily oriented towards capturing and utilization of contextual data for actionable knowledge [2-4]. A number of systems to handle context awareness were proposed by the research community [5-7]. However, a holistic approach to application of context sensitivity to a wide scope of systems, as proposed in this paper, is not elaborated up to now.

One of the key problems is how to extract context from the knowledge process. In the research presented in this paper is decided (see the text to follow) to model context with ontologies, and, therefore, context extraction mainly is an issue of context reasoning and context provisioning: how to infer high level context information from low level raw context data. Based on the formal description of context information, context can be processed with contextual reasoning mechanisms [8-9].

Defining context for applying context awareness can be difficult [10-11]. Existing formal context models support formality and address a certain level of context reasoning [12 -13]. For example, the modeling of context in the case of DSS for Quality Assurance (QA) in Software (SW) development process presents an additional challenge, which has not been addressed up to now, as the services are highly dynamic and reside in distributed environments. With the emerging and maturing of semantic web technologies, Ontology based context modeling becomes a new trend both in academy and industry. Present research on context modeling is mostly focused on ontology. Compared to other methods, ontology based method has many advantages as it allows for context-modeling at a semantic level, establishing a common understanding of terms and meaning and enabling context sharing, reasoning and reuse [14].

In the area of self-learning systems, the research has demonstrated that the application of machine learning techniques, dynamic self-adaptation and operator's feedback in the loop promises to increase the intelligence of the overall system [15-17]. In production systems in particular, these methods have been proven to be especially useful for monitoring/diagnosis and control [18-20]. However, the applications of self-learning systems in, e.g., industrial practice are still in initial phase. In this paper a novel approach in such systems is presented, which is context aware, adaptive to contextual changes at run time and learns from adaptation and operator's action.

Of special interest for the work presented in this paper is SOA, i.e., the relation between context sensitivity systems and SOA. Scalable SOA holds promise for seamless integration, interoperation and flexibility in different environment [21-22]. But, there is a lack of adoption of overall SOA based self-adaptive systems in, e.g., discrete manufacturing environment and DSS. In this paper, SOA based context sensitive solutions are proposed as add-on features in existing infrastructure.

III. Proposed Approach for Self-Leaning Context Sesitivity

A. Proposed concept

The challenge is to define a solution able to handle wide scope of 'disturbances'/changes coming from either external conditions, or process/plant/DSS parameters changes, requiring control adaptations. The proposed approach is to (on-line) identify current dynamically changing context in which the system operates and to 'use' this identified context to adapt control. Therefore, the proposed approach includes a context extractor (as a generalized 'observer' providing current context) and an adapter (as 'active' control part) – see Fig. 1.

As explained above, SOA principles are the most appropriate for implementation of such systems. Context awareness, providing information about the process & plant or DSS and the circumstances under which the system operates, is a promising approach to assure, needed dynamic self- adaptation to changes in the context, including changes in processes & equipment parameters or parameters used within DSS. Wide applicability is enabled by self-learning features.

This approach has not been explored up to now. The basic assumption is that holistic, simultaneous and harmonized usage of self-learning context sensitivity, based on (on-line) extraction of a current context as indicated in Fig. 1, for adaptation of systems, is the effective way to assure considerable advantages regarding efficiency and availability of systems.

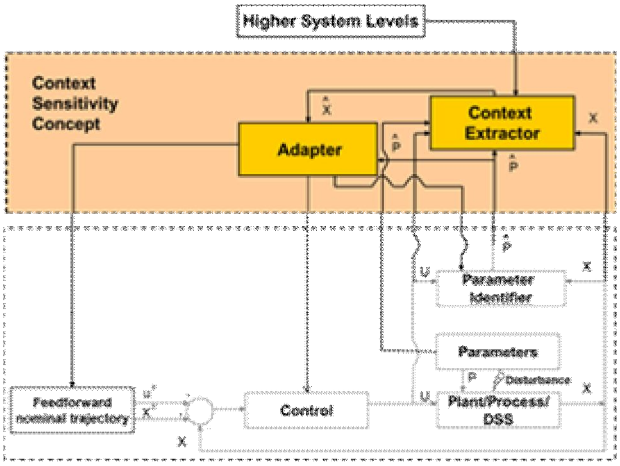

Figure 1. Approach proposed

B. General architecture

A generic solution for context based adaptation of systems is proposed which can be applied in various types of systems, specifically manufacturing systems and DSS. The features and functionality for the overall architecture are specified and developed as generic solutions easily adjustable to wide scope of systems. The system adapts to run time critical contextual changes and learns from it. Learning can also be enhanced by operator's feedback and experiences, especially in the cold start phase. The overall proposed reference architecture is illustrated in Fig. 2. The architecture is designed following SOA principles as an add-on to the standard control following the conceptual approach as presented in Fig. 1. The components of the proposed system include [23, 24]:

- Context Extractor, Adapter, Self-learning services - see the text to follow
- Validation module: the identified solutions are required to be validated, where the user can manually/automatically accepts/rejects any new solution. The validation sends the feedback to the other modules.

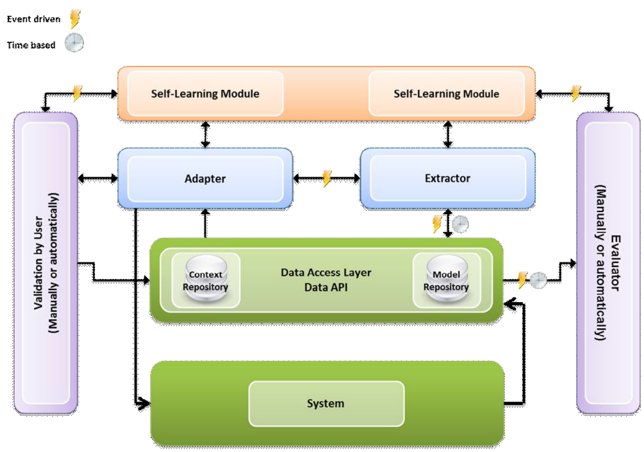

Figure 2. Proposed reference architecture [23, 24]

• Evaluator: Performance of adaptation and context extraction are measured by the evaluator either manually (in cold start) via operator's feedback or automatically via mapping against objective functions at run time. Evaluation results are sent to the Learning modules.

• Data access layer: Generic component, responsible for accessing the various system layers.

• Data Processing: These services are responsible for the bidirectional processing of information and perform, e.g., pre-processing of monitored raw data acquired via the data access layer, before the context is identified. Main functionality is to transform the raw data in a format which serves as basis for context identification. The Model Repository contains ontology based plant specific models for equipment, production processes/products and/or DSS. The models are shared by different software components at run time. The Context Repository allows update and storage of extracted/processed contextual information for later retrieval. Information flow among the modules is event driven in some cases and time based in other cases.

• Service Infrastructure: underpinning framework ensuring information is securely gathered from trusted context data sources and that the control updates are communicated to control systems with appropriate levels of authentication. The communication authentication components ensure seamless and secure connectivity with existing information system communication protocols and security mechanisms.

C. *Context model*

As indicated above, the basic assumption of the proposed approach is that monitoring the plant/process/DSS and its environment enriched with context can help to be aware of any potential change that may have an effect the system behaviors and/or decision making process. Therefore, a research key is the definition of a „holistic" and dynamic context model and ontologies to enable context awareness, allowing taking into account the context of the system operation and /or decision making processes (e.g., processes, equipment and product information, users, teams, etc.) [25]. As ontology allows for knowledge sharing, logic inference and knowledge reuse, it is a widely accepted approach for semantic-rich context modeling. Therefore, ontology is used for context modeling. Based on a context ontology, logic based context reasoning is realized, such as consistency validating, subsumption checking, etc. More importantly, domain specific rules are defined to infer implicit context from explicit context, and high level context from low level context. Other statistic and machine learning approaches can be adopted for non-logic context reasoning.

The definition of the context model is a key approach to assure usability of the proposed solution in different domains. The application of the solution to a specific domain normally requires adjustment of the context model. Therefore, a general and extensible context model is proposed. It is in a format that meets several requirements: help to describe and capture context easily; help to manipulate context; facilitate context consumption by KM services. The proposed Context Model defines two models as

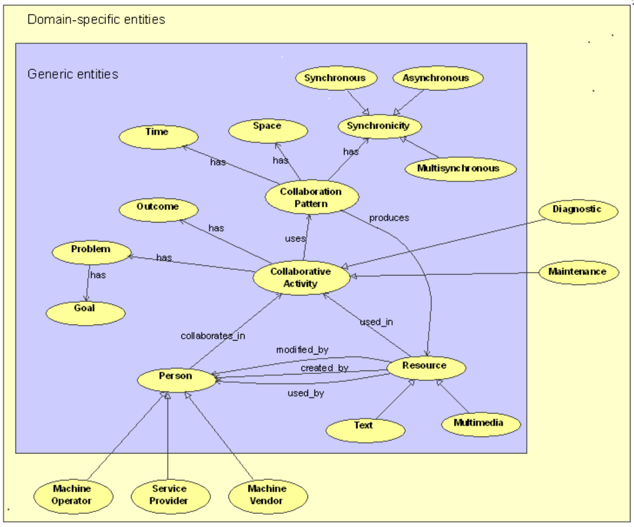

Figure 3. Context Model – ontology genereic and domain specific

sets of ontologies: Generic Context Model and Sector-Specific Context Model. Figure 3 presents an example of two models. The research is focused on development guidelines to effectively define context models for various applications. The basic principles for context modeling were followed: (1) Support description of main context: In practices, all context information cannot be modeled. The context model should consider those most relevant concepts and information needed to meet the requirement of context sensitive adoption. (2) Model the context that is (easy) acquirable: The context factors considered should be identifiable and acquirable, whether provided through computer monitoring automatically, or by user input explicitly. (3) Trade-off between investment of context modeling/extracting and effects of context sensitive adoption: Intuitively, if we could model as much context factors in as much details, the accuracy of context will be higher. However, this does not come for free: more time and efforts are needed for context modeling and more computing resources are needed to handle the context, which will bring deficiency to the adoption process.

D. *Context extractor*

The Context Extractor is the set of embedded services responsible for identifying changes in the context of the environment. The current identified context is used to extract available context knowledge. The results of the Context Extractor are used in the Adapter which is responsible for updating the system behavior. The Context Extractor uses all monitored "raw data" provided via the data access layer to derive the machine's current contextual situation. Using the ontology/context model the monitored data are evaluated and the context extracted. Based on the identified context, situations can be compared to previous ones and stored. A continuous process, coordinating with the monitoring and followed by the adaption process is built based on reasoning around the extraction of a contextual situation. The main elements of the context extractor are:

 ISBN: 978-1-61208-319-3

• Context Identification: It is responsible for the identification of the current context, based on monitored raw data, the adopted ontology and historic context information stored in the context repository.

• Rule Engine: Responsible for providing appropriate rules for the identification of context – context reasoning.

• User Interfaces: User interfaces for maintaining and administering the rules and the context repository.

• Application Specific Modules: This set of services includes application specific components and user interfaces.

E. Adapter

The Adapter is the set of services responsible for updating system behavior (locally and/or globally) in response to a change of context in the environment. The adaptation is based upon available context knowledge to identify the best set of rules to employ in each context. The Adapter is informed by the Context Extractor about a variation on the system (change of context) and adapts the system to handle the new situation. The Adapter is guided by a set of rules that describe how the system should behave in each particular context. These rules can be updated through learning based on lifecycle history data, context and user validation. Although triggered by the Context Extractor whenever a change of context is detected, the Adapter is able to periodically enquire the Learning module to check for a new set of rules that promise to improve system performance in accordance with production objectives. This information can be then presented to the user in order to validate or not the new proposal. In the positive scenario, this proposal will be saved and become the new de facto set of rules, which will delineate system behavior for that particular context. Each application case comprises a specific set of rules and output, but the Adapter structure is generic:

• Rule Engine: an engine framework that is able to process application specific rules.

• Context Action Selector: These services are responsible for the definition of the adaptation depending on a particular context. These services interpret the results obtained by running the rules in the Rule Engine.

• Context Action Optimizer: Set of services to update of the current set of rules, either due to a Learning module suggestion (validated by the user) or by direct input.

• Application Specific Module: This set includes services that need to be developed for each application individually. Application specific rules and information templates are also defined here.

F. Self-Learning

The proposed concept for self-learning represents the capability of a complete system to learn along its life-cycle based on the experience retrieved from past and current relations and share of information between its different elements in a distributed manner. Learning services allow the system to learn relying on Data Mining (DM) and operator's feedback to update execution of adaptation and context extraction at run time. The results (adaptation rules) are suggested to the Validation module, and must be backed up by the user (either upon request or automatically). Adaptation rules and context identification procedures are updated accordingly. DM has been considered to be appropriate approach for the concerned applications in complex production systems and DSS. The approach relies upon the extraction of previously unknown and potentially useful patterns from data sets. DM is often referred in the context of Knowledge Discovery in Databases (KDD), which consists in selecting data samples from a large database, treating it and analyzing it for pattern extracting. The sampling process is used because, most of the times, the total amount of data is too large to be fully analyzed. There are several types of methods for DM to discover the patterns in the data. The generic solution prosed in this paper includes a set of algorithms from which for each specific application the most appropriate one can be selected within system set-up. The cold start is applying normally supervised learning. The methods included in the generic solution are: Classification – learning of a function aiming to map the data inside a set of classes in the best way possible, Regression - focuses on the relationship between a dependent variable and one or more independent variable, Clustering – seeks to find a finite number of categories or clusters to classify the data, Association Rule Learning - Discovers relations between variables in databases and describes them as rules.

IV. APPLICATIONS

As indicated above, the proposed concept is applicable to wide scope of systems. Two specific applications were investigated in practice: application for the control of manufacturing processes and application for adaptation of decision support system within software engineering.

A. Application for adaptive manufacturing systems

The approach has been applied for adaptation of various manufacturing processes to environmental and parameters changes in several manufacturing companies in various sectors [25]. One of the applications was to achieve high adaptation of the machines in shoe industry to the changing conditions. Therefore, the proposed solution is applied to react to the changing situations/contexts associated with variations in different parameter sets. The parameter variations of both controlled plants (machines) and environments in which the systems are operating are in terms of pressure and temperature, speed frequencies of drives and pumps, proper material mix ratio and filling of materials into shoe forms. For example, one of the scenarios addresses synchronicity of the different valve circuits when dosing of several components. As the valve synchronization is designed by a mechanical system, and/or due to valve abrasion, with different valve opening times, caused by, e.g., different force requirements or different air supply, it may come to flaws in the product. Currently, the synchronization can only be adjusted by a technician during downtime of the machines. By implementation of the proposed self-learning solution an automatic adjustment of the valve switching to different conditions is achieved. The context monitoring serves as a basis for identifying valve adjustment parameters. Device centric infrastructure is based on machines, the corresponding PLC' and operational PCs. Ethernet, Field

 ISBN: 978-1-61208-319-3

bus, CAN Open, Profibus based communication protocols are used to communicate within the equipment level. Main information sources from device layer are sensors for pressure/temperature, pumps/drives for speed frequencies and controllers for parameter sets. Enterprise level parameters are accessible in plant database, which are mostly associated with varied customization of shoe types and sizes related to the different production lot numbers. Operators adjust the boundary values or parameter sets through the operational PC of machines. The context extractor services extract and process relevant parameters from the database to input to the Adapter set of services. The adapted solutions, in terms of adjusted parameters, changes to speed frequencies of pumps/drives, changes in the process cycle time, etc., are sent to the Validation module for user's feedback. Based on adaptation results and operator's feedback, the learning services learn how changing cycle times and ambient conditions are influencing the production process (e.g., above explained valve synchronization) and also update the rules for context identification, adaptation and extension. From application point of view, the main advantages are that the operator is not forced to bring highest skills to run the equipment, but can more concentrate on the core processes while the production equipment is self-controlling all relevant parameters to keep the process running. Contrary to the classical adaptive control solutions, the generic self-learning modules integrating more intelligence into the production, are easily applicable to various machines/processes, gaining a higher benefit for the producer while less investment in the human resources are required.

B. Applicaton for descision support system in software engineering

The main objective of the application was to create context sensitive decision support services within flexible QA of SW development projects [26]. The new QA process is supported by the proposed solution composed of several knowledge, context sensitive services that are able to detect changes in the scope and requirements of a SSW component (or changes in its development process) and provide the adequate set of assessments as a basis for an accurate measurement of the quality of the process and product at any time and allow for effective decision making within QA. The Internet Services monitor the different stages of the software development process, interoperating with the existing applications and systems to provide quantitative information about the quality of each phase (i.e., project management, requirements gathering, functional and technical design, development and testing), the project as a whole and the resulting product. They also monitor context under which the SW is developed and decisions on QA have to be made. Data obtained in real-time by the monitoring services are used in an indistinctive way by software engineers, designers, developers, testers and managers alike for different collaborative decision making. In the case of services for monitoring, analyzing and enhancing SW development processes the notion of context refers to preferences and skills of users, physical capabilities of the equipment and environment conditions, coming from different kinds of information sources like bug-tracking systems, code repositories, etc. The key ontology, entitled Activity-Centric Collaboration Ontology (ACCO), allows for representing the context of collaborative work situations in the form of explicit machine interpretable knowledge. It enables intuitive representation of knowledge about collaborative work and serves as the base for further knowledge sharing, refining and reuse. It explicitly describes collaboration related activities, people, resources, and the relationships among them. The problem is how to better describe dynamic SW engineering tasks and decision making processes (i.e., which aspects regarding collaboration are relevant for decision making) in domain specific collaborative work, and how to integrate the context model into infrastructure and tools, in order to enhance context-sensitivity of these tools, and to facilitate context extraction from the content provided by the monitoring services. The appropriate relations between the concepts and various associated attributes in the selected context model are elaborated. The extractor extracts the context changes and informs the adapter to adapt the DSS parameters/rules accordingly. The approach is assessed in 2 different cases in order to validate the results under different conditions. The first case belongs to a large software company developing large Internet projects based on Rational Unified Process methodology. The second case belongs to a SME developing complex projects based on agile methodologies The analysis and testing indicate good potentials to improve QA process by the context sensitive decision support services. The benefits are highly depended on the complexity of the development project and their dynamics. The cost/benefit ratio asks for a deeper analysis of the specific company's development processes, but in the specific cases of complex SW processes the proposed approach is likely to bring benefits of more than 30% w.r.t. to classical DSS.

V. CONCLUSIONS

The research presented in the paper is one of the first attempts to apply a self-learning context sensitivity solution to support adaptivity of a wide scope of systems. The main benefit of the proposed generic solution is that it can be easily adapted to fit specific conditions of each system. The applicability of the solution to manufacturing systems and DSS in QA for SW development is demonstrated. The generic innovative context model has been proposed and developed, and the main adjustment of the solution to the specific systems is a definition of specific context models relevant for manufacturing systems and software engineering process. New applications of such approach for adaptations in flexible manufacturing and effective decision making in agile SW development are elaborated. These specific applications investigated in practice demonstrate that the proposed concept and ICT solution are promising to be applicable to wide scope of system, asking for minimal adjustments. As the applications of context sensitivity for both flexible manufacturing systems and DSS for software development processes have not yet been sufficiently researched, the presented solutions are also novelty in these two domains.

Many research problems, however, are still under consideration. The decisions which raw data are worthy to on-line collect/provide by monitoring services (which means efforts/coasts to integrate services with various systems, which include these data) in order to better extract the context and support decisions making, have to be made on case basis and are specific for each applications. The methodology on how to analyze cost/benefit ratio for various applications is developed. Alternatively, feature selection approaches fining the 'best' subset of features for the system adaptation have to be investigated [27]. The key research issues to be solved are how to refine the context models. Automatic update of the context model based on the observed changes in environment is a subject of the further research. Another problem under study is how to assure better automatic evaluation and validation of the results to make learning process more autonomous.

ACKNOWLEDGMENT

This work is partly supported by the Self-Learning (Reliable self-learning production system based on context aware services) project of EU's 7th FP, under the grant agreement no. NMP-2008-228857 and the U-QASAR (Universal Quality Assurance & Control Services for Internet Applications with Volatile Requirements and Contexts) project of EU's 7th FP, under the grant agreement no. ICT-FP7 -318082. This document does not represent the opinion of the European Community, and the Community is not responsible for any use that might be made of its content.

REFERENCES

[1] M. Sorli, and D. Stokic, Innovating in Product/Process Development, Springer-Verlag, Heidelberg, London, New York, 2009.

[2] D. Mladenić, "Active Project Website"., Retrieved January 10th, 2012, from http://www.active-project.eu/fileadmin/public_documents/D231_context_definition_and_representation.pdf, 2009, [retrieved December 2013]

[3] Y. Tao, X. Tianyuan, and Z. Linxuan, "Context-centered design knowledge management", Computer Integrated Manufacturing Systems, 10 , 2004, pp. 1541-1545.

[4] J. H. Ahn, J. H. Lee, K. Cho, and J. S. Park, "Utilizing knowledge context in virtual collaborative work", Decision Support Systems, 39, 2005, pp. 563-582.

[5] P. Bellavista, A. Corradi, R. Montanari, and A. Toninelli, "Context-aware semantic discovery for next generation mobile systems", IEEE Comm. Magazine, 44, 2006, pp. 62-71.

[6] S. Kim, E. Suh, and K. Yoo, A study of context inference for Web based information systems, Electronic Commerce Research and Applications 6, 2007, pp. 146-158.

[7] J. W. Chang and Y. W. Kim, Design and implementation of middleware and context server for context awareness, High Performance Computing and Communications, Proceedings, LNCS 4208, 2006,pp. 487-494.

[8] M. Luther, et al., Situational reasoning – a practical OWL use case. Chengdu, Jiuzhaigou, China, 2005.

[9] J. Forstadius,O. Lassila, and T. Seppänen, RDF-Based Model for Context-Aware Reasoning in Rich Service Environment, Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE Computer Society, 2005

[10] K. A. Dey, Understanding and Using Context. Personal and Ubiquitous Computing Journal, 1, 2001,pp. 4-7.

[11] T. Kindberg, et al., People, places, things: Web presence for the real world, Third IEEE Workshop on Mobile Computing Systems and Applications, 2000.

[12] K. Henricksen, J. Indulska, and A. Rakotonirainy, Modeling Context Information in Pervasive Computing Systems, Pervasive Computing, 2002, pp. 79.

[13] A. Ranganathan and R.H. Campbell, A middleware for context-aware agents in ubiquitous computing environments. Springer-Verlag, Rio de Janeiro, Brazil, 2003.

[14] N. Georgalas, et al. Towards a Model-Driven Approach for Ontology-Based Context-Aware Application Development: A Case Study. In: MOMPES '07. 4th Intern. Workshop on Model-Based Methodologies for Pervasive and Embedded Software, 2007. E. Alpaydin, Introduction to machine learning. The MIT Press. 2004.

[15] D. Michie, D. Spiegelhalter, C. Taylor, and J. Campbell, Machine learning, neural and statistical classification. Ellis Horwood Series In Artificial Intelligence, 1995, p. 289

[16] J Carbonell, R. Michalski, and T. Mitchell, An overview of machine learning. Machine learning: An artificial intelligence approach, Tioga Press, Palo Alto, 1, pp. 3–23, 1983.

[17] S. C. Liu and S. Y. Liu, 'An Efficient Expert System for Machine Fault Diagnosis,' The Intern. Journal of Advanced Manufacturing Technology, vol. 21, pp. 691-698, 2003.

[18] N. Palluat, D. Racoceanu D, and N. Zerhouni 'A neuro-fuzzy monitoring system application to flexible production systems,' Comput. Ind., vol. 57, pp. 528-538, 2006.

[19] Y. Jianbo, X. Lifeng, and Z, Xiaojun, 'Intelligent monitoring and diagnosis of manufacturing processes using an integrated approach of KBANN and GA,' Comput. Ind., vol. 59, pp. 489-501, 2008.

[20] L. Monostori, 'AI and machine learning techniques for managing complexity, changes and uncertainties in manufacturing,' Engineering applications of artificial intelligence, vol. 16, pp. 277-291, 2003.

[21] L. Ribeiro, J. Barata, A.W. Colombo, and G. Jammes, 'A Generic Communication Interface for DPWS-based Web Services,' in IEEE International Conference on Industrial Informatics INDIN, Daejeon, Korea, IEEE, 2008.

[22] W. Mahnke and S-H. Leitner, "OPC Unified Architecture - The future standard for comm. and information modeling in automation" 3/2009 ABB Review 3/2009, pp. 56-61

[23] Self-Learning, EU project NMP-2008-228857 "Reliable self-learning production system based on context aware services", Public report, 2010.

[24] D.Stokic, S. Scholze, and J. Barata, Self-Learning Embedded Services for Integration of Complex, Flexible Production Systems, Proc. of IECON Symposium, Melbourne, 2011, pp. 652-658.

[25] T. Strang and C. Linnhoff-Popien, A Context Modeling Survey, Workshop on Advanced Context Modelling, Reasoning and Management as part of the Conference on Ubiquitous Computing - The Sixth International Conference on Ubiquitous Computing, 2004.

[26] O. Kotte, A. Elorriaga, D. Stokic, and S. Scholze, Context Sensitive Solution for Collaborative Decision Making on Quality Assurence in Software Development Processes, Proc. of the KES-IDT Symp., 2013.

[27] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research, 2, pp. 1157-1182, 2003.