TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Yulia Evchina

A Hybrid Context-aware Middleware for Relevant Information Delivery in Multi-Role and Multi-User Monitoring Systems

An Application to the Building Management Domain

Julkaisu 1549 • Publication 1549

Tampere 2018

Tampereen teknillinen yliopisto. Julkaisu 1549
Tampere University of Technology. Publication 1549

Yulia Evchina

A Hybrid Context-aware Middleware for Relevant Information Delivery in Multi-Role and Multi-User Monitoring Systems

An Application to the Building Management Domain

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Festia Building, Auditorium Pieni Sali 1, at Tampere University of Technology, on the 1st of June 2018, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2018

Doctoral candidate:	Yulia Evchina Laboratory of Automation and Hydraulic Engineering Faculty of Engineering Sciences Tampere University of Technology Finland
Supervisor:	Jose Luis Martinez Lastra, Professor Laboratory of Automation and Hydraulic Engineering Faculty of Engineering Sciences Tampere University of Technology Finland
Pre-examiners:	Marga Marcos Muñoz, Professor Department of Automatic Control and Systems Engineering University of the Basque Country Spain
	Alessandro Margara, Assistant Professor Department of Computer Science and Engineering Politecnico di Milano Italy
Opponent:	Frank Golatowski, Senior Research Associate and Lecturer, Dr.-Ing. Institute of Applied Microelectronics and Computer Engineering University of Rostock Germany

ISBN 978-952-15-4146-9 (printed)
ISBN 978-952-15-4157-5 (PDF)
ISSN 1459-2045

Evchina, Yulia: A Hybrid Context-aware Middleware for Relevant Information Delivery in Multi-Role and Multi-User Monitoring Systems: An Application to the Building Management Domain

Faculty of Engineering Sciences, Tampere University of Technology, Finland 2017

Keywords: Building Management Systems (BMS), Complex Event Processing (CEP), Semantic Web Technologies, alarm and notification management, context-aware systems

Abstract

Recent advances in information and communications technology (ICT) have greatly extended capabilities and functionalities of control and monitoring systems including Building Management Systems (BMS). Specifically, it is now possible to integrate diverse set of devices and information systems providing heterogeneous data. This data, in turn, is now available on the higher levels of the system architectures, providing more information on the matter at hand and enabling principal possibility of better-informed decisions. Furthermore, the diversity and availability of information have made control and monitoring systems more attractive to new user groups, who now have the opportunity to find needed information, which was not available before. Thus, modern control and monitoring systems are well-equipped, multi-functional systems, which incorporate great number and variety of data sources and are used by multiple users with their special tasks and information needs.

In theory, the diversity and availability of new data should lead to more informed users and better decisions. In practice, it overwhelms user capacities to perceive all available information and leads to the situations, where important data is hindered and lost, therefore complicating understanding of the ongoing status. Thus, there is a need in development of new solutions, which would reduce the unnecessary information burden to the users of the system, while keeping them well informed with respect to their personal needs and responsibilities.

This dissertation proposes the middleware for relevant information delivery in multi-role and multi-user BMS, which is capable of analysing ongoing situations in the environment and delivering information personalized to specific user needs. The middleware implementation is based on a novel hybrid approach, which involve semantic modelling of the contextual information and fusion of this information with runtime device data by means of Complex Event Processing (CEP). The context model is actively used at the configuration stages of the middleware, which enables flexible redirection of information flows, simplified (re)configuration of the solution, and consideration of additional information at the runtime phases. The CEP utilizes contextual information and enables temporal reasoning support in combination with runtime analysis capabilities, thus processing ongoing data from devices and delivering personalized information flows. In addition, the work proposes classification and combination principles of ongoing system notifications, which further specialize information flows in accordance to user needs and environment status.

The middleware and corresponding principles (e.g. knowledge modelling, classification and combination of ongoing notifications) have been designed contemplating the building management (BM) domain. A set of experiments on real data from rehabilitation facility has been carried out demonstrating applicability of the approach with respect to delivered information and performance considerations. It is expected that with minor modifications the approach has the potential of being adopted for control and monitoring systems of discrete manufacturing domain.

Preface

The research that is documented in this thesis was finalized within the faculty of Engineering Sciences, laboratory of Automation and Hydraulic Engineering of Tampere University of Technology, Finland and was carried out during 2011-2017. Financial support for the research has been provided through the following sources:

- ASTUTE (Pro-active decision support for data-intensive environments), 2011-2012: funded by ARTEMIS First Call 2010 Program under Grant Agreement number 269334;

- Doctoral Programme of the President of the Tampere University of Technology, 2012-2017.

Acknowledgements

This thesis marks important, but intermediate point of exciting journey of pursuing and testing my dreams to become a true researcher. The journey would not have been possible without direct or indirect involvement of many amazing people, whom I had a luck to meet through many years before and during PhD studies.

First, I would like to express my deep gratitude to my supervisor Prof. Jose Luis Martinez Lastra for giving me the opportunity to learn and work at FAST-Lab. and for supporting and improving my ideas about the directions of this work.

I would like to thank the pre-examiners of this thesis Prof. Marga Marcos Muñoz (University of the Basque Country) and Assistant Professor Alessandro Margara (Politecnico di Milano) for their time and effort in reviewing the manuscript and providing valuable feedback.

I would also like to thank Timo Palmroth and Tommi Vahtera from THT Control Oy for providing me with data from monitoring system, which made it possible to test my ideas.

Many thanks go to the university members giving additional support to the doctoral program from different sides. To Hanna Maavirta and Taina Torvinen for making it easier to get the information and needed help. To Ulla Siltaloppi for always welcoming atmosphere and enormous help in matters of dealing with Finland as a foreign country. To Anna Nykänen for arranging amazing doctoral days and meetings and for answering numerous emails with endless questions.

I would like to warmly thank my colleagues and friends at FAST-Lab. for creating this unique and nice environment at the time of my PhD studies. Aleksandra for introducing me to general principles of research, especially in the beginning of this work. Juha for giving me examples of excellent programming skills in his programs. Andrei for his endless optimism, which inspires people around and teaches you to inspire people. Jani and Corina for discussions and small talks. Matti, Anne, and Sonja for the support in all numerous and often unexpected matters, which appear in the lab. My special thanks go to Anna and Angelica for our inspiring meetings and discussions about research and life in general, which were a valuable encouragement throughout the PhD years.

There are no words which would convey enough thanks to my parents, especially to my mother Svetlana, for giving me all unconditional support to start this journey, strength and courage to pursue my dreams. I am also deeply grateful to my sister Evgeniya for all the talks we shared and showing me the courage of her own in achieving goals.

Finally, I would like to express my deepest heartfelt gratitude to my husband Alexey for his enormous support during all these years and for sharing our dreams and making them come true together; and to our daughters, Emilia and Victoria, whose smiles make days brighter and laughs get mind much needed time off work.

Yulia Evchina
Tampere, April 2018

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ARM	Association Rule Mining
ASP	Answer Set Programming
BM	Building Management
BMS	Building Management System(s)
CEP	Complex Event Processing
CPPS	Cyber-Physical Production Systems
CPS	Cyber-Physical Systems
DSMS	Data Stream Management Systems
EDBS	Event-Driven Backward Chaining
EPL	Event Processing Language
FAS	Factory Automation Systems
HCI	Human-Computer Interaction
HMM	Hidden Markov Models
ICPS	Industrial Cyber-Physical Systems
ICT	Information and Communications Technology
IIoT	Industrial Internet of Things
IoT	Internet of Things
LSD	Linked Stream Data
MAS	Multi-Agent Systems
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
REST	Representation State Transfer
RIF	Rule Interchange Format

RS	Recommender Systems
SA	Situation Awareness
SCADA	Supervisory Control and Data Acquisition
SOA	Service Oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SSN	Semantic Sensor Network Ontology
SVM	Support Vector Machine
SW	Semantic Web
SWRL	Semantic Web Rule Language

Referred Publications

Published Manuscripts

This compendium thesis is based on the following publications:

Publication I Evchina, Y., & Martinez Lastra, J. L. (2016). An ICT-driven hybrid automation system for elderly care support: a rehabilitation facility study case. *Journal of Housing For the Elderly*, *30*(1), 52–74. https://doi.org/10.1080/02763893.2015.1129382 [JuFo 1]

Publication II Evchina, Y., Dvoryanchikova, A., & Martinez Lastra, J. L. (2013). Semantic information management for user and context aware smart home with social services. In *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support* (pp. 262–268). Ieee. https://doi.org/10.1109/CogSIMA.2013.6523856 [JuFo 0]

Publication III Evchina, Y., Puttonen, J., Dvoryanchikova, A., & Martinez Lastra, J. L. (2015). Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. *Engineering Applications of Artificial Intelligence*, *43*, 111–126. https://doi.org/10.1016/j.engappai.2015.04.008 [JuFo 1]

Publication IV Evchina, Y., & Martinez Lastra, J. L. (2016). Hybrid approach for selective delivery of information streams in data-intensive monitoring systems. *Advanced Engineering Informatics*, *30*(3), 537–552. https://doi.org/10.1016/j.aei.2016.07.006 [JuFo 1]

Publication V Evchina, Y., & Martinez Lastra, J. L. (2018). An approach to combining related notifications in large-scale building management systems with a rehabilitation facility case study. *Automation in Construction*, *87*, 106–116. https://doi.org/10.1016/j.autcon.2017.12.020 [JuFo 1]

Author’s Contributions

Publication I. “An ICT-driven hybrid automation system for elderly care support: a rehabilitation facility study case”

The doctoral student contributed the majority of the work in this article including concept and writing. Prof. Lastra contributed to the writing style by providing feedbacks. Ville Hämäläinen has performed implementation of the work under the doctoral student and Dr. Dvoryanchikova supervision; his work is acknowledged in the acknowledgements section of the paper.

Publication II. “Semantic information management for user and context aware smart home with social services”

The doctoral student has performed most of the work for this paper including formulation of the method, paper concept, and writing. Prof. Lastra and Dr. Dvoryanchikova have contributed in writing process by providing feedbacks.

Publication III. “Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems.”

The doctoral student has contributed the majority of the work: formulation of the method, partial implementation, paper concept, and writing. Dr. Puttonen has provided implementation support for majority of the tools used in the work. Dr. Dvoryanchikova has provided support in writing style; Prof. Lastra has supervised the work by providing feedbacks.

Publication IV. “Hybrid approach for selective delivery of information streams in data-intensive monitoring systems”

The doctoral student has done the work reported in this manuscript, specifically: proposing the method, implementation developments, concept of the paper, writing. Prof. Lastra has supervised the work and has contributed into conceptual vision of the paper.

Publication V. “An approach to combining related notifications in large scale BM systems with the example of rehabilitation facility use case.”

In this manuscript, the doctoral student has proposed the methods, performed implementation, developed concept of the paper, and wrote it. Prof. Lastra has supervised the work and has provided feedbacks during the process of creation of the paper.

1 Introduction

1.1 Background

Building Management System (BMS) is a digital control system enabling control and automation of mechanical and electrical equipment installed in the managed facilities. These facilities could be commercial, industrial, institutional, or private home buildings. The general purposes of such systems are to ensure operational performance, reduce energy consumption of buildings, and increase the comfort and safety of building occupants. Historically, BMS have greatly benefited from Factory Automation Systems (FAS), which paved the way to initial technological solutions of BMS. For example, due to FAS advancements, BMS had only one commonly accepted automation model consisting of management, automation, and field levels, as opposed to gradual evolution of FAS automation hierarchy with multiple models and standards [Sauter, Soucek, Kastner, & Dietrich, 2011].

Recent dramatic advances in information and communications technology (ICT) have made similarities between BMS and FAS even more prominent. Both of the system types actively adopt newly developed smart embedded devices [Sauter et al., 2011], feature integration of heterogeneous data sources ranging from sensors to information systems [Frost & Sullivan, 2016a], enhance human-computer interaction (HCI) through the usage of mobile devices, and emphasize importance of modularity for developed solutions [Martinez Lastra, 2004], [Delamer, 2006], [Puttonen, 2014]. Subsequently, even more flat two-tier architecture of BMS and FAS is predicted as a technological convergence of connected world. This architecture would consist of the field area networks and IP-based networks, integrating operation and information technologies of the systems [Frost & Sullivan, 2016b], [Sauter et al., 2011].

As a result, the technological advances have greatly extended functionalities and capabilities of BMS and FAS. Specifically, they have opened the potential possibility of advanced analytical support for better-informed decisions. This became possible by using newly available shop floor and field data on higher levels of the architectures. In addition, active usage of mobile devices for control and monitoring have enabled easy and comfortable access to this information from everywhere. Therefore, the diversity and availability of information have made BMS and FAS more attractive to new user groups, who now have the possibility to find needed information, which was not available before. Thus, modern BMS and FAS are well-equipped, multi-functional systems, which incorporate great number and variety of data sources, provide mobile interfaces for control and monitoring, and are used by multiple users with their special tasks and information needs.

1.2 Justification for the Research

In theory, diversity and availability of new data sources should lead to more informed users and better decisions. In practice, it overwhelms users with data, which potentially hinders important information and complicates the decision process. The concept of information overload could be well illustrated with inverted U-curve presented in Figure 1-1. At the beginning, the decision accuracy is growing with the increase of available information. However, at certain point, the heavy information load will confuse users, affect ability to set priorities, and make prior information harder to recall, which results in decreased decision accuracy [Eppler & Mengis, 2004]. Eventually, this would lead to reduced safety of the systems and higher probabilities of operator errors, which become even more noticeable under time pressure for decision-making.

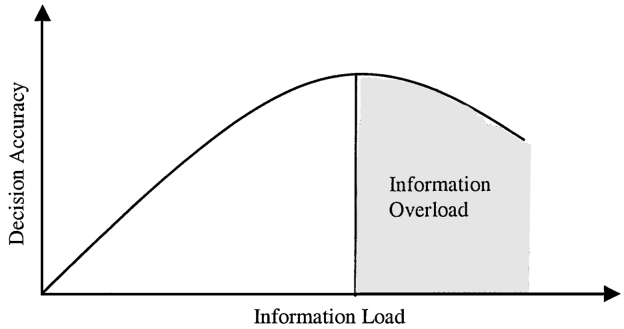


Figure 1-1: Information overload as inverted U-curve [Eppler & Mengis, 2004]

The operator errors are still one of the primary factor of system failures in control and monitoring systems. This fact is especially studied in process automation and discrete manufacturing systems, where system failures could be life threatening and/or are highly expensive. The operator-related statistics could be summarised as follows:

> *"The contribution of human error is in a range of 30-90% of all system failures, according to reports of incidents or accidents in a variety of industries"* [Seong, 2009, p. 157].
>
> *"...research indicates that nearly 80% of production downtime is preventable. And half of this is due to operator error. The monetary costs of this failure in the petrochemical industry alone are estimated at $20 billion per year"* [Larson, 2014].
>
> *"The 76% of SA [situation awareness] errors in pilots are traced to problems in the perception of needed information"* [Seong, 2009, p. 172].
>
> *"One point that research made clear is that the magnitude of information that facility operators are confronted with – from set points to equipment schedules, alarms to energy-use data – can be staggering. In some facilities, the data points can number in the tens of thousands. Facility operators want to know where the problems exist in their buildings. But the challenge of finding that information, quickly, can be overwhelming"* [Johnson Controls, 2015].

To summarize, ICT advances have overcome the problem of lack of information at the time of making decisions in control and monitoring systems. Instead, the number of data sources have become so tremendous that the problem of information overload has arisen. There is a need in development of solutions, which would reduce the unnecessary information burden to the user, while keeping him/her informed for better decisions. This need is even more aggravated by the fact that most BMS and FAS systems have enabled access through mobile devices, which pose limitations on amount of displayed information.

1.3 Research Description

1.3.1 Key Terms and Definitions

There is quite large degree of ambiguity in the definition of terms related to this research work; thus, the most important terms are defined here. Specifically, context, situation(s) and relevant information in application to control and monitoring systems are discussed.

The situation(s) and context are often used interchangeably, especially in the field of situation-aware and context-aware systems. More information about these systems will be given in sections 2.2.3 and 2.2.4 respectively; for now, the focus is put only on definitions and meaning of these terms with respect to control and monitoring systems. The closest definitions of context and situation were defined by Dey [2001]. The ***context*** is *"any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. The ***situation*** is *"a description of the states of relevant entities"*. With respect to control and monitoring systems, the interpretation of these definitions is formed by perception mechanisms of these systems. Specifically, control and monitoring systems perceive parts of the context information through its sensing devices (or other information sources). The perceived information and its interconnections form the states of the sensed environment or situations. In other words, both context and situation information describe the entities; the difference is that context is real world information, while situations are the reflections of the context through sensing mechanisms and other information sources of control and monitoring systems.

Regarding relevant information, there is no generally accepted term, so this description will be given based on general definitions. In general, term relevant means "*having significant and demonstrable bearing on the matter at hand*"[1] and *"connected with what is happening or being discussed"*[2]. Thus, ***relevant information*** could be defined as meaningful and important information at the time of its happening on the matter at hand. With respect to control and monitoring systems, this information must relate to matter at hand (e.g., tasks, needs, and environment status), consider ongoing context (e.g., in other context certain information pieces

[1] Merriam-Webster dictionary definition (accessed 19.10.2017): https://www.merriam-webster.com/dictionary/relevant

[2] Cambridge Dictionary definition (accessed 19.10.2017): https://dictionary.cambridge.org/dictionary/english/relevant?a=british

might no longer be relevant), and be meaningful in representation and planning of further actions.

1.3.2 Problem Statement

Given terms and definitions from the subsection above, the problem of information overload could be informally rephrased as too many context readings transforming into too many system situations, which still make too little sense to the users. Of course, there are some mechanisms in modern supervisory control and data acquisition (SCADA) and alarm management systems, which allow simple filtering of the information and notifications. However, the filtering is primarily based on sensed values with other options being too expensive to implement due to the need of deep system knowledge and limited configuration capabilities of such systems. In addition, these filtering options are at best oriented to different user groups with no consideration of personal information needs. Thus, the problem statement for this research work could be formulated as follows:

> *How to deliver relevant information to the user according to the ongoing context, e.g. status of the environment, user role, needs, and responsibilities?*

The publications included in this thesis answer the problem as follows:

- *What are the information needs in the considered domain?* Publication I answers this question and provides details of the field studies and usability tests summarizing requirements of information delivery and system functionality in Building Management (BM) domain for different user groups.

- *How to model contextual information of the environment?* Publication II answers this question and provides details of principles of knowledge modelling based on requirements highlighted in Publication I.

- *How to integrate and use the contextual model for relevant information delivery?* This question is answered by Publication III, which demonstrates the knowledge-based approach for relevant information delivery. This paper also features reconfiguration and context-aware scenarios demonstrating flexibility of the approach for redirection of information flows and simplified system configuration.

- *How to consider ongoing context at system runtime?* Publications IV addresses this question and provide details of the novel hybrid approach for relevant information delivery. The paper provides implementation details of developed middleware supporting this approach and general guidelines for developing similar solutions.

- *How to reduce information load in the form of notifications to each user of the system?* The use of contextual model at the system runtime enables tracking and detection of multiple situations. On the one hand, the notifications become more specific to certain situations. On the other hand, it increases the information flow delivered to the users. The Publication V addresses this issue and proposes the way of classification and combination of ongoing notifications in BMS at runtime.

The approach incorporates in the middleware for relevant information delivery, thus enabling personalized delivery of reduced amount of notifications.

1.3.3 Hypothesis

The present-day control and monitoring systems are focused on interpretation and delivering of quantitatively sensed data; however, the qualitative aspects of the context are often missing and do not participate in the context interpretation. In this work, it is believed that combination of qualitative and quantitative aspects of the context in the modelling of system data will bring the transformation from raw sensed data to meaningful information. In other words, by considering more of the context information, it will be possible to deliver less data and more essential information. Thus, the hypothesis of the work is formulated as follows:

> *By modelling parts of the world, which are related to ongoing user and system context, and fusing this model with heterogeneous runtime system data, it is possible to extract relevant information to particular user with respect to his/her responsibilities, needs, and ongoing status of the environment.*

1.3.4 Objectives

The main objective of this work is the development of middleware capable to deliver relevant information by interpreting context information and fusing this knowledge with runtime device data. The middleware must support multi-role and multi-user environments with the special focus on building management systems. In addition, the features of modern control and monitoring systems impose the following list of qualitative attributes, which must be supported by middleware: expressivity, scalability, runtime performance, and (re)configurability.

The additional objective is the development of technology-independent principles of relevant information delivery. Specifically, knowledge modelling (information to capture in the knowledge models, factors to consider), classification and combination principles of ongoing notifications, and conceptual blocks for relevant information delivery should be addressed in this work.

1.3.5 Contributions

This work provides the following list of original contributions:

- *Knowledge modelling principles for relevant information delivery*: the work covers information needs in related domains and proposes a novel knowledge model for relevant information delivery, which reflect qualitative context information (e.g. places layout, user needs and responsibilities) and assist in configuration process of the middleware.

- *Hybrid approach for relevant information delivery*: a novel technological solution for relevant information delivery is proposed. The solution features combination of semantic knowledge models for describing and uniform handling of heterogeneous data and Complex Event Processing for runtime data handling and processing. The combination of these technologies has enabled addressing the qualitative attributes required by the solution.

- *Principles of classification and combination of ongoing notifications and redirection of information flows*: the work proposes novel principles of how to deal with stream data and active notifications in the system. Specifically, it proposes classification of existing notifications, assigning proper priorities, and combining ongoing notifications at runtime without information loss.

- *Middleware tools and methods*: the above-mentioned contributions have resulted in development of a novel middleware enabling relevant information delivery in monitoring systems and addressing required qualitative attributes. The replicability of middleware is ensured by providing abstractions of required classes and step-like description of actions during implementation process.

1.4 Methodology

The development of this research follows the steps described below.

- *Survey of information needs, applicable tools and methods*: the survey is based on extensive review of information needs in control and monitoring systems serving as a basis for defining qualitative attributes for relevant information delivery. In addition, the detected information needs are verified with preliminary mock-up implementation and corresponding usability tests providing more insides on actual user needs and possible requirements for relevant information delivery. This work is followed by review of available systems and technologies evaluated as potential candidates for relevant information delivery in respect to defined qualitative attributes.

- *Development of knowledge models*: the knowledge models capturing information about entities and reflecting user information needs are developed at this stage. Moreover, early prototyping with initial implementation approach will test applicability of models and additionally will provide iterative feedback on requirements of the middleware.

- *Development of principles of runtime data analysis*: to deliver relevant information at a runtime with respect to information needs and underlying knowledge models, the principles of runtime data handling should be developed. Specifically, this includes redirection of information flows (i.e., how to define, what kind of information user sees under what conditions), formal classification of existing notifications and runtime combination principles (i.e. types and priorities of existing notifications, and how they could be handled at runtime to reduce information overload to users of the system).

- *Development of middleware for relevant information delivery*: the middleware is developed based on identified technology candidates, developed knowledge models and principles of runtime data analysis resulting in formulation of hybrid approach for relevant information delivery in control and monitoring systems.

This middleware also serves as a basis for empirical studies of this research work.

- *Empirical study*: a set of experiments with real data from rehabilitation facility is performed. These experiments cover different aspects of relevant information delivery, specifically:
    - redirection of information flows from sensors, which are delivered based on user role, needs, and responsibilities;
    - assessment of proposed combination principles of ongoing notifications (i.e. evaluating the effects of applying these principles to data from rehabilitation facility);
    - reconfiguration scenarios of the systems (e.g. adding new devices, changing user roles and responsibilities in the system);
    - extensive performance evaluation of the proposed solution (e.g., the effects of variability in number of devices and intensity of data flows).

1.5 Assumptions and Limitations of Scope

Although present-day BMS and FAS are very similar in architecture and capabilities, this work particularly focuses on needs and requirements of BMS. Moreover, the contribution is targeted at reducing information load in the ongoing context, which further limits the scope to supervisory control providing overall building view [Delamer, 2006, p. 31]. Thus, data analytics and building optimisation are outside of the scope for this research. The further delimitation comes from the fact that this research is focused on information handling, which leaves consideration of displaying devices outside of the scope. In that way, no display limitations were considered in this research; instead, the focus was put on defining and delivering of relevant information in particular context, which could be later adopted to particular displaying devices.

The solution is seen as middleware integrable into different control and monitoring systems; therefore, some of the needed features are outsourced from those systems. In particular, it is assumed that data from devices comes with no need of pre-processing and filtering. It is further assumed that middleware could rely on security measures of the monitoring system (e.g., login support, secure communication, server protection); thus, leaving security concerns outside of the primary focus of this research work.

1.6 Thesis Outline

The rest of the thesis is structured as follows. Chapter 2 provides the review of information needs, systems and technologies related to this research work. It is concluded with the literature assessment, highlighting existing gaps and further directions for this research. Chapter 3 summarises the methodology and associated results of all publications included in this work. Finally, Chapter 4 concludes the research with summary of the results, lessons learned, and directions for future work.

2 Literature and Technology Review

This chapter provides a review and assessment of systems and technologies connected to relevant information delivery with the purpose of choosing suitable technologies for the middleware. The chapter begins with evaluation and analysis of information needs in related domains resulting in formulation of qualitative attributes, which should be fulfilled by solution of relevant information delivery (section 2.1). Then, the chapter continues with section 2.2 presenting systems and architectures from various domains with properties of selective information delivery. This section analyses the focus area of each system and highlights the most suitable approaches for the BM domain. Next, section 2.3 evaluates technologies and tools of chosen system type and provides summary of how these technologies fulfil qualitative requirements of the solution. Finally, section 2.4 provides summary of the literature review and assessment of potential technologies for the required solution.

2.1 Information Needs in Related Domains

> *"Personalization is motivated by the recognition that a user has needs and that meeting them successfully is likely to lead to a successful relationship with the user"* [Gonzalez, Chen, & Dahanayake, 2008, Chapter XI].

Information needs of different operators and other system users greatly define relevant information in specific situations and form a basis for defining requirements, which should be met by the solution. These needs vary depending on the situation, time, location, collaboration with other users, ongoing tasks etc. The needs are also determined by domain of system operation and automation tasks; although in many cases, several related domains should be considered to analyse existing standards and adopt best practices.

The BM domain and factory floor monitoring systems have recently become significantly closer in structure and capabilities. While BMS have more integration of different components and increased number of functionality, the factory floor systems have decreased the number of levels in their architecture and have become more open to integrating components from different vendors [Sauter et al., 2011]. All of it became possible with wide adoption of communication standards, which led to flattening of architectures and increased similarity of these systems. Thus, the following domains of the monitoring systems are chosen for analysis of information needs: process automation, discrete manufacturing, and building management. Undoubtedly, these domains are different, but increased commonalities and well-supported standards of factory floor systems make them good candidates for comparison and derivation of requirements.

2.1.1 Process Automation

Process automation is dealing with control of continuous processes. In industry and manufacturing, examples are chemical, paper, oil, energy production etc. Outside of industry, the example of process automation system could be flight cockpits, where the crew must follow flight conditions and work together with the automation system of the cockpit. One of the main features of process automation is its low tolerance of shutting down. It is either too expensive, because of production or restarting costs, or simply impossible (e.g. one cannot shut down flight and restart it if something goes wrong). Usually, these systems automate multivariable processes and the task of the operator is to follow values of process variables and adjust automation parameters, when it is needed, to prevent system failures.

With the ICT advances and adoption of modern IT solutions, the observed trends are increased complexity of automation system and decreased number of process operators. This creates a contradiction between increased capabilities of automation systems and the need of highly skilled operators to work with it, which was noted more than thirty years ago and is known as “ironies of automation” [Bainbridge, 1983]. Most of these “ironies” are still relevant in modern process automation systems, some of which have even become more prominent in recent years.

The information needs are formed by many factors including specific features of the automation systems and functions of the system operators. One of the main features of process automation systems is their manual and automated modes of work. These modes result in challenges, such as determining right amount of workload and providing needed information during transition periods. The amount of workload should be balanced between too high, leading to overload and possible system failures, and too low leading to underload, boredom and eventual error [Seong, 2009, p. 173], [Bainbridge, 1983]. During the automated performance, when human acts as observer, the primary task is to prevent boredom, while during manual performance, when operator is in control of the system, the task is to provide all essential information, but not to overwhelm user with it. Transition periods between manual and automated modes of work create special information needs [Kaber, Riley, Tan, & Endsley, 2001], [Bainbridge, 1983]. The user must be fully aware and informed between those transitions, and transition itself should be supported with needed information (e.g. when the system goes to manual mode, the user might be interested in values of additional parameters, which were not crucial and hidden during the automated mode).

In such a way, even though the evolution of process automation systems has dramatically extended their capabilities, these systems still rely and require highly skilled and knowledgeable operators. These skills and knowledge, in turn, determine personal information needs, which depend on knowledge of the process and available options for control, skills of operator obtained through thorough training, and personal experience [Seong, 2009, p. 172]. It should be also noted that in highly automated environments, where user is mostly acts as system monitor and is out of control loop, operator might experience erosion of skills, which should be taken into consideration by additional trainings [O’Hara & Hall, 1991].

In the complex systems, which require team of the operators for proper system functioning, coordination between the team members and the automation system creates additional information needs. Task allocation, coordination, status updates, all of it should be delivered to operators helping them to accomplish their personal primary tasks.

Thus, information needs in process industries are complex and depend on many factors, including complexity of controlled process, whether the monitoring is performed by single operator or by team, skills and knowledge of the operators, manual or automated modes of work. The monitoring of the process is the primary task, so workload balance is important and special attention should be paid to transition periods between manual and automated modes. Consideration of these factors becomes even more challenging under the time pressure of abnormal situations.

2.1.1.1 Notes on Available Standards

Due to high requirements to system safety and high expenses of system failures, the process industry is well supported with HMI related standards. The examples include ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems[3], ISO 11064-5:2008 Ergonomic design of control centres - Part 5: Displays and controls[4], and Human-System Interface Design Review Guidelines (NUREG-0700, Revision 2)[5]. These standards and recommendations aim to improve design of safe, reliable systems; however, their direct application is not straightforward.

Good HMI design is a laborious process, which must consider many application specific requirements; however, HMI related standards balance between being too generic and too specific. Generic standards attempt to have wide applicability to the point, where they are hard to use for specific applications, while specific standards might have good practical tips, but be useful for only handful of users. HMI design standards are generally hard to adopt for relevant information delivery; they only sparingly contain details of the information, which should be displayed in HMIs, and are more concerned with visual representation of this information. More practical tips of what information and how should be delivered to user could be found from alarm management domain.

Alarm management consists of the set of techniques, tools, standards, and procedures, which aim to improve the effectiveness of alarm systems. The examples of alarm management standards in industries are EEMUA-191 (2007) Alarm Systems: A Guide to Design, Management, and Procurement[6] and ANSI/ISA-18.2-2016, Management of Alarm Systems for the Process Industries[7]. These standards

[3] ANSI/ISA-101.01-2015, Human Machine Interfaces for Process Automation Systems (accessed 13.05.2017): https://www.isa.org/store/ansi/isa-10101-2015,-human-machine-interfaces-for-process-automation-systems/41921133

[4] ISO 11064-5:2008 Ergonomic design of control centres - Part 5: Displays and controls (accessed 13.05.2017): https://www.iso.org/standard/44691.html

[5] Human-System Interface Design Review Guidelines (NUREG-0700, Revision 2) (accessed 13.05.2017): https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/

[6] EEMUA-191 (2007) Alarm Systems: A Guide to Design, Management, and Procurement (accessed 14.05.2017): https://www.eemua.org/Products/Publications/Print/EEMUA-Publication-191.aspx

[7] ANSI/ISA-18.2-2016, Management of Alarm Systems for the Process Industries (accessed 14.05.2017): https://www.isa.org/store/ansi/isa-182-2016,-management-of-alarm-systems-for-the-process-industries/46962105

contain good practical tips on alarm philosophy and design as well as well-defined classification between different types of notifications in monitoring systems. This classification and alarm design tips have inspired classification and management of abnormal situations for building management domain discussed in this work.

2.1.2 Discrete Manufacturing

Discrete manufacturing (or factory automation) focuses on production and assembly of distinct items, such as automobiles, smartphones, toys, etc. Factory systems automate manufacturing operations and assembly of distinct products heavily relying on multiple robots and conveyors (manufacturing lines, materials handling, assembly lines). Typically, these systems are more tolerant to shut down; usually, only costs are involved, not safety of people and personnel. However, the costs are still high, so one of the primary goals of operators in such systems is to locate and fix malfunctioning equipment as soon as possible or delegate necessary tasks to responsible personnel.

The automation and manual tasks are usually well defined, so the problem of transition and understanding situation between manual and automation modes of the system are not that critical in comparison to process automation. The malfunctioning equipment could be shut down and fixed without human operators performing manual control. Instead, quick localization of malfunction among multiple equipment, right delegation of tasks, and good support with appropriate instructions are the primary needs of automation in discrete manufacturing [Nieto Lee, Gonzalez Moctezuma, & Martinez Lastra, 2013].

Increased number of user roles in discrete manufacturing dictates information needs. While in the process automation primary role and responsibility of the operators is monitoring and adjustment of system parameters, discrete manufacturing introduces multiple roles and tasks. Operators, technicians, supervisors, all of them require different kinds of information from the workshop floor and different levels of representation. For example, the technician might be more interested to get detailed information of the robot, which he/she is currently fixing, while supervisor is more interested to have general information during downtime, such as duration and needed fixes, in order to make corrections in the production plan.

Large number of equipment and devices in manufacturing lines is another aspect formulating information needs. Numerous equipment is accompanied by multiple notifications. Often, nuisance alarms overload operators, who eventually stop paying proper attention. Notifications and alarms should be well designed and delivered to only needed personnel with informative description helping localize and understand problem. When problem is localized and delegated, further up-to-date instructions should be provided, which help quickly fix the problem.

Thus, information needs in discrete manufacturing highly depend on role and tasks of the system users. This dependency requires ways allowing high personal customization of information and HMIs. Multiplicity of equipment require careful notification design, easy problem localization, and at-hand availability of manuals and instructions during the maintenance of the equipment.

2.1.3 Building Management

Building management systems (BMS) are concerned with automation and monitoring of the environment in the commercial, industrial, or institutional facilities. The purpose of these systems is to ensure operational performance, reduce energy consumption of buildings, and increase comfort and safety of building occupants [Sauter et al., 2011]. Contemporary building management systems attempt to cross-integrate different building functions under one system in order to optimize energy consumption with respect to multiple factors inside and outside of the building, such as weather and residents activity [Nguyen & Aiello, 2013]. The future of BMS is seen in providing multiple services tailored to the needs of building occupants and facility managers. New trend in building automation is to consider not only energy consumption of the building, but also to maximize comfort of building occupants and to facilitate maintenance of a facility [Frost & Sullivan, 2011].

The automation complexity of BMS is considerably lower than in process or discrete automation; almost all the functionalities and functions could be simply automated, shutdown is more tolerable than in previous systems. The BMS do not automate complex processes or robots; however, the number and diversity of devices on the site could be impressive, which create specific challenges for information delivery.

All devices on the site produce large amounts of information. This information must be processed and be accessible to different users. However, finding needed information or locating arisen problem among multiple sources is not a trivial task [Johnson Controls, 2015]. Similarly to discrete manufacturing, poor notifications design and nuisance alarms often result in overwhelming of user capacities and improper handling of notifications. Thus, the information should be carefully processed and delivered to users in the form, which help to locate the problem.

Availability of multifarious sources of data and increased functionality of BMS, attract more user groups, who might benefit from BMS data in solving their daily tasks. Managers of the facility, maintenance personnel, nurses – all of them could find information of interest from BMS. For example, nurse in the managed facility could have access to part of the information related to patient health status and daily habits, such as presence and absence of residents at home, cooking and cleaning activities. In such a way, for most of the users, BMS is not a primary target of interest, but rather an additional source of valuable information. As a result, the problem of information underload is not relevant for BMS; more important is to see situation at one glance and to handle it as fast as possible. Hence, it is crucial to consider user requirements and provide some ways for flexible handling and delivering of the information.

Thus, availability of multiple devices and improved capabilities of BMS have opened them to various user groups. These groups and individual users differ in their roles, tasks, and objectives, which form principal information needs. Large numbers of data points and diversity of information needs require ways for processing and selective delivery of available information.

2.1.4 Qualitative Attributes

Information needs in related domains help to analyse requirements and form qualitative attributes, which should be met by solution. Table 2-1 summarizes challenges and associated information needs in considered domains.

Table 2-1: Operator challenges and information needs in related domains

Domain	Specific operator challenges	Associated information needs
Process automation	Switching between manual and automated modes of the system	Clear information about system mode and state; Information associated to each system mode; Preparation of the user between mode switch (e.g., additional information on hidden process states) Clear information on all decisions taken by the system (with possible explanations of why they have been taken)
	Underload/overload during the operating modes of the system	The balanced workload of the operator; Information associated with tasks
	Operator tasks require good knowledge of the process and excellent skills in handling it	Support (but not overwhelming) of unexperienced operators; “Refreshing” skills and knowledge of experienced operators
	Decisions under time pressure	Adjustment of amount of delivered information; Prioritizing of tasks and delivered information
	Coordination between team members	Functional division of tasks and responsibilities; Informing on important actions of all team members
	Nuisance alarms	Good alarm design and prioritization of tasks
Discrete manufacturing	Overwhelming number of notifications (errors of equipment, additional information) Nuisance alarms	Prioritizing notifications with respect to user present tasks and role in the system Easy access to all related information to locate alarm and fix it
	Difficulties to fix alarms (instructions missing or outdated, localization)	Restricting of redundant information during the problem fixes (i.e. unrelated to problem information, when user is in the shop floor fixing the problem) Availability of timely instructions
	Operator is exposed to irrelevant information without respect to his ongoing tasks	Taking into consideration primary tasks and diversity of users of the system

Domain	Specific operator challenges	Associated information needs
	Coordination between team members	Functional division of tasks and responsibilities; Informing on important actions of all team members
Building management	High number of devices under monitoring, while system monitoring is not a primary task	Distinguishing between various users of the system and delivering only needed information Providing ways to understand situation at one glance (underload is not a challenge here)
	Nuisance alarms	Prioritize, filter, combine and deliver only needed to specific user notifications
	High number of facility management tasks	Additional information, such as maintenance schedules, reminders, calendars, events, should be available for users and integrated into system
	Problems must be quickly fixed	Clear tasks and responsibilities allocation among users of the system Support in problem fixes, such as reducing information load to only relevant to problem at hand, providing additional instructions or contact information of relevant companies if needed

The considered domains have their distinct features and commonalities in information needs. The process automation deals with control of complex processes. Specific challenges of these systems are switching between manual and automated modes of work and support of the operators in highly complex automation tasks. At the same time, discrete manufacturing and building management systems are more challenged with high number of data points and greater variety of users. This variety requires ways for flexible configuration of the systems. Common to all domains are the needs in well-prioritized structured information, which help to solve ongoing issues. In relation to BM domain, this also requires adaptation of good notification and alarm design practices.

Meeting challenges and information needs in the BM domain requires certain qualitative attributes of the middleware. The presence of multiple heterogeneous devices and different users requires *expressivity* from the solution. Large number of devices in BMS needs *scalability*. Processing ongoing data and delivering important information to users requires *runtime performance*. Finally, importance of flexible reconfiguration in manufacturing systems has been noted in many works [Martinez Lastra, 2004], [Delamer, 2006]. This is also applicable in the BM domain, where large system sizes and variety of users often result in changes in the system. Thus, *(re)configurability* is an important attribute required from solution. Table 2-2 summarizes these attributes for BM domain.

Table 2-2: Qualitative attributes required from middleware

Attribute	Description
Expressivity	The capability of solution to capture characteristics of heterogeneous system entities and multifarious relationships
Scalability	The capability of solution to process data from systems of different sizes
Runtime performance	The capability to process data within short timespan
(Re)configurability	The capability of solution to easily adapt to changes in the system (e.g. new devices, users, changing access rights)

These attributes help to choose suitable technologies for building middleware for relevant information delivery in BM domain. In order to narrow the scope of suitable technologies, systems with property of selective information delivery are discussed in the next subsection. The purpose of this subsection is to choose appropriate system type for building management domain, and then to go deeper into available technologies for this type.

2.2 Architectures and Systems

The scope of this section is to review systems and architectures that have a goal of selecting information, services, actions, etc. out of many available options. The aim is to analyse general mechanisms used in these systems and application areas in order to choose BM-suitable type for deep analysis of available technologies.

2.2.1 Expert Systems

An expert system emulates decision-making ability of a human expert. The first expert systems have sprang in early 70s from the attempt of the AI (Artificial Intelligence) community to improve problem-solving mechanisms of existing solutions by incorporating domain-specific background knowledge of the experts [Russell & Norvig, 2010, p. 22]. Usually, these systems accept complex queries and process them over large knowledge bases to find suitable options or to infer appropriate answers.

The most important part of any expert system is the knowledge base. First, this base is supposed to be constructed and rigorously evaluated in collaboration with experts of the field [Josefiok, Krahn, & Sauer, 2015]. The next important step is the choice of the implementation approach. There are different ways of knowledge representation with the most popular being Bayesian networks [Constantinou, Fenton, Marsh, & Radlinski, 2016], fuzzy logic [Skoczylas, 2014], and rule-based approaches [Yang, Fu, Chen, Xu, & Yang, 2016]. Each one has advantages and limitations, which should be considered for specific applications. For example, Bayesian networks are well suited for expressing probabilistic knowledge and they can be used on available data for training of the network. Rule based approaches are better suited for complex problems, where causality model of Bayesian networks is hard to understand [Onisko, Lucas, & Druzdzel, 2001]. However, large number of rules becomes quickly unmanageable, which restricts their usage.

The combination of reasoning mechanisms with formal knowledge representation have been applied successfully in many fields including medical diagnosis, natural language processing, industry and military applications. Although application areas are quite wide, the broad knowledge bases require high computational power, thus making application to runtime and large-scale environments quite challenging. Still, the expert systems field is origin to many powerful knowledge representation and reasoning techniques, such as the first order logic and ontologies, which are used in other systems and will be discussed later in this work.

2.2.2 Recommender Systems

The recommender systems (RS) are a subclass of information filtering systems that provide suggestions for various items to users [Ricci, Rokach, & Shapira, 2011]. These systems help users to deal with information overload and suggest personalized recommendations among available options. They have been an active area of research starting from the mid-90s with development of the web and large number of databases.

The type of the filtering algorithm usually defines the type of the RS. There are four most widely used filtering algorithms: content-based filtering, collaborative filtering, demographic filtering, and hybrid filtering [Bobadilla, Ortega, Hernando, & Gutiérrez, 2013], [Adomavicius & Tuzhilin, 2005]. Content-based filtering considers choices of the user made in the past; it also considers the content of the chosen objects to find similar objects in the database. Collaborative filtering makes suggestions based on similarity between ratings provided by users; these ratings can also be collected implicitly by analysing activity of users and their interaction with content of a website. The demographic filtering provides recommendations based on the commonalities of user profiles, e.g. gender, interests, age. Hybrid filtering often uses different combination of these three algorithms as well as various probabilistic methods. The choice of the algorithm depends on available data, purpose of filtering and application scenario. For example, collaborative filtering shows good results in providing original recommendations; however, it is challenged with scalability issues for large communities. In order to overcome this challenge, collaborative filtering could be combined with demographic algorithm, pre-clustering population and improving scalability of the solution [Gupta & Gadge, 2015].

The RS have plenty use cases including search of media content, advertisement, social networks. The recent advancement is consideration of user's ongoing context [Adomavicius & Tuzhilin, 2015], such as location [Symeonidis, Ntempos, & Manolopoulos, 2014], [Wu, Kao, Wu, & Huang, 2015], time [Joho, Jatowt, & Blanco, 2015], and mood [Kranjc et al., 2015]. The principal difference between recommender and expert systems is the ability of expert systems to infer new knowledge. Being subclass of information filtering systems, recommender systems operate over existing information without changing it. While this is not the problem for pure RS use cases, this might be a challenge for relevant information delivery, where available information must be processed and sometimes modified for personalized delivery. Thus, the recommender systems are a powerful tool for reducing information load to users, when searched content is well defined and described.

2.2.3 Situation-Aware Systems

Another aspect of information overload problem is how humans percept information. It is important not only to detect what is happening in the environment, but also to present it to users in the most convenient way, so users might feel informed rather than overwhelmed. In order to achieve this objective, systems must understand how humans percept and process information. The concept of situation awareness (SA) from the field of human factors tries to address these questions and gives the internalized mental model of operators' environment (Figure 2-1).

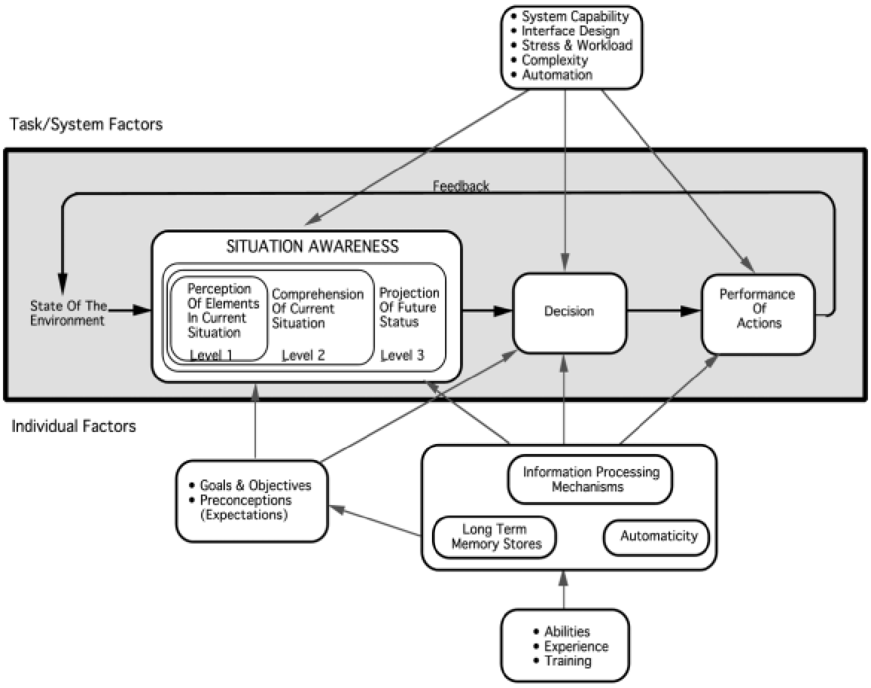


Figure 2-1: Model of SA in decision making [Endsley, 1995]

The model attempts to depict internal decision-making processes of the operators, their relationships, and factors affecting decision-making. As it can be seen from Figure 2-1 in the "Task/System Factors" part, the system design and functionality affect operators' situation awareness, decision and performance of actions. Being greatly affected by system capabilities, the three-level situation awareness is the most discussed in the research community.

The three levels of SA reflect perception, comprehension and projection. The level 1 awareness is achieved, if operator is able to perceive critical factors of the environment. When operator fully understands meaning of these factors in relation to his goals, level 2 of comprehension is achieved. Finally, level 3 of SA corresponds to operators' ability to predict, what will happen next [J. D. Lee & Kirlik, 2013, p. 89]. The systems are considered to have the highest degree of situation awareness, when they successfully address all three levels of situation awareness. In order to achieve this, the research community has presented the number of works addressing the ways to model situation awareness in the systems [Kolbe, Zaslavsky, Kubler, Robert,

& Le Traon, 2017], [Feng, Teng, & Tan, 2009], [Kokar, Matheus, & Baclawski, 2009], to design SA systems [Luokkala & Virrantaus, 2014], and to evaluate the degree of SA demonstrated by systems [Salmon et al., 2009].

The SA model has been widely discussed and as a result has certain degree of a criticism. The most prominent critical points are linearity of the model, lack of the verification, and closed nature of the model to external world. The author of SA model addresses these critical points in the recent paper and further clarifies possible misconceptions and misunderstandings [Endsley, 2015]. There are some alternative models of the situation awareness proposed in the literature and addressing certain issues of Endsley's model [Bainbridge, 1997], [Chiappe, Rorie, Morgan, & Vu, 2014]; however, they did not gain proper attention and Endsley's model is still the most discussed model in the research community.

The SA research has appeared in avionics domain, but quickly it has been recognized in other complex systems including military, emergency, driving, and industry applications [J. D. Lee & Kirlik, 2013, p. 89], [Luokkala & Virrantaus, 2014], [Kolbe et al., 2017]. These use cases demonstrate importance of humans in the system operation, and lead to understanding of how essential it is for the system to be aware of the human operator.

2.2.4 Context-Aware Systems

Although proper modelling of human perception is important in supporting human operators, the SA theory is mostly focused on its internal processes and pays less attention to environmental information and its relevancy. The field of research which utilizes explicit and implicit information of the environment (including users of the environment) in order to deliver personalized information, services and actions is called context-aware computing; and the systems featuring context-aware computing are called context-aware systems [Dey, 2001], [Perera, Zaslavsky, Christen, & Georgakopoulos, 2014].

2.2.4.1 Definition of Context

Context-aware systems have sprung from pervasive and ubiquitous computing, when computational power of embedded devices have allowed them to process more information and be "interested" in surroundings. Even though the terms "context" and "context-awareness" have appeared in literature starting from mid-90s, there is still an ambiguity and sometimes even contra versions of their meaning. The mainstream definition of context and context-awareness was suggested by Dey:

> *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to*

> *the interaction between a user and an application, including the user and applications themselves."* [Dey, 2001]
>
> *"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."* [Dey, 2001]

2.2.4.2 Context Categorisation

Context is defined as any information that is related to interaction between user and application/system. While it allows freedom in choosing arbitrary information, in many cases it is more convenient to categorise it by some characteristics for unified handling of similar information. This context categorisation is another area, which does not have standard approach. Naturally, it is difficult, if not impossible, to divide context into categories, which would suffice to any context-aware application. Thus, there are many approaches to grouping of context information summarized in several works [Soylu, De Causmaecker, & Desmet, 2009] [Perera et al., 2014].

One of the approaches is to categorize context by the need of additional computation; i.e. if the value is used as it is, this information might be called low-level context [Soylu et al., 2009] or primary [Perera et al., 2014], and computed information is called high-level context or secondary. Another approach is to categorize context by its dynamicity [Soylu et al., 2009]. This approach might be useful in some application for validation of context data or providing historical context. Although these approaches have their place in applications, they are quite limited in expressing varieties of context information and are usually combined with conceptual context categorisations. These concept categories could be around the person, his/her tasks and objectives, space, location or combinations of any of them [Perera et al., 2014], [Nadoveza & Kiritsis, 2014], [Rodriguez, Bravo, & Guzman, 2012]. One of the classical categorizations, suggested by Dey, Abowd, & Salber [2001, p. 107], includes identity, location, activity, and time as essential characteristics of any context. Still, categorization and handling of context information is greatly defined by objectives of specific applications.

2.2.4.3 Fundamental Blocks of Context-Aware Systems

The accepted definitions of context and context-aware systems are quite broad and allow numerous interpretations, approaches, and implementations of context-aware systems. For this reason, it is challenging to classify these systems and find common grounds on technological level. However, context-aware systems are quite similar on conceptual level, and literature analysis has allowed formulation of conceptual framework based on [Baldauf, Dustdar, & Rosenberg, 2007], [Alegre, Augusto, & Clark, 2016], [Schmidt, 2014] works, which is demonstrated in Figure 2-2.

The framework consists of five layers. The first layer represents data sources; this includes not only physical and logical sensors, but also any other useful information sources, which might provide better understanding of context (e.g. calendars, weather forecast, information history). Second layer represents the need of data pre-processing. This need appears, when there is a problem of data source quality or the data from sensors is specific and requires processing before its context interpretation (e.g. ECG sensor). This layer is also responsible for gathering data from virtual or logical sensors through middleware [Perera et al., 2014].

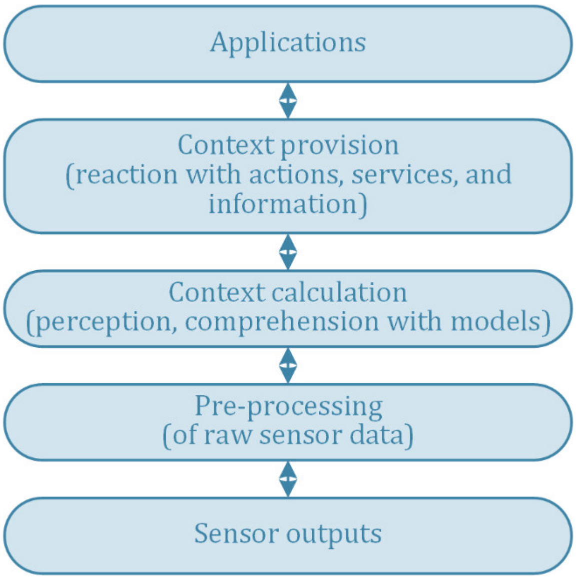


Figure 2-2: Conceptual framework of context-aware systems

Next layer is context calculation, which perceives and comprehends gathered information. This layer is represented in some form in any context-aware application and it usually involves some forms of context models. These models reflect the knowledge of the particular system about surrounding context. The modelling of context could be accomplished with various techniques including key-value, markup schemes, object-oriented, and ontology models [Alegre et al., 2016], [Bettini et al., 2010], [Baldauf et al., 2007], [Strang & Linnhoff-Popien, 2004]. Each of the modelling techniques has their advantages and disadvantages; however, ontological models are recognized as the most expressive knowledge modelling techniques, which also are the most adopted in the literature.

The context provision layer is responsible for formulation of possible actions and/or delivery of personalized services and information based on the sensed context. In implementations, this layer is represented by reasoning mechanisms of the systems, which interpret sensed context and might create new pieces of information and/or knowledge. This layer also have plenty of techniques including supervised and unsupervised learning, fuzzy logic, probabilistic and semantic reasoning, various rule-based mechanisms [J. H. Lee, Lee, Kim, Wang, & Love, 2014], [Alegre et al., 2016], [Perera et al., 2014]. Often, system implementation blurs the border between context calculation and context provision layers. For example, context knowledge bases could be in the form of rules, which activate in certain conditions [J. H. Lee et al., 2014]. Also, ontological models already assume some reasoning mechanisms, which might be further extended with e.g. SWRL rules.

Finally, application conceptual layer represents a consumer of the information and services. Again, this could be a separate application, receiving data from context-aware server, or device/application itself, which react to ongoing context.

2.2.4.4 Application Domains

With generally appealing description of reacting to ongoing context, context-aware systems are extremely popular in many domains including different kinds of smart environments [Han, Lee, & Crespi, 2014], [Fenza, Furno, & Loia, 2012], industry [Uddin, Puttonen, Scholze, Dvoryanchikova, & Martinez Lastra, 2012], [Khilwani, Harding, & Choudhary, 2009], logistics [Meissen, Pfennigschmidt, Voisard, & Wahnfried, 2005], driving [Baumgartner, Gottesheim, Mitsch, Retschitzegger, & Schwinger, 2010], [Terroso-Saenz, Valdes-Vela, Campuzano, Botia, & Skarmeta-Gomez, 2015], [Sun, Wu, & Pan, 2009]. Even though (or because of) applications are numerous, context-aware systems are challenging for classification of different technological approaches and could be unified only at conceptual level. In relation to relevant information delivery in BM domain, context-aware systems promise appealing properties of a solution to be sensitive to ongoing environment and specific requirements of users, thus making it the most suitable approach so far for consideration of available technologies.

2.2.5 Cyber-Physical Systems

The increase of computational power and communication capabilities of embedded devices has also led to recognition of their potential to become self-aware and self-organized systems sensing, monitoring and controlling physical environment. This vision presents new research challenges in relation to communication, virtual modelling, and adequate acting of such devices under ongoing environment conditions. The research area addressing these challenges for systems represented in physical and cyber world has united under the name of cyber-physical systems (CPS).

The CPS research is relatively new interdisciplinary technological field, which still establishes structure and technology of its systems. The CPS view has gained attention in several domains, among which industry is the closest to BM domain. Industrial view on CPS, sometimes called Cyber-Physical Production Systems (CPPS) or Industrial Cyber-Physical Systems (ICPS), is based on Industry 4.0 vision of decentralized quickly reconfigurable entities with self-aware capabilities [Almada-Lobo, 2017]. The recent effort was put to establish unified architecture of such systems capable of addressing all related requirements and needs (Figure 2-3).

The five-level architecture to some degree correspond to standard automation pyramid with additional functionalities and capabilities. The level 1 corresponds to the sensor level with the “smart connection” referring to the possibility of easily connecting of multifarious sensors and devices. Level 2 transforms data to information on machine level, thus bringing self-awareness to machines. Level 3 represents cyber level of the systems containing virtual representations of machines and connecting them into communicating network in the cyber space. This network of machines is supposed to have analytical capabilities enabling better status analysis for each machine and self-comparison abilities. Level 4 further elaborate analytical capabilities of the CPS building overall knowledge of the system and supporting user with proper representations and decision-making capabilities. Finally, level 5 is seen as a feedback provider from cyber space to physical space, which also provides corrective actions based on the decisions of cognition level [J. Lee et al., 2015].

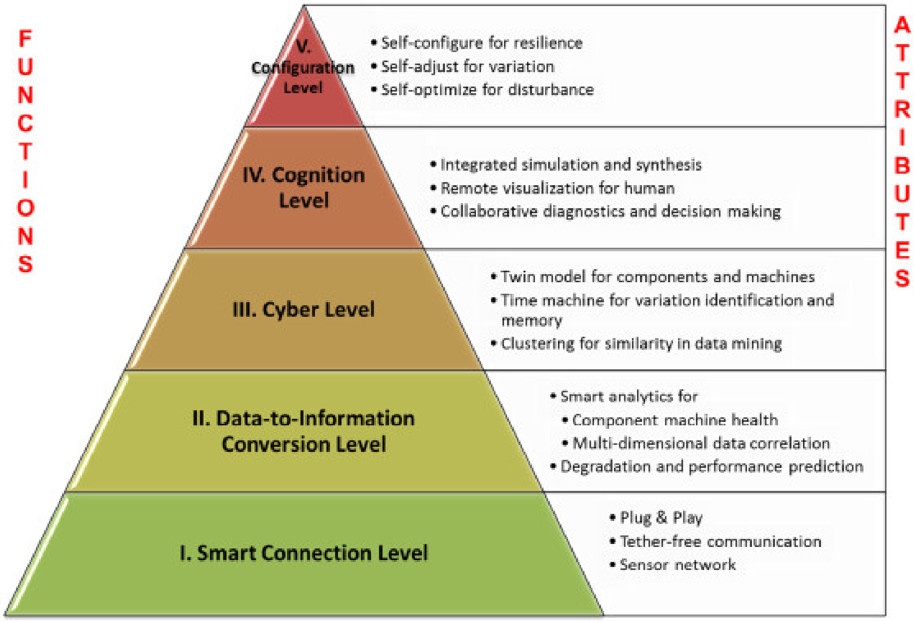

FUNCTIONS
ATTRIBUTES
V. Configuration Level
• Self-configure for resilience
• Self-adjust for variation
• Self-optimize for disturbance
IV. Cognition Level
• Integrated simulation and synthesis
• Remote visualization for human
• Collaborative diagnostics and decision making
III. Cyber Level
• Twin model for components and machines
• Time machine for variation identification and memory
• Clustering for similarity in data mining
II. Data-to-Information Conversion Level
• Smart analytics for
• Component machine health
• Multi-dimensional data correlation
• Degradation and performance prediction
I. Smart Connection Level
• Plug & Play
• Tether-free communication
• Sensor network

Figure 2-3: Architecture for implementation of CPS [J. Lee, Bagheri, & Kao, 2015]

To some extent, the proposed CPS architecture is similar to conceptual framework of context-aware systems especially in having knowledge representation and reasoning capabilities. However, context-aware systems are more focused on system-user interactions in the ongoing context, while CPS have primary goal of achieving self-awareness and self-control capabilities of machines in the ongoing conditions of the environment through cyber space. This focus on "self-" aspect of the machines brings specific research challenges including modelling and analysis of multifarious system devices, distributed sensing, control, and computation, and, finally, seamless integration of physical and cyber subsystems [Ray, 2013].

The CPS is multidisciplinary area, which can adopt technologies from many related fields. Being naturally decentralized and distributed, the key CPS enabling technologies are seen in Multi-Agent Systems (MAS), Service Oriented Architectures (SOA), and cloud computing [Leitao, Colombo, & Karnouskos, 2016], [Colombo et al., 2014, p. 2]. These fields provide technological grounds for achieving distributed intelligence (MAS achievements), interoperability between multiple devices and systems (SOA achievements), and coping with large scale of such systems (cloud-based computing technologies). Also, the cyber and analytical part of CPS may adopt available models and technologies from control theory and expert systems [Ray, 2013]. All in all, the CPS research is a new field with multiple research questions; however, it has good starting points from the range of available technologies and systems.

The potential application areas of CPS are numerous; they include large-scale autonomous manufacturing, intelligent sensor networks for monitoring and controlling various spaces, smart buildings and cities, advanced robotic applications [Ray, 2013]. In relation to relevant information delivery, CPS are more focused on devices and their self-communication and management rather than on users of the system and their information needs. Thus, even though CPS aim to have cyber part representing physical world, this model would be more device and process oriented, in contrast to user and context orientation required by relevant information delivery.

2.2.6 Summary of Architectures and Systems

Considered system types deal with large amounts of data and select/infer information, services, and actions using various approaches and mechanisms. The key summary characteristics of these systems could be found in Table 2-3.

Table 2-3: Summary of considered system types

System	Data source	Response time requirements	Initial purpose	Core properties	Applications
Expert systems	Databases, expert knowledge	Not critical (varies)	Resolving complex tasks requiring expert knowledge	Exploiting and formalizing knowledge of experts	Diagnostic, medical, search
Recommender systems	Databases, user profiles, sometimes location information	Not critical (varies)	Selecting right information for right users	Predictions of preferences and ratings using various filtering algorithms	Library search, movie search, delivery of services, context sensitive advertisement,
Situation-aware systems	Rich sensory input, knowledge models	Near real time	Improving awareness of system operators	Modelling human cognitive processes	Military, avionics, emergency, time critical decision making
Context-aware systems	Rich sensory input, knowledge models	Near real time	Adaptation of behaviour, services, information according to environment	Environment modelling and reasoning	Manufacturing, smart spaces, service delivery, avionics, emergency
Cyber-physical systems	Rich sensory input, knowledge models	Near real time	Improved interactions of physical objects in cyberspace	Virtual representation (models) of physical devices	Manufacturing, smart buildings and cities, robotics

According to summary table, expert and recommender systems provide the ways for selecting needed information and services; however, they usually operate over large close-to-static content and usually have milder response time requirements in comparison to BM domain. For these reasons, expert and recommender systems are not further considered in this work.

On the other hand, situation-aware, context-aware, and cyber-physical systems operate over similar environment conditions and use cases, and usually have strict response time requirements. Moreover, all these systems utilize in some form knowledge modelling and reasoning mechanisms. The difference appears in the focus area of each system, and subsequent qualitative requirements. If we consider some general control and monitoring system operating within environment, and having general blocks, such as sensing and actuating mechanisms, logic, and interfaces, then the focus area of each system could be schematically shown as in Figure 2-4. In this figure, the blue boxes represent the conceptual general blocks of control and monitoring systems, while the red boxes schematically show the focus area of considered system types with respect to the conceptual blocks of the systems.

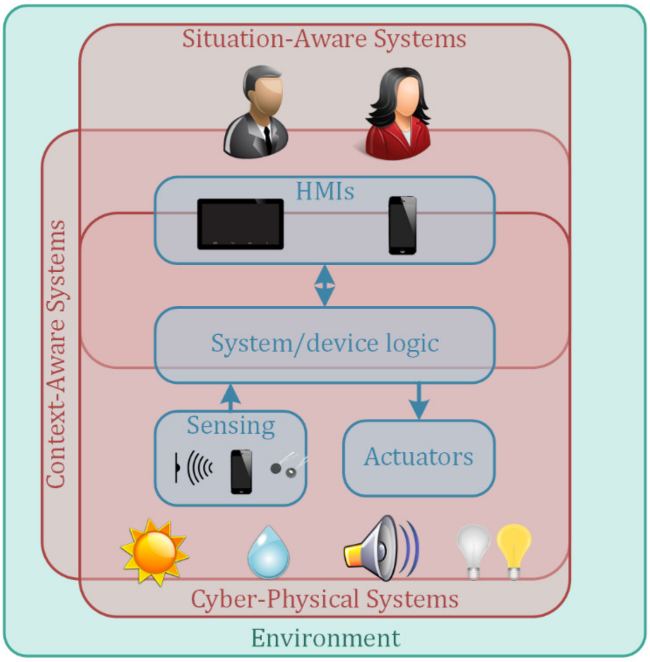


Figure 2-4: Focus area of considered system types

As it can be concluded from Figure 2-4 and previous discussions, the situation-aware systems are more focused on modelling operator perception processes and are not concerned in ways of environmental sensing and inferencing new information about environment. These systems deal with available information and attempt to present it to operators in the most convenient for their perception way. While it is a right direction for improving operators' responsiveness to ongoing displayed information, the relevant information delivery needs more attention to environmental data to locate important ongoing information pieces.

On the other hand, the focus of cyber-physical systems is put on devices and their communication and self-management capabilities with the environment. As a result, at present research community primarily addresses distributed nature of such systems, leaving information needs of the operators slightly aside.

Finally, context-aware systems seem to have the most suitable focus for relevant information delivery. These systems consider ongoing status of the environment and users as part of the environment in order to meet their personalized needs. For these reasons, it was decided to perform deep technological analysis of context-aware systems, presented in the next section, and locate technologies meeting qualitative attributes discussed in section 2.1.4.

2.3 Techniques and Technologies of Context-Aware Systems

As it was mentioned earlier, context-aware systems are widely discussed in the literature and have a lot of different approaches and implementations. Moreover, techniques and technologies used in these systems often come from various fields, thus making many technological intersections with different system types. In general, these techniques could be divided in three parts: data-driven, knowledge-driven, and event-driven. These parts vary in their usage across different levels of conceptual context-aware framework, which is depicted in Figure 2-5.

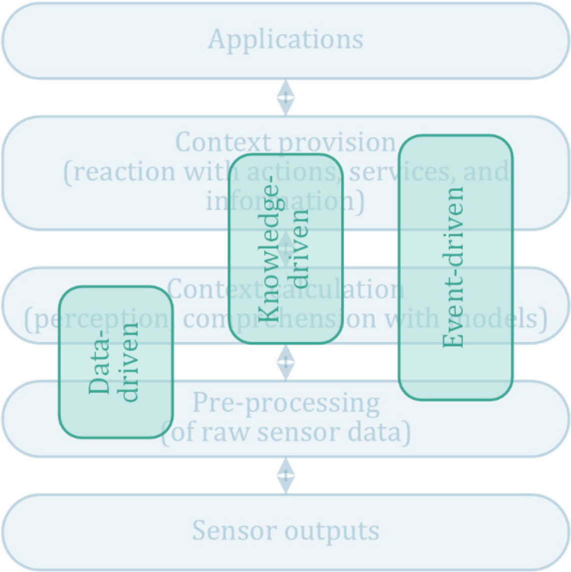


Figure 2-5: General distribution of techniques and technologies across context-aware framework

The difference in usage comes from the difference in strong features of various technologies. Thus, the pre-processing level is mostly dominated by data-driven methods, which provide powerful data mining techniques for dealing with large datasets and finding meaningful information there. Often, these methods are used as intermediate step for preliminary context inference (such as user activities), which later are combined with other methods. Context calculation and context provision are mostly occupied by knowledge-driven and event-driven methods. Knowledge-driven methods are strong in providing ways for expressive modelling of surrounding context, while event-driven methods and technologies provide capabilities to react to ongoing changes in the environment. The next subsections discuss each approach in detail.

2.3.1 Data-Driven

Data-driven methods utilize data mining and machine learning techniques for creating prediction or classification models on available large datasets. In context-aware systems, data-driven methods are actively used for detection and prediction of human activities, detecting usage patterns for different devices, and for clustering of any massive data (e.g. clustering nearby sensors in large-scale systems). In general, these methods could be divided on supervised and unsupervised learning.

2.3.1.1 Supervised Learning

Supervised learning aims to obtain prediction or clustering model with usage of labelled training dataset [Russell & Norvig, 2010, p. 695]. In application to the streaming device data, the typical steps in the supervise learning include data segmentation, feature extraction, and application of classification algorithms. Each of the steps has different options to choose from, and the research community has been involved in testing these options and proposing new ways of handling supervised learning process.

The first step is the stream data segmentation, which has the purpose of segmenting the continuous stream of events into chunks containing meaningful information (e.g. some activity in case of activity recognition). The most common approaches of segmenting data streams are fixed time windows (even time intervals) or event windows (even number of events in each window). However, these approaches require careful consideration of the window intervals, and often, are not capable of addressing requirements of the systems (e.g., activities could be drastically different in time intervals). The recent trend is to propose dynamic sensor segmentation, which allows different window sizes. One way is to divide the event stream into non-overlapping windows of different sizes. For this purpose Okeyo, Chen, Wang, & Sterritt [2014] propose to have ontological representation of expected activities and shrink or expand windows depending on the type of detected activity and its expected duration. Another approach, proposed in [Wan, O'Grady, & O'Hare, 2015] and [Ye, Stevenson, & Dobson, 2015], attempts to deal with concurrent activities and split event stream into overlapping windows of different sizes. They use the notion of semantic similarity between events, thus transforming the task of concurrent activity recognition into sequential single activity recognition.

After the sensor segmentation, the next step is feature selection. Features are the set of distinct characteristics, which allow distinguishing one outcome from the other. They are usually a combination of statistical, time frequency, and heuristic features [Chernbumroong, Cang, Atkins, & Yu, 2013]. Too large set of features require high computational power without necessarily bringing high accuracy of classification; too small feature set risks to dismiss important distinguishing features, which would result in low prediction accuracy. Banos, Damas, Pomares, Prieto, & Rojas [2012] have proposed the way of ranking features. The ranking procedure consists of calculating the discriminative factor of each feature (i.e. how well it can distinguish different classification classes) and feature robustness (i.e. how discriminative this feature with different sensors). With the combination of this feature selection procedure and Support Vector Machine (SVM) classification, they were able to achieve the accuracy of activity recognition of above 95% in semi-naturalistic sensor data from wearable devices.

Finally, segmented windows with calculated features are used with different classification learning algorithms to obtain classification models. The learning methods are numerous, they include neural networks [Lotfi, Langensiepen, Mahmoud, & Akhlaghinia, 2012], SVM [Chernbumroong et al., 2013], [Jae, Boreom, & Kwang, 2011], Bayesian networks [Nazerfard & Cook, 2015], hidden Markov models (HMM). Throughout the literature, HMM seems to be the most popular technique for activity recognition of single user sequential activities with reported accuracy of above 83% [Okeyo, Chen, Wang, et al., 2014]. However, it was reported that for activity predictions, the more accurate models are usually variations of Bayesian networks [Wan et al., 2015], [Nazerfard & Cook, 2015].

Overall, supervised learning methods are quite successful in classifying and predicting activities, especially for non-trivial datasets of wearable devices. However, these methods require large personalized datasets for learning, which are later non-generalizable to other persons, thus making application of these methods challengeable in multi-person systems. Moreover, the methods are hardly applicable in changing environments (e.g. for new devices and users in the system). Roggen et al. [2013] have addressed this problem by proposing opportunistic framework capable to deal with changing conditions in the system, however the general practices are not accepted yet.

2.3.1.2 Unsupervised Learning

Unsupervised learning algorithms inference hidden structure of datasets from unlabelled data. In context-aware systems applied in BM domain, the unsupervised learning is usually represented by Association Rule Mining (ARM). The ARM is a rule-based method for discovering hidden relations between variables in a dataset. It usually consists of two phases: (1) finding all frequent items with the certain level of support (i.e., frequency of meeting specified items across all dataset), (2) from the frequent items, generating rules with the certain level of confidence (i.e., the confirmation of existing dependency between chosen items).

Usually, rule mining techniques try to find dependencies based on temporal relationships between variables. They are actively used on large datasets of BMS. For example, Lopera Gonzalez & Amft [2016] have used temporal rule mining for device localization problem. They have extracted rules based on the time sequence events between different sensors, which allowed them to cluster BM devices based on their actual location. Another approach, proposed by Miller, Nagy, & Schlueter [2015], analyses outliers (or infrequent patterns) of mined activities in large BMS with the purpose of discovering unexpected energy usage, and thus proposing the ways to reduce energy consumption. In their work, Miller et al. were able to infer malfunctioning parts in BMS and propose corresponding changes. An interesting way of using rule mining was proposed by Guillame-Bert & Crowley [2012]. In this work, authors have worked with labelled dataset and proposed algorithm for deriving dependency between sensor triggers and labelled activities. The algorithm was able to explain more events than any of the presented competitors (i.e. show higher levels of rule confidence).

Thus, the ARM algorithm application to BMS is an active area of research. Some of the works even propose the general framework for applying these methods in BMS [Fan, Xiao, & Yan, 2015], [Xiao & Fan, 2014]. Potentially, ARM algorithms promise powerful tool for discovering hidden patterns in large volumes of BM data. However, up to now, the discovered patterns are quite simple in nature (e.g. one sensor trigger is followed by another); moreover, this method requires careful setting and processing of results, otherwise the resulting rule set might become large and unmanageable [Yu, Haghighat, Fung, Morofsky, & Yoshino, 2011].

2.3.2 Knowledge-Driven

Knowledge-driven methods use a top-down approach to information modelling and inferencing. These methods capture the knowledge of a domain in different types of knowledge models and use them to provide reasoning over available data. In context-aware systems, these methods are actively used for context modelling. The resulting models are usually intuitive to human users, which simplifies their adoption and usage for wider range of the applications. Two types of the knowledge-driven methods are especially popular in context-aware systems: Semantic Web (SW) technologies and Answer Set Programming (ASP).

2.3.2.1 Answer Set Programming

Answer Set Programming (ASP) is a well-established paradigm of declarative programming originated from the fields of logic programming and non-monotonic reasoning [Steffen & Schweizer, 2014]. The core idea of the ASP is the logical representation of the problem in the form of rules, such that answer sets of those rules are the answers of the problem. The answer set search is performed with specially designed solvers.

The power of ASP programs lies in ability to perform non-monotonic reasoning (i.e. the addition of new knowledge may lead to the revision of previously drawn conclusions), thus, enabling to deal with incomplete information [Brewka, Eiter, & Truszczyński, 2011]. In addition, in many search problems, it might be quite appealing to express the problem in a logic way (i.e. all search restrictions and conditions) and get the answer of the problem, rather than dealing with programming of the solution. Therefore, the ASP is quite popular in applications such as graph route planning programs (e.g., finding the shortest paths, Hamiltonian paths, or solving travelling salesmen problem) or other difficult search problems [Steffen & Schweizer, 2014]. The major enabler of growing ASP popularity is the advancement of ASP solvers. There are quite many research groups dealing with ASP solver development, which even led to annual ASP solver competition showing promising results and new application areas [Steffen & Schweizer, 2014].

ASP is quite appealing solution for expressing multiple nested constraints of the problem, such as multiple context situations and conditions. Thus, ASP has slowly started to gain attention in context-aware systems and industry. For example, Grasso, Iiritano, Lio, Ricca, & Scalise [2010] have used ASP for the problem of team building in large seaport. The multiple employees of the company should be daily arranged in the teams based on the different constraints, such as requirement of multiple skills and fair workload. The paper gave example of the core program with four main constrains expressed as rules. The output of the program gives teams of employees satisfying constraints. The performance of the program greatly depended on the timespan for which the team plans were build. Specifically, it took 25.3 seconds to calculate daily shift plan and 490.6 seconds for monthly shifts for 130 employers of the seaport. Another industrial example of ASP usage was proposed by Alirezaie & Loutfi [2015]. In this work, the authors have proposed the usage of ASP for explaining abnormal values of multiple sensors in gas industry, thus helping to reduce information load for operators. The authors have used Semantic Sensor Network (SSN[8]) ontology for describing sensor data and converting it into facts of the ASP program. The ASP rules were describing multiple reasons for sensor values. Thus, when certain set of sensors was fulfilling conditions, it could be interpreted as explanation of sensor values.

In summary, ASP is a promising paradigm of solving search problems with multiple constraints, which allows focus on the problem rather than requiring programming of the solution. This property of expressing constraints and automatic search for answers satisfying constraints is appealing especially in the field of context-aware systems with numerous devices and context situations. However, the main present restricting factor in application of ASP to context-aware and industrial systems is the performance of modern solvers. Even though they considerably advanced in the recent years, still the execution times could reach hundreds of seconds (20-200 depending on solver used) for one hundred of solved instances [Steffen & Schweizer, 2014], which greatly complicates their usage for streaming data.

2.3.2.2 Semantic Web Technologies

Semantic Web technologies are the family of W3C recommendation standards for describing and relating data on the Web and inside enterprises. Among these standards, the most popular are OWL[9] ontologies and SPARQL[10] queries. Although SW technologies were originally developed for describing and relating static information on the web, it is possible to use them within dynamic systems with additional technologies, such as SPARQL/Update[11] (allowing ontologies to reflect present state of the environment with dynamic updates) and SWRL[12] rules (for expressing conditional rules within OWL languages).

Ontology is the knowledge representation technique, which describes concepts and their relationships. OWL ontologies define a wide range of the possible properties for concepts and relationships. Essentially, OWL ontologies are an extension of RDF[13] and RDFS[14] triples of subjects, predicates, and objects. All these standards are

[8] Semantic Sensor Network (SSN) Ontology (accessed 27.09.2017): https://www.w3.org/2005/Incubator/ssn/ssnx/ssn

[9] Web Ontology Language (OWL) (accessed 26.09.2017): http://www.w3.org/2004/OWL/

[10] SPARQL Protocol and RDF Query Language (accessed 26.09.2017): http://www.w3.org/wiki/SPARQL

[11] SPARQL 1.1 Update, update language for RDF graphs (accessed 26.09.2017): http://www.w3.org/TR/sparql11-update/

[12] Semantic Web Rule Language (accessed 26.09.2017): http://www.w3.org/Submission/SWRL/

[13] Resource Description Framework (RDF) (accessed 26.09.2017): https://www.w3.org/2001/sw/wiki/RDF

[14] RDF Schema (RDFS) (accessed 26.09.2017): https://www.w3.org/2001/sw/wiki/RDFS

the part of the W3C's Semantic Web technology stack[15]. The extension part is directed to adding more expressivity to the knowledge modelling techniques, which might be used for reasoning and checking consistency. As a result, many researchers report ontologies to be the most expressive technique among key-value, markup, graphical, object-oriented and ontological models for capturing context information in context-aware applications [Hoareau & Satoh, 2009], [Bolchini, Curino, Quintarelli, Schreiber, & Tanca, 2007].

Semantics in Context-Aware Applications

Semantic Web technologies are applied to context-aware systems in many application domains including home health monitoring [Esposito et al., 2008], industrial applications [Xue, Chang, & Liu, 2014], [Uddin et al., 2012], road traffic management [Baumgartner et al., 2010], and delivering personalized services [Ruta, Scioscia, Loseto, & Di Sciascio, 2014], [Rodriguez et al., 2012], [Furno & Zimeo, 2014], [Smith & Bianchini, 2014], [Puttonen, Lobov, Cavia Soto, & Martinez Lastra, 2015], [Puttonen, Lobov, Cavia Soto, & Martinez Lastra, 2010] to name a few. In general, usage of semantics in context-aware applications could be associated with semantic service descriptions in SOA-like solutions or context information modelling in other distributed systems.

The Service Oriented Architectures (SOA) are the software design principles based on loosely coupled components encapsulated into services and communicating over various media. The communication model of these components enables flexible coupling, reusability, encapsulation, and interoperability aspects. The communication model generally consists of service repository, service consumers, and service providers. The service repository keeps information about services of the platform and enables discovery or search of the services for the clients (or service consumers). This could be a standalone application or mechanism incorporated into services. Service providers register themselves into service repository and respond to the clients, while clients discover available services through the service repository and invoke them through agreed mechanisms. When SOA-like solutions bring context-aware aspect into their implementations, it usually results in bringing more awareness of the context to services.

Bringing more awareness of context into the services by means of semantics usually involves two things: (1) extending semantic descriptions of the services, and (2) developing new approaches to service composition using extended service semantics [Nacer & Aissani, 2014]. There are numerous research works addressing these challenges. For example, Mayer, Verborgh, Kovatsch, & Mattern [2016] are proposing to improve description of functionality for RESTful (Representation State Transfer) services with extended RESTdesc specification. They belief, that this improved description of functionality would enable self-reconfiguration of manufacturing lines at a runtime. The Urbieta, Conzalez-Beltran, Ben Mokhtar, Anwar Hossain, & Capra [2017] have proposed the wEASLE service description model for context-aware applications. The wEASLE is abstract model with the descriptions implemented in OWL. The descriptions include semantic signature

[15] Semantic Web: Linked Data on the Web (accessed 26.09.2017): https://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24)

(names and types of service inputs and outputs), context-aware behaviour specification (pre-conditions, post-conditions, and effect descriptions of actions), and conversation (the possible order, in which capabilities of the service might be executed). The purpose is to utilize this model at the time of service composition tasks in order to meet needs and goals of user in context-aware assisted living environments. Another OWL-like approach for service descriptions is proposed by Furno & Zimeo [2014]. The approach is based on extension of OWL-S[16] service ontology with context conditions and adaptation rules for improved delivery of entertainment services.

The works tackling service composition challenges are usually taking advantage of extended service descriptions to provide better matches. For example, Forkan, Khalil, & Tari [2014] have proposed the creation of tree-like structure describing possible contexts and services associated to that context. The service composition is converted to the task of searching best context matches (tree nodes representations) and finding corresponding leaves (or services). The tree model is created with the help of CoCaMAAL model, which is a unified ontological representation of patients, devices, and computational services for ambient assisted living domain. Another approach, proposed by Santofimia, Fahlman, del Toro, Moya, & Lopez [2011], take advantage of new generalized model of Service, Device, Event, and Action for ambient intelligence systems. The authors are proposing this model and providing implementation of the planner, which chooses suitable devices and services for each given user at present location and time. Also, Han et al. [2014] describe the way of ontological service descriptions and further service compositions, introducing developed Composition Plan Description Language (CPDL). CPDL is an xml-like description of users and available services for the given location; these descriptions are created for each context, so the corresponding services might be invoked, when context conditions are fulfilled. Based on the analysis of the available works, it is evident that semantic service composition is a relevant area of research with quite wide range of the possible application domains. However, each of the presented approaches is the unique way of modelling context and services, which often based on design “from scratch”. Thus, the developed models and service descriptions are poor generalizable and have little acceptance outside of the scope of the research works proposing them.

Another way of using ontological models is the creation of context models, which are updated and queried dynamically. In these systems, the context model captures relevant for the application concepts and relationships. Later, this model is usually updated with data coming from devices, and then is queried for needed information. The works describing these approaches often utilize semantic reasoners and SWRL rules for improved expressivity and reasoning capabilities. For example, Armand, Filliat, & Ibanez-Guzman [2014] have proposed the system for context-aware driving assistance, which models road situations and descriptions between traffic entities in the ontology. The sort of “predictions” or context inferences are made with the help of rules, which create new entities in the ontology depending on ongoing situation. These new entities are returned as a query results, based on

[16] OWL-S: Semantic Markup for Web Services (accessed 27.09.2017): https://www.w3.org/Submission/OWL-S/

which the proper actions could be taken. The Nadoveza & Kirtsis [2014] have proposed the ontology for modelling context of business situations to address the problem of information overload in business applications. Their model contains business specific information of users and processes, which is further enhanced with SWRL rules describing selection principles of information suitable to specific users. Another example of modelling context with ontologies could be found in Xue et al. [2014] work. The authors have proposed context-aware system improving safety of the miners. The proposed system is enabled by context modelling of miner-related situations in the Coal Mine Semantic Sensor Network (CMSSN) ontology. It is one of the few works, which attempts to reuse already developed ontologies. Specifically, the CMSSN model is composed of partial adoption of SSN ontology, DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering) Ultralite (DUL[17]) version, and extension with coal specific terminology and contexts. Again, these works demonstrate wide applicability of SW technologies in different domains; however, they also reveal deep challenges associated with pure SW approaches for dynamic distributed systems, which are tackled in the following subsection.

Challenges Associated with SW Technologies

The ontological modelling is a powerful expressive way of capturing information about the concepts and their relationships. Originally, OWL ontologies were developed for description of almost static (at least in the past) information on the Web. Dynamic sensory inputs of context-aware applications pose additional challenges for context modelling with ontologies. Specifically, additional measures must be taken to address reusability issues between different applications of the same domain, temporal reasoning support (processing events with respect to the time scale), uncertainty handling (e.g., missing or false sensor data), and scalability issues [Ye, Dasiopoulou, et al., 2015], [Díaz Rodríguez, Cuéllar, Lilius, & Delgado Calvo-Flores, 2014].

The challenges associated with *reusability* of ontologies originating from the fact that applications are highly heterogeneous in their nature and often operate over information from different domains. Thus, it is quite often that ontologies are developed from scratch with a specific application in mind. The problem arises, when several of such applications want to share data between them. In these cases, applications often face conflicts and ambiguity between used knowledge models. The research community addressed this problem from different perspective. For example, Jardim-Goncalves, Coutinho, Cretan, da Silva, & Ghodous [2014] have proposed negotiation platform for sustaining interoperability between different enterprises. The NEGOSEIO (Negotiations for Sustainable Enterprise Interoperability with Ontologies) framework provides cloud infrastructure for negotiation services and procedures. The procedures are aimed to resolve conflicts of the participating sides and align their business ontologies accordingly. Another way is to introduce formal procedures for enriching ontologies with new terms, which would ensure no conflicts and ambiguity between business-important

[17] DOLCE Ultralite ontology (accessed 27.09.2017): http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite

terminology [Borgo, 2014]. Finally, the quality of resulting ontologies could be checked. Rico, Caliusco, Chiotti, & Galli [2014] have proposed the OntoQualitas software, which measures submitted ontologies against multiple quality criteria. The quality criteria include ability of the ontology to sufficiently represent knowledge and infer new one (e.g., presence in the ontology of necessary and sufficient conditions, existential and universal restrictions, domains and ranges of relations, etc), measurement of ontology conciseness, and correct interpretation of the information.

Another way to address reusability and interoperability is to adopt already existing ontologies. This is also not easy task, since developed ontologies tend to be either too broad and generic, which are hard to adopt for practical use, or too narrow and specific, which are practical, but mostly act as single point solutions [Morris & Kulvatunyou, 2017]. To address this problem, some of the researchers have tried to develop guidelines and methodologies for adoption of existing generic ontologies. For example, Fernandez-Lopez, Gomez-Perez, & Suarez-Figueroa [2013] have proposed the methodology for identifying candidate ontologies, concepts, properties and relationships for reuse. The work also provides steps for proper ontology assembly. Another attempt of ontological standardization and generalization comes from industrial domain. In there, the Industrial Ontologies Foundry (IOF) activity is proposed. It aims to gather and create the set of industrial ontologies, which would cover the need of interoperability of tools, data, and supply chain partners in manufacturing systems [Morris & Kulvatunyou, 2017]. The results of the first workshop have reported around 25 existing manufacturing-related ontologies, which are a starting point for forming requirements and needs. Therefore, even though there is a wide variety of developed ontologies and vocabularies, there are already some indicators of gradual convergence of ontologies towards consistent information sharing.

The problem of ***temporal reasoning support*** originates from the fact that first ontologies were developed with almost static information in mind. Thus, originally there were no support for expressing and reasoning over temporal relationships between entities in the ontology, which is apparent drawback for context-aware environments. The research community also addressed this problem from different perspectives. One of the main approaches is to introduce temporal formalism into ontology itself [Okeyo, Chen, & Wang, 2014], [Krieger, 2010], [Hobbs & Pan, 2004]. For this purposes, most of the works rely on ISO 8601[18] standard for time representation and Allen's temporal calculus for defining relationships between time instances [Ye, Dasiopoulou, et al., 2015]. These approaches often result in quite heavy representations, which often just describe entities from time perspective, but do not allow reasoning and processing of the information with respect to time. Another approach is to develop a lightweight temporal model with a set of tools enabling temporal reasoning among the concepts [O'Connor & Das, 2011], which binds the application to certain type of temporal ontology and supported concepts in it. Alternatively, the simple temporal relationships can be supported with the SWRL, which offers an opportunity to express custom relationships and rules and to

[18] ISO 8601 Date and time format (accessed 28.09.2017): https://www.iso.org/iso-8601-date-and-time-format.html

analyze temporal relationships with built-ins for comparison[19]. Still, temporal reasoning for ontology-based knowledge models is an ongoing issue, which does not have a standardized approach yet.

The ontological modelling is based on the first-order description logic, which means that its statements can be evaluated to either true or false. Original ontologies are not capable to deal with *uncertainty*, which is inherited attribute of context-aware systems (e.g. false or missing readings of sensors, lack of context information, etc.). Thus, introducing uncertainty concepts to ontological models has also been tackled by many researches as a needed property of semantic models. One of the straight forward ways is to introduce terminology for uncertainty suggested by W3C. The dedicated group has proposed Uncertainty Ontology[20] for this purpose. Although it gives quite expressive instrument for describing uncertainty, its origins, nature, and type, it does not actually provide reasoning capabilities to deal with uncertainty in real applications. Another way was proposed by Diaz Rodriguez et al. [2014], where the authors have suggested the fusion of ontologies with fuzzy logic and embracing the rule of “whatever is more or less true is true” instead of “only altogether true sentences are true”. The proposal includes modifications of ontology, which would be supported by fuzzy reasoner (fuzzyDL in this case) calculating the degree of belief for certain activities. Although it is a big step towards uncertainty handling, the considered in the paper use cases are rather simple and unrealistic. In addition, paper does not provide details on how original beliefs were formulated for different event types.

The most popular approach for uncertainty handling is the division of responsibilities between semantic reasoning and machine learning for uncertainty handling with the following cross sharing of results. For example, Ongenae et al. [2013] have proposed the approach, which utilizes machine learning techniques for inferring the probabilities of user activities. The resulting probabilities are later fused into ontology. This new knowledge provides sort of ranking of inferred by ontology activities, thus improving accuracy of sensor readings. The similar approaches were suggested by Aloulou, Mokhtari, Tiberghien, Endelin, & Biswas [2015] and Noor, Salcic, & Wang [2016]. These research groups have proposed the way of calculating uncertainty for sensor readings. This uncertainty is later incorporated into ontological models (Aloulou et al. have proposed to do it as an additional property of the predicate connecting subjects and objects in the ontology). On later stages, the information from ontology with associated probabilities is used in combination with Dempster–Shafer theory (DST), which provides mechanisms for reasoning with uncertainty and calculating final decision of most probable events and situations. To summarize, the present-day context-aware solutions based on ontological modelling are capable to deal with uncertainty. Since the original semantic tools are not capable to support such reasoning, these approaches are mostly hybrid-based requiring additional effort in adaptation and development.

[19] SWRL built-ins for comparison (accessed 28.09.2017): http://www.daml.org/rules/proposal/builtins.html#8.1

[20] W3C Uncertainty Reasoning for the World Wide Web Incubator Group (accessed 28.09.2017): https://www.w3.org/2005/Incubator/urw3/group/draftReport.html

Finally, utilizing SW technologies especially in large-scale systems requires careful consideration of *scalability* issues, since reasoning is always computationally expensive [Margara, Urbani, Van Harmelen, & Bal, 2014]. The performance bottlenecks could arise in two places: reasoning and retrieving query results. One of the strongest factors affecting reasoning performance is the size of the knowledge model (i.e. the number of individuals) [Kang, Li, & Krishnaswamy, 2012], while the factors affecting query performance include the number of returned by query results and query complexity [Horrocks, Li, Turi, & Bechhofer, 2004]. The performance times could greatly vary from a few milliseconds for some applications to tens of seconds for others [Gellrich, Lunkwitz, Dennert, & Kabitzsch, 2012], [Kang et al., 2012]. Thus, each application should carefully consider different parameters of the model and implement rigorous performance tests for ensuring declared operating parameters.

To summarize, SW technologies is a very broad area of research with applications in different domains and various systems. The broad application area resulted in formulating challenges associated with SW technologies. Although these challenges are real and require addressing by practical applications, in the context of this section they should be interpreted as possible ways for improving SW technologies, rather than significant limitations.

2.3.3 Event-Driven

Event-driven methods are focused on events or messages as fundamental building blocks of communication. These methods analyse emerged in the systems events (e.g. performing filtering, event correlation, further message distribution) and carry out corresponding actions. Context-aware systems is a natural application area of such methods, where distributed components of the system communicate via events and messages. This subsection covers publish/subscribe middleware, Complex Event Processing (CEP), and Stream Reasoning.

2.3.3.1 Publish/Subscribe Middleware

The publish/subscribe middleware are solutions providing asynchronous communication between multiple entities of the system. The approach is especially popular in Wireless Sensor Networks, which consist of multiple communicating devices and require efficient ways of communication [Seeger, Laerhoven, Sauer, & Buchmann, 2013], [Russello, Mostarda, & Dulay, 2011], [Shi, Deng, & Qin, 2011]. The publishers produce information to the consumers (subscribers). Usually, publishers and subscribers are unaware of each other, i.e. publisher does not know to whom it produces information, while consumer is unaware from where the piece of information came. The communication is usually done through the set of message brokers, which handle subscriptions and redirect the messages. There are several possible ways of how subscriber declares interests in the certain type of messages. The most common subscription mechanisms are topic and content based. In the topic-based mechanism consumer subscribes to certain types of topics (or channels), while in the content based mechanisms the delivered messages are filtered based on the actual content of the message. Thus, the selective delivery of information could be achieved by client subscriptions to the certain types of published messages.

The publish/subscribe paradigm is a widespread approach in distributed, event-based applications, thus bringing research interests into various aspects of publish/subscribe communication. For example, some of the works are concerned with providing scalability of solution in terms of number of subscribers and message rates [Arias Fisteus, Fernández García, Sánchez Fernández, & Fuentes-Lorenzo, 2014] or modelling communication networks for more efficient message routing [Shi et al., 2011]; while the others propose approaches for generalized performance evaluation of publish/subscribe systems [Mühl, Parzyjegla, & Prellwitz, 2015]. Also, quite intensive topic of research is in increasing flexibility of subscription mechanisms. For example, in [Cañas, Pacheco, Kemme, Kienzle, & Jacobsen, 2015] authors propose the graph-bases subscription mechanism, where the graphs model part of the system entities and simplify subscriptions for users interested in certain type of the information; the work described in [Arias Fisteus et al., 2014] deals with semantically annotated data streams, and also proposes filtering mechanisms based on semantics of published data.

While publish/subscribe is a powerful approach in redirecting and delivering streams of information in distributed systems, it has some limitations in application to selective delivery in monitoring systems. The limitation comes from the fact that publish/subscribe paradigm is originally developed as a way of communication, not the analysis of data streams; thus, the capabilities to perform ongoing analysis of data streams are pretty much limited to content-based filtering of information, and are not capable to detect complex interrelated patterns of events between several streams.

2.3.3.2 Complex Event Processing

Complex Event Processing (CEP) is a set of tools and techniques for analysing and controlling the complex series of interrelated events of distributed information systems [Luckham, 2002]. Essentially, CEP engines analyse ongoing events and match them against predefined patterns or rules to trigger corresponding actions [Cugola & Margara, 2012]. In context-aware applications, CEP is often used as a way to track multiple conditions and trigger needed actions or services, when condition is met.

The available CEP solutions are very different in nature and supported functionalities; thus, it is difficult to provide common classification. Some of the works have proposed to classify CEP solutions based on the underlying technological background. For example, Cugola & Margara [2012] split CEP solutions into two general classes: (1) Data Stream Management Systems (DSMS), and (2) Complex Event Processing. In this way, they distinguish some of the CEP solutions, which originally came from database management systems. The difference is that DSMS solutions operate over the data streams to produce new streams of data. In order to do it, the users create continuous or standing queries, which are put into system and continuously return results, when query is matched. Often, these queries are SQL-like, which are quite limited in expressing complex interrelationships between streams. On the other hand, CEP solutions put great effort on tracking of complex event patterns. Usually, these systems operate over production rules of the form `if-condition(s) then-action(s)` and are sometimes seen as evolution of publish/subscribe mechanisms (which only provide topic and/or content-based filtering). However, even with this suggestion, authors acknowledge that sometimes the classification border between different CEP solutions is blurred.

Another classification approach was suggested by Eckert, Bry, Brodt, Poppe, & Hausmann [2011]. The authors propose classification based on the type of underlying event processing language (EPL). They distinguish five types of EPLs: (1) composition operators, which are similar to the first-order logic syntax, (2) data stream query languages, which are similar to SQL queries, (3) production rules, which are `if-condition(s) then-action(s)` type, (4) timed state machines, which looks similar to ladder logic programming, and (5) logic languages, which similar to proposition logic with Prolog engine being the main example. Even though the authors have been quite successful in classification of existing solutions, they still acknowledge, that there are great differences in languages between different solutions within the same group. Thus, CEP approach is quite popular in many domains; however, one need to be careful to search for solution, which would suit the needs of the applications, because there are many available options differing in underlying technologies, expressivity, and suggested languages.

CEP in Context-Aware Applications

Usually CEP engines are standalone tools, which might be easily adopted by application via provided API. They provide scalable infrastructure for analysing streams of different events. The examples of free and commercially available engines are Drools Fusion[21], Esper[22], Oracle Complex Event Processing[23], Apache Flink[24], etc. The research community have been also addressing the development of such engines for special needs. For example, Zappia, Paganelli, & Parlanti [2012] have proposed and developed Lightweight Stage-based Event Processor (LiSEP) engine, which is a portable lightweight solution featuring extensibility and scalability. The engine is developed in accordance with SEDA (staged event-driven architecture) pattern, which decomposes complex event-driven application into a set of stages connected by queues and decouples event and thread scheduling from application logic (thus enabling features of LiSEP). The engine provides API and supports SQL-like syntax as EPL.

There are some works, which bring their CEP solution for better management of RFID devices and events. For example, Zhao, Liu, & Lin [2012] have proposed the way of RFID devices integration using CEP approach. The approach is seen as incorporation of business logic into RFID edge systems. For these purposes, the

[21] Drools Fusion (accessed 12.10.2017): http://drools.jboss.org/drools-fusion.html

[22] Esper event processing for Java (accessed 12.10.2017): http://www.espertech.com/products/esper.php

[23] Oracle Complex Event Processing (accessed 12.10.2017): http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html

[24] Apache Flink (accessed 12.10.2017): https://flink.apache.org/

authors have developed the formal way of event and rule modelling based on event calculus and used it in combination with Discrete Event Calculus (DEC) reasoner, which simplifies tracking and interpretation of RFID-associated events. Another CEP approach for RFID device system was proposed by Xin & Zhang [2014]. In this work, authors have focused on RFID-enabled business delivery scenario. In this scenario, the RFID events and transactions' related information (such as tags, transactions, sellers, and buyers of the system) are put into ontology for semantic interpretation, which is later sent to developed semantic complex event processing engine (ESCEP-E). Based on this information, ESCEP-E analyses timestamps and possible actions to determine destination of the parcel in the conveyor (e.g. manual operation, sorting, special handling for fragile items, etc.). Although the given examples might answer the described applications needs, mostly they are proprietary solutions with narrow specific EPLs and very little adoption outside of the designated applications.

More common practice is the usage of available CEP engines. For example, Terroso-Saenz et al. [2015] have used the Esper CEP in vehicular context-aware application. In this work, the authors have split the architecture of the system into separate blocks, responsible for detection of certain activities and/or complex events. Each block features usage of Esper CEP for understanding of local features and events. Later, this information is used in detecting patterns and in on-line density-based clustering for determining occupancy and places (landmarks) of the vehicle. Another example, proposed by Yao, Chu, & Li [2011], features usage of Drools CEP. In this work, authors have been working with RFID devices in the hospital scenario. They have presented the workflow of using CEP engine, which provides semantic interpretations of originally not human friendly RFID data. As a result, the authors were able to present this information in the way convenient for understanding, which also included alerts and messages, when needed.

Challenges Associated with CEP

The CEP approach is a powerful way of tracking multiple conditions in various distributed systems. The main entities of CEP are events, so the temporal relationships, interrelation between events, and scalability are often inherited properties of CEP solutions. However, the sole event focus might hold significant drawbacks. In general, CEP solutions are hard to properly configure ensuring stable predictable work, and it is even harder to switch between different engines, if needed.

There are different approaches addressing *CEP configuration* challenge. One of the ways is to provide additional support for CEP engines. For example, Drools Fusion is accompanied by Drools Guvnor[25] repository. The repository provides GUI, editors and tools to manage Drools rule databases: creation, versioning, functions, and access rights. This greatly simplifies the process of creation and management of rules; however, it only supports specific engine, and would require additional efforts for other engine providers in development and support of similar products. Another way was proposed by Weiss, Mandl, & Schill [2011]. The authors have recognised the need to ensure quality assurance for CEP applications, and have proposed

[25] Drools Guvnor (accessed 12.10.2017): http://drools.jboss.org/drools-guvnor

`CEPTestConnector` interface, which supports CEP requirements for functional test. Specifically, it allows to test the number of expected rule firings, attributes of events, time control features, etc. Still, the proper configuration and rule quality insurance are major challenges associated with CEP.

If there is a need to switch between different CEP engines, the problem of ***EPL diversity*** arises. The multitude of CEP engines is accompanied by multitude of CEP languages, which lack standardisation and exchange mechanisms. As a result, the rule databases cannot be seamlessly translated between different engines; the process of translation requires significant effort in learning new EPL, translating rules, and ensuring the final quality of new rule database. The research and industry communities have attempted to approach this problem. For example, Boubeta-Puig, Ortiz, & Medina-Bulo [2014] have proposed the EPL meta model to support experts in creation and management of CEP rules. The authors' vision is to simplify creation of such rules by keeping domain experts oblivious to concrete syntax corresponding to specific EPL. Another approach was proposed by Taylor & Leidinger [2011]. The authors have hidden the CEP rules creation and configuration process under configuration of ontology. Furthermore, to exclude the need of learning additional technology, the authors have developed Protégé plugin, which hides ontology at the configuration process. Afterwards, the ontology is translated into native CEP rules supported by middleware, which are ready to load and run in the corresponding programs.

The industry has approached this problem by creation of dedicated work groups and standards. Specifically, Rule Interchange Format[26] (RIF) by W3C group and Production Rule Representation[27] (PRR) standard by Object Management Group[28] (OMG) are worth mentioning. RIF group has focused on several rule dialects, such as several types of logic and production rules, and have tried to create mechanisms for easier rule exchange between those dialects. As a result, RIF become W3C recommendation in 2010 with partial developments for some of the dialects. The OMG is an open membership, not-for-profit computer industry standards consortium. One of the standards of this consortium is PPR specification defining conceptual classes of the rules, which could be adopted by different CEP engine vendors for rule translations. Although PPR was released in 2009, it does not seem to have great adoption rate.

To summarize, CEP technology is a powerful way of interrelating events in distributed environments. It allows analysis of complex temporal relationships across multiple places with CEP engines provided by multiple vendors and research communities. However, it is quite challenging to ensure consistency and predicted behaviour of large rule databases, which becomes even more challenging if change of CEP engine needs to be done.

[26] Rule Interchange Format working group (accessed 12.10.2017): https://www.w3.org/TR/rif-overview/

[27] Production Rule Representation standard (accessed 12.10.2017): http://www.omg.org/spec/PRR/

[28] Object Management Group (accessed 12.10.2017): http://www.omg.org/about/index.htm

2.3.3.3 Stream Reasoning

The CEP approach is only capable to anticipate context defined in the rule, i.e. it is not possible to derive implicit knowledge from explicit rule statements. On the other hand, SW are capable to infer implicit knowledge, but are not intended to deal with active data streams. With this in mind, the concept of Linked Stream Data (LSD) has emerged, as a Linked Data (or semantic knowledge modelling) principles applied to sensor stream data [Sequeda & Corcho, 2009]. The LSD is seen as a bridge between logic reasoning and stream processing [Ye, Dasiopoulou, et al., 2015]; thus, the engines enabling this kind of reasoning are called Stream Reasoning engines. Briefly, the stream reasoners operate over sensor data streams, enriched with RDF descriptions, and provide results to continuous queries[29] specified by users. The results are based on reasoning over RDF data streams and corresponding RDF schemas; thus, seamlessly enriching stream processing with knowledge reasoning capabilities [Le-Phuoc, Dao-Tran, et al., 2012]. Since the concept is bridging the gap between semantic reasoning and stream processing, the context-aware systems are one of the best candidates for exploiting this emerging technology in their solutions.

Engines for Stream Reasoning

The stream reasoning is an emerging area of research, so research community have started to address it by developing dedicated stream reasoning engines. The most common stream reasoning engines, which are also discussed in this work, are C-SPARQL, EP-SPARQL, CQELS, and Sparkwave.

The ***C-SPARQL*** engine was proposed by Barbieri, Braga, Ceri, Valle, & Grossniklaus [2010]. The approach underlying this engine includes splitting the stream query into two parts and delegating these parts to specific engines. The first one is the static part of the query (i.e. related to knowledge modelling), which is delegated to reasoning engine through SPARQL plugin (Jena in this case). The second one is stream related part, which is delegated to DSMS engine (Esper in this case). In this way, the authors were able to reuse existing engines and concentrate their effort on the query wrapping part and development of C-SPARQL language for submitting continuous queries with extended semantics for time windows and sensor stream relationships. However, the drawback of this division of responsibilities lies in inability to update the non-stream data, because the static part uses semantic reasoner, which does not support continuous queries [Le-Phuoc, Parreira, & Hauswirth, 2012].

The ***EP-SPARQL***, proposed by Anicic, Fodor, Rudolph, & Stojanovic [2011], is backed by Prolog-like ETALIS logic engine. Essentially, it translates RDF triples of queries into logic programs. The execution of EP-SPARQL is based on event-driven backward chaining (EDBS) of logic rules, making it possible to mix domain

[29] Continuous querying is the mechanism of constant evaluation of query pattern when new data is added to the system and updating the corresponding result set as an opposite to traditional queries when a query should be invoked explicitly by the caller and the result set remains the same regardless of the new data in the model.

knowledge and stream data. Thus, it is an integrated approach to knowledge reasoning and stream processing [Le-Phuoc, Parreira, et al., 2012].

The *CQELS* engine was proposed by Le-Phuoc, Parreira, et al. [2012]. Unlike previous engines, this engine is fully developed by one research group. The native development enabled usage of RDF triples as a first-class data elements and implementation of efficient data structures and adaptive performance behaviours based on intensity of data streams. For expressing continuous queries, the SPARQL 1.1 is extended to CQELS language. Unlike many related works, where only engines and performance tests are provided, the authors of the engine have exploit it in developed Linked Stream Middleware (LSM) for Internet of Things (IoT) applications [Le-Phuoc, Nguyen-Mau, Parreira, & Hauswirth, 2012]. The LSM is oriented towards large-scale data-intensive distributed systems, where authors further elaborate on scaling scenarios for linked stream data processing.

Finally, *Sparkwave* engine, proposed by Komazec, Cerri, & Fensel [2012], features RETE algorithm as underlying engine implementation extending it to support time-based sliding windows. RETE algorithm is a pattern-matching algorithm for implementing production rule systems, which is often used for CEP engine implementations. The engine is also natively developed and is oriented to high-performance achieved in two ways: (1) RETE algorithm on its own trades memory consumption for improved performance, (2) the authors of Sparkwave explicitly trade reasoning capabilities over performance, reducing supported statements in reasoning (although other engines do the same things, however without emphasis on supported reasoning capabilities). The authors have tested the performance of Sparkwave engine in comparison to CQELS and C-SPARQL, where Sparkwave considerably outperformed these engines in throughput and memory consumption.

Since developed engines greatly vary in their implementation approaches and application scenarios, it is quite challenging to compare their performance. For this purposes, Le-Phuoc, Dao-Tran, et al. [2012] have developed evaluation framework and methodology for data generation, system testing, and result analysis of stream reasoning systems. This framework has been applied to C-SPARQL, JTALIS (according to Le-Phuoc, Dao-Tran, et al., EP-SPARQL had problems in RDF parsing, so they used JTALIS as Java wrapper for ETALIS), and CQELS engines. Their results showed that C-SPARQL has considerably lower throughput performance (by several orders of magnitude). In turn, CQELS shows better performance, mainly because it uses native development approach, and other engines heavily depend on underlying systems for performance (e.g. DSMS for C-SPARQL, and Prolog for JTALIS). However, there is a possibility that their framework is tailored for specific applications supported by CQELS. For example, Ren, Khrouf, Kazi-Aoul, Chabchoub, & Cure [2016] have also proposed experiments testing stream rate, number of triples, time window size, number of streams and static data size (i.e. information in knowledge model, which does not change with stream arrival) on execution time and memory consumption of CQELS and C-SPARQL engines. According to their experiments, the C-SPARQL was outperforming CQELS engine in three out of four experimental queries (the only low performance query was due to the large number of static data). Although, the authors acknowledge that they were measuring the average time of retrieving results, since C-SPARQL only supports batch query processing in opposition to eager mechanisms, when each query results are re-evaluated on each update from the streams. Still, performance results of other authors reveal even more disturbing details. For example, Dejonghe [2016] and Komazec et al. [2012] reveal that actually CQELS and C-SPARQL public distributions do not support any reasoning capabilities. Thus, EP-SPARQL (supporting RDFS reasoning in Prolog) and Sparkwave (supporting RDFS subset and `owl:inverseOf` and `owl:SymmetricProperty` relationships) are the most expressive stream reasoning engines available with Sparkwave being also the most high performance engine.

Challenges Associated with Stream Reasoning

The stream reasoning aims to bridge the gap between knowledge reasoning and stream processing; thus, the main challenges lie in scalability and reasoning capabilities of proposed solutions, since reasoning is computationally expensive and sensor data streams require high throughput. These requirements are not straightforward to address, but they are mandatory for the application area of stream reasoning systems; thus, some of the researchers are referring to stream reasoning as still being in its infancy period [Margara et al., 2014].

For now, most of the researchers have focused on ***scalability improvement*** for reasoning engines, which comes at the cost of knowledge reasoning capabilities. For example, CQELS and public distribution of C-SPARQL do not support any reasoning capabilities, while EP-SPARQL and Sparkwave, being the most advanced in reasoning capabilities, only support part of RDFS (and two of OWL relationships for Sparkwave) with no full OWL support. Thus at the present, the bridge between knowledge reasoning and stream processing mostly comes from one side of RDF-like stream processing.

There are some approaches attempting to improve ***reasoning capabilities***. These works propose pipeline architecture for stream reasoning systems. The idea behind is to use stream reasoners for preliminary filtering of data streams and connect them to other engines for advanced knowledge reasoning of remaining data. For example, Dejonghe [2016] has proposed to connect stream reasoner with classical semantic reasoner for context-aware applications, while Mileo, Abdelrahman, Policarpio, & Hauswirth [2013] have proposed to use ASP solvers on filtered data streams for advanced reasoning capabilities. Their vision have resulted in formulation of PhD thesis plan elaborated by Pham [2015]. However, none of the approaches has resulted in implementation with performance tests reported so far.

To summarize, the stream reasoning vision finds a promising application area in context-aware systems by closing the gap between knowledge reasoning and stream processing. However, the reality of advancement for stream reasoners reveal average results for both worlds: stream reasoners do not provide the same level of stream processing performance as other CEP engines, and they considerably lack the knowledge reasoning capabilities in comparison to knowledge reasoners. The recent works propose the combination of the stream reasoners with other knowledge reasoning techniques (e.g. semantic reasoners and ASP solvers), but none of them has resulted in implementation yet.

2.3.4 Summary of Techniques and Technologies

Context-aware systems is a large area of research with multiple technologies involved. Special attention should be put on the choice of underlying technologies, which will be able to address application needs. Table 2-4 summarizes technologies against the qualitative attributes highlighted in section 2.1.4. Based on the considered technologies and their advancements/drawbacks in certain areas, it was decided to present two additional attributes connected with expressivity of the solution, namely: temporal reasoning support and uncertainty handling. The temporal reasoning support is connected with ability of technology to relate data from different sources against the time scale, while the uncertainty handling summarises the ability to deal with erroneous sensor data or lack of knowledge. Also, the evaluation of technologies is done without consideration of mentioned hybrid approaches (e.g., SW technologies are still graded as providing weak support for temporal reasoning and uncertainty handling, even though there are some proposals partially overcoming these difficulties), because they are tailored to specific use cases without wide acceptance outside of the application.

Table 2-4: Summary table of technologies and their qualitative attributes

Techniques and Technologies		Expressivity	Temporal reasoning support	Uncertainty handling	Runtime performance	Scalability	Ease of configuration
Data-driven	Supervised learning	∼	∼	●	●**	∼***	∼
	Unsupervised learning	∼	○*	●	●**	∼***	∼
Knowledge-driven	Answer Set Programming	●	∼	-	∼	∼	○
	Semantic Web Technologies	●	∼	-	○	∼	●
Event-driven	Publish/Subscribe middleware	∼	∼****	-	●	●	○
	Complex Event Processing	○	●	∼	●	●	∼
	Stream Reasoning	∼*****	●	∼	●	○	○

● - high
○ - moderate
∼ - low
- - none

* - mined rules consider temporal relationships
** - the performance is quite high, after the models are created
*** - it is difficult to provide solution to multiple places (for training models, a lot of personalized data is required)
**** - no temporal support between different streams is provided
***** - it is possible to have expressive models, but it is not possible to reason over them

As it could be seen from Table 2-4, the strongest support for needed qualitative attributes is provided by CEP and Stream Reasoning. However, these technologies lack the ability to reason over the knowledge of the domain (expressivity feature) and are quite weak in configuration support. The data-driven approaches provide strong features for uncertainty handling; however, they are not user friendly in configuration and require more effort in building models for classification. In addition, these methods require extensive labelled datasets for training, which are not generalizable to multiple users/apartments, and thus are hard to apply in BMS. Finally, ASP and SW are very strong technologies in providing expressivity of knowledge models; however, they lack temporal reasoning support and uncertainty handling mechanisms as well as struggling with reasoning over large systems.

Thus, to fully address the required qualitative attributes of the solution, the combination of technologies is seen as a viable option. Specifically, CEP and SW are considered as the best match for strengthening the final solution. The SW provide strong support in modelling domain knowledge, which could be used at the configuration stage of the solution; while CEP provides runtime reasoning capabilities and scalability properties for the final solution. The solution would still lack the uncertainty handling capabilities, which are only partially covered by CEP (e.g. its capability to vary parameters and thresholds for rules, and thus partially handle uncertainty). However, as it was mentioned in section 1.5 (Assumptions and Limitations of Scope), it is assumed that low-level data filtering and handling are done outside of the middleware, thus allowing to leave advanced uncertainty handling considerations for future work.

2.4 Literature Assessment

Modern BMS are large scale and include multiple heterogeneous devices monitoring environment conditions and users' status. Delivering this information in full overwhelms users and hinders potentially important information; thus, complicating daily tasks and potentially creating dangerous situations. Therefore, relevant information delivery is an important goal, which would improve qualitative aspects of delivered information and reduce information load to users of the system.

Relevant information is greatly defined by information needs prevailing in the domain of interest. With respect to BM, high number of data points and variety of users mostly dictate information needs. Specifically, delivery of well-prioritized structured information is valued, which would be tailored to user roles and ongoing tasks in the environment. In addition, BM domain lacks standardized notification practices; thus, the adaptation of good notification and alarm design practices to BM domain is needed. In that respect, the process automation domain is technologically related to BM and has a good standardisation support in alarm design; thus, it is seen as the best source of information for BM domain. Furthermore, high number of devices and users in the system, increases chances of introduction of changes in the system (e.g., new devices and/or users, access right management, configuration of existing devices); thus, in addition to scalability and good runtime performance, the solution of relevant information delivery requires ways for flexible configuration of the system.

The delivery of selective information and services out of many available options is addressed in many system types, specifically: expert systems, recommender systems, situation-aware, context-aware, and cyber-physical systems have been considered. These systems differ in the types of data sources, core technological properties, and considered application areas. Context-aware systems seem to provide the best match between conceptual idea of the system and requirements for relevant information delivery; they consider ongoing status of the environment and users as part of the environment in order to meet their personalized needs, which seems to be the way of accomplishing relevant information delivery.

Context-aware systems have multiple implementations, which are possible to unify only at conceptual level. Thus, the techniques and technologies of context-aware systems are diverse in nature. In general, they can be divided in data-driven, knowledge-driven, and event-driven. Data-driven technologies dominate on lower layers of conceptual context-aware architecture. These technologies are often used for activity recognition and information filtering from sensors of the system. Knowledge-driven methods dominate context calculation and context provision layers of the architecture. They are powerful in modelling domain knowledge and expressing different relationships between concepts of the system. Finally, event-driven methods operate over the events in the system and are dispersed across several layers of the architecture. They provide different event-related properties including aggregation, matching of complex event patterns, and modelling of the domain knowledge in the form of rules.

The detailed description of considered technologies of context-aware systems and their matching against required qualitative attributes of relevant information delivery revealed that combination of SW and CEP is the best match for addressing all of the requirements. Specifically, expressivity and ease of configuration of SW technologies might be used at the configuration stage of the middleware, where available information must be categorised and semantically interpreted for better classification and runtime analysis. The CEP, on the other hand, might be used at the runtime of the middleware, where information from devices must be quickly processed and delivered to various users in a timely manner.

3 Delivery of Relevant Information in BMS

The development of middleware for relevant information delivery should be approached from different perspectives starting from analysis of user information needs to forming technological solution enabling relevant information delivery. This section presents five peer-reviewed publications composing the core of this research work.

The rest of the chapter is organised as follows. Publication I provides details of user information needs and reports the results of preliminary usability tests of the system supporting these information needs.

Publication II contributes to the principles of knowledge modelling for relevant information delivery and proposes the first implementation architecture supporting utilization of this knowledge model.

Publication III presents initial knowledge-based technological solution to relevant information delivery and provides design and implementation details of this approach. In addition, the paper provides examples of context-aware and reconfiguration scenarios for the system, demonstrating flexibility of the approach. Part of the performance test results suggested potential problems of extending this approach to large use cases leading to development of improved hybrid approach.

Publications IV contributes with the improved technological solution for relevant information delivery. It includes combination of previously used ontological knowledge models with CEP for advanced temporal reasoning support and performance improvement. In addition, the combination of these technologies enables simplified tracking of multiple situations in the system. The paper gives the implementation details of the proposed solution, provides general guidelines for developing of similar software for different systems based on general principles, provides examples of redirection of information flows in the system, and gives details of the performance tests for configuration and runtime parts of the middleware.

One of the consequences of tracking multiple situations is the increased diversity of possible notifications. On the one hand, the information becomes more related to specific situations in the environment. On the other hand, it results in increased number of notifications delivered to the user. The Publication V addresses this issue and proposes classification and combination principles of ongoing notifications in the BMS. The paper features formal description of these principles and provides results of applying these principles to a rehabilitation facility case study.

3.1 Assessment of Information Needs (Publication I)

ICT advances have dramatically extended capabilities of control and monitoring systems. One of the prominent trends is the design of smart spaces capable to monitor status and activities of its users and to address personalized needs. In relation to this trend, modern BMS incorporate multiple heterogeneous devices and, in principle, could provide diverse functionalities within monitored areas. However, to successfully address particular user needs and promote acceptance of a new technology, the systems must address user requirements, capabilities, and preferences in delivering information.

This paper provides the details of the iterative field studies summarizing expected system functionality and information needs of different user groups for the case study of a rehabilitation facility. It also features the description of developed prototype solution supporting functionalities and personalized information delivery indicated during the field studies. It is worth noticing, that at this stage no technological proposal of how to deliver relevant information has been made. Instead, the prototype was made with the usage of conventional techniques, such as SCADA HMIs and large amount of scripting and database management. As a result, the actual functionality of the solution has been limited to only essential functions needed for usability tests. The paper also reports preliminary usability test results indicating positive acceptance of an extended functionalities of BMS by residents and caregivers of the facility. In relation to relevant information delivery, the results of this paper helped to better interpret possible user requirements in BM domain and served as a basis for development of knowledge models at later stages of the research.

3.2 Knowledge Modelling Principles (Publication II)

In traditional control and monitoring systems, data from devices are strongly connected with application logic; the meaning of data entities and their relationships are distributed across programming logic and database relationships, making it hard to manage and interpret multiple data points for multiple users. For example, introduction of new devices and users in the system in most of the cases is not a trivial task. It includes thorough check of underlying programs and scripts and multiple configuration steps to ensure integrity between databases and application logic. Because of expensive reconfiguration steps, in most of the cases users are exposed to the same general interfaces delivering overall information without considering personal user needs and responsibilities. Thus, there is a need in more flexible ways of managing information and data streams in control and monitoring systems. These ways should support easier reconfiguration and information handling for large-scale multi-role and multi-user systems.

This paper addresses the problem of delivering personalized information in contemporary control and monitoring systems. It proposes semantic knowledge model capable of decoupling data from application logic and facilitating reconfiguration of multi-user large-scale systems. The paper adapts requirements and user information needs from the field studies and transforms them into semantic knowledge model of the system, capturing essential relationships between different system entities. In addition, the paper features description of illustrative scenarios demonstrating personalized information processing and filtering with the usage of designed knowledge model and proposes implementation architecture supporting the approach and continuing the work presented in [Evchina, Dvoryanchikova, & Martinez Lastra, 2012] and [Nieto Lee, Evchina, Dvoryanchikova, & Martinez Lastra, 2013].

3.3 Knowledge-Based Approach for Delivery of Relevant Information (Publication III)

Further advances in implementation and knowledge model design have led to proposal of a knowledge-based approach for relevant information delivery. The approach includes designed knowledge model as a cornerstone of information management. The model contains the information about users, devices, and locations of the system. The model is updated with runtime data from devices with further information processing by semantic reasoner. The paper also provides reconfiguration and context-aware scenarios demonstrating applicability of the approach in different situations. The traditional challenge of semantic reasoning in relation to temporal reasoning support was partially overcome by using SWRL built-ins for comparison[30].

The paper also provides the results of the performance tests presenting required times for the update and query of the model with respect to its size. In general, performance tests gave good results, especially in the part of model updating. However, the query times greatly depended on the number of returned query results, as demonstrated in Fig. 14 of the paper. In the paper, this problem has been solved by splitting the queries and careful estimation of the number of returned results, which might be problematic for other cases. In addition, the SWRL comparison built-ins allow only simplest forms of temporal relationships between temporal data, which makes the expression of time relationships between different sensors quite bulky and error prone.

In overall, knowledge-based approach is a powerful way of expressing semantic relationships between entities, which decouples data from application logic and enables easier system reconfiguration and handling of information streams; however, it requires improvements in the part of runtime performance and data analysis capabilities. As a result, new hybrid approach for relevant information delivery has been proposed.

3.4 Hybrid Approach for Delivery of Relevant Information (Publication IV)

The hybrid approach combines semantic knowledge models with CEP. The essential idea is to split configuration and runtime parts of the middleware for addressing all qualitative requirements of the required solution. The semantic knowledge models are powerful in expression and configuration capabilities, while CEP provides strong temporal reasoning support and scalable runtime analysis capabilities. The

[30] SWRL built-ins for comparison (accessed 06.11.2017): http://www.daml.org/rules/proposal/builtins.html#8.1

early implementation of the proposal with the focus on delivery of selective information streams could be found in [Evchina & Martinez Lastra, 2015].

The Publication IV paper features detailed description of the hybrid approach including application ontology, data abstractions, query and CEP rules examples adopted in the method. The core idea is the fusion of information from semantic knowledge models with runtime data from devices. This fusion is made through data abstraction classes by means of CEP engine (see section 3 and 4 of the paper for more details). The major benefits of this approach are twofold: 1) the behaviour of the system could be easily changed by configuring only underlying ontology and 2) utilization of CEP at runtime makes system event-driven and reactive to frequent changes in the environment.

The paper also provides reconfiguration scenarios of the system and proves feasibility of the approach by extensive performance tests of the configuration and runtime parts of the middleware. It was possible to achieve processing time under one second for 1000 updates per second (corresponding to 10,000 devices in the performance tests). As a result, hybrid approach enables uniform handling of multiple heterogeneous devices in the system, powerful reasoning capabilities over runtime data, and flexible reconfiguration of data streams and corresponding notifications for personalized delivery of information.

3.5 Classification and Combination Principles of Ongoing Notifications (Publication V)

Improved reasoning capabilities of control and monitoring systems enable detection of multiple abnormal situations, which leads to two-sided effect. On the one hand, the notifications become more specific and descriptive for certain situations. On the other hand, the overall flow of notifications increases, which might overwhelm users and result in worse perception of overall environment status. The ways of handling of multiple notifications are more studied in the process automation domain, where shut downs are safety critical and/or highly expensive. The practices of handling multiple abnormal situations are called alarm management.

Usually, the research works in the field of alarm management practices attempt to find badly designed alarms. For these purposes, they analyse alarm log data and other information related to the plant (e.g. topology, process data). The purpose of this analysis is to detect alarms which might cause alarm floods (the condition at which the rate of the alarms is higher than the capacity of the operators to interpret them), alarm chatter (the condition at which certain alarm types activate and deactivate excessively), or correlated alarms with the same causes. However, quite often the excessive notification delivery rate in BM domain is not the problem of badly designed alarms, but rather the result of detection of multiple situations, which might be related in certain context. In this way, it is important to enable mechanisms analysing this relation of multiple notifications at runtime, and thus reducing information load to the users.

This paper proposes the novel approach to classification and combination of ongoing notifications in BMS. It introduces formal principles of classifying possible notifications in the system with the guidelines on assigning notification types, priorities and their relationships. This classification is later used for analysis and combination of ongoing notifications in the system. The aim is to reduce repeating information pieces delivered as part of multiple notifications. The major benefit of this approach is the ability to analyse ongoing situations and deliver reduced information flow at a runtime. The approach has been applied to the data of the rehabilitation facility, where it was possible to reduce the number of active delivered notifications by around 42% at the peak number of ongoing notifications. The approach has been fully integrated into developed middleware of Publications IV and V, thus finalizing the scope of the research on relevant information delivery.

3.6 Summary

The main contributions of the thesis could be generally divided in three fields: knowledge modelling, CEP, and alarm management (Figure 3-1).

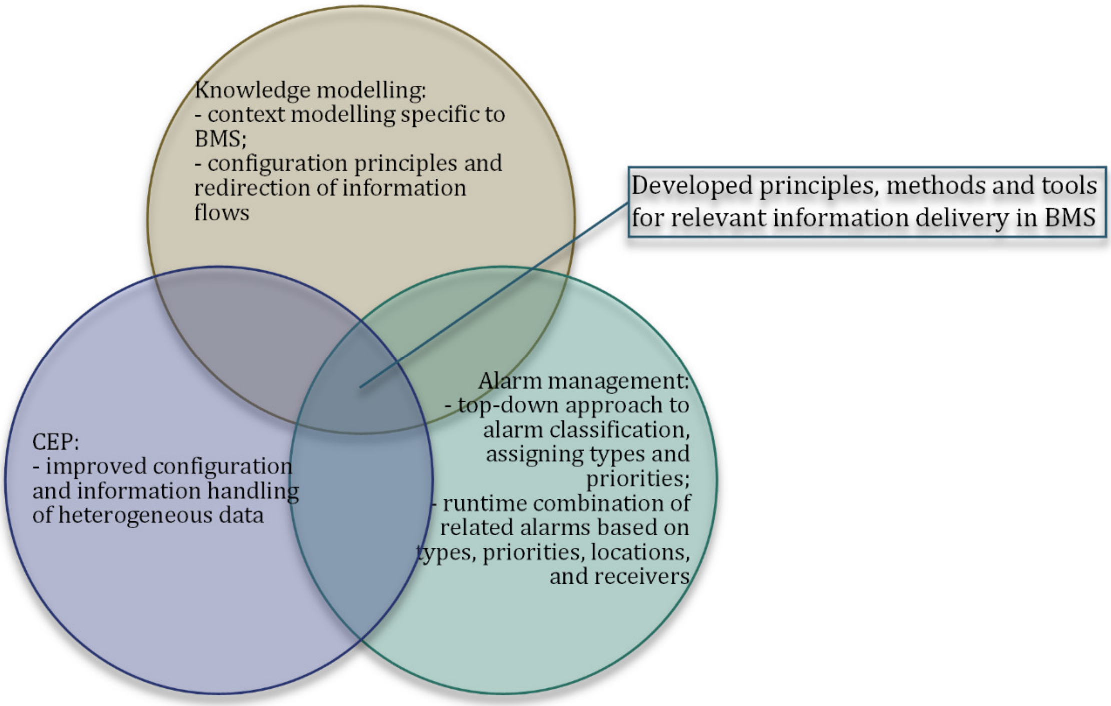

Knowledge modelling:
- context modelling specific to BMS;
- configuration principles and redirection of information flows
Developed principles, methods and tools for relevant information delivery in BMS
CEP:
- improved configuration and information handling of heterogeneous data
Alarm management:
- top-down approach to alarm classification, assigning types and priorities;
- runtime combination of related alarms based on types, priorities, locations, and receivers

Figure 3-1: Main contributions of the thesis

The research work has been started from analysing information delivery requirements in BMS and reflecting them in the designed knowledge model. Then it continued with proposing a knowledge-based approach to delivery of relevant information, which enabled simpler reconfiguration and redirection of information flows in the system (knowledge modelling contributions).

The need of improved performance and runtime analysis capabilities have led to proposal of hybrid approach for relevant information delivery, combining semantic knowledge models with CEP. While CEP provides advanced temporal reasoning and runtime analysing support, semantic knowledge models provide configuration and information handling capabilities to the solution (CEP contribution in Figure 3-1).

Finally, improved capability of detecting multiple ongoing situations in the environment discovered the need of analysing relationships between ongoing notifications. This need resulted in the proposal of novel classification and combination principles of ongoing notifications contributing to alarm management field. As a result, the set of principles, methods and tools have been designed and developed for relevant information delivery in BMS.

4 Conclusions

4.1 Original Contributions

The following original contributions have been delivered with this dissertation:

- *Assessment of information needs in BM domain (Publication I)*: the field studies, developed prototype and initial usability tests have helped to understand actual information needs of potential user groups in BM domain and form additional set of requirements for relevant information delivery.

- *Knowledge modelling principles addressing information needs (Publication II)*: new knowledge modelling principles have been proposed addressing information needs and capturing qualitative contextual information of BM domain.

- *A novel hybrid approach for relevant information delivery in BM domain (Publications III and IV)*: the proposed knowledge models have decoupled data from application logic and have enabled simplified reconfiguration within monitoring systems for redirection and personalised delivery of information flows (Publication III). By combining these models at the configuration stage of the middleware and fusing this knowledge with runtime data from devices by means of the CEP engine, it is possible to track multiple diverse situations in the environment and deliver personalized information according to actual user needs (Publications IV).

- *A novel classification and combination principles of ongoing notifications in BM domain (Publication V)*: A novel approach for classifying and combining of ongoing notifications in BM domain has been proposed. The approach enables reduction of repeating information pieces, which are delivered as a part of multiple notifications. In addition, the major benefit of the proposed solution is its ability to work at runtime, thus addressing ongoing user needs.

4.2 Lessons Learned

One of the main lessons learned during this research work is an absolute need of a careful analysis of the problem and available technological solutions. The number of available technologies, frameworks, implementation libraries, and development environments is staggering and constantly growing with each new technology promising to be the golden key and addressing all needs. However, the promise to address it all and/or the desire to fit into buzzword trend often results in technology misuse, leading to solutions addressing very specific set of requirements and capable to work only under very limited conditions.

This research work attempts to address crucial requirements of information needs in BM domain and provides technological solution capable of satisfying these information needs and requirements of BMS. The proposal is based on careful estimation of available technological solutions resulting in hybrid approach combining the advantages of chosen technologies. This work is an interdisciplinary research that has been influenced by the following research areas:

- Ontologies and cognitive science;
- Complex event processing;
- Software architecture and design;
- Distributed control systems;
- Service-oriented architectures and Web Services

4.3 Potential Enhancements

The further enhancements are envisioned as a direct continuation of the work:

- *A formal validation of relevant information delivery with real users*: although preliminary usability tests have been conducted in the beginning of the research work to highlight actual user needs and requirements, the formal validation of the final solution is envisioned as a first immediate step. This work would include estimation of actual reactions of the operators to changed information flows and how this change affects decisions at runtime.
- *Distributing solution across multiple engines*: although hybrid approach has significantly improved the performance of the solution in comparison to knowledge-driven approach, it might be further improved by distributing runtime analysis capabilities across multiple engines and redirecting certain data streams to these engines.
- *Considering limitations of displaying devices*: mobile devices pose significant limitations on amount of displayed information. This work was focused on understanding user requirements, enabling flexible redirection of information flows, and improving analysis capabilities of data from devices. The next logical step is the consideration of display limitations for different device types and how it affects the delivered information.
- *Uncertainty handling*: in this work, it is assumed that data from devices is already pre-processed and validated. However, not all of the underlying control and monitoring systems have the sufficient means for data pre-processing. Thus, including this functionality and performing research in the field of data fusion and pre-processing techniques is the next logical step.

4.4 Future Directions

As a result of this research work, the following research directions can be proposed:

- *Adapting approach to other domains of control and monitoring systems*: the technological similarity between control and monitoring systems of other domains (such as discrete manufacturing and, to some degree, process automation) is growing. Thus, adaptation of this solution to other domains of control and monitoring systems is one of the rational steps for possible research directions.

- *Smart data processing*: the presented approach is based on such technologies as semantic knowledge models and Complex Event Processing. These techniques are multi-functional and can be adopted for processing of information for different purposes going beyond the relevant information delivery. The examples include activity detection, analytics on energy consumption, appliances usage, and behaviour analysis. In addition, the other techniques and technologies applied to the data of control and monitoring systems could be investigated resulting in proposal of best practices for processing such data for different purposes and extending functions of BMS to contributors of smart spaces.

- *Analysing actual capabilities of system users*: the present approach considers user needs specified based on their roles in the system and personal responsibilities. It would be very interesting to include analysis of actual perception capabilities and needs. For example, how stress and time pressure actually affect perception of individual users and what can be done about it; how it is better to present information to particular user with respect to his/her personal capabilities. For these purposes, the bridge between situation-aware research on modelling human perception mechanisms and context-aware systems analysing ongoing situations in the environment is seen as a future direction.

- Finally, the *development of standards for alarm management in BMS* is seen as a needed area of research: the process industry is well supported by alarm management standards, which bring clear vision of difference between different notifications in these systems and try to highlight best practices in the industry. However, these standards do not address specificity of BM systems and environments. Thus, there is a need in development of such alarm management standards, which would address needs of operators in BMS and summarize the best practices and approaches to alarm handling in this domain.

4.5 Concluding Remarks

The primary objective of this research work has been achieved by designing and developing the middleware capable of relevant information delivery in BMS. The middleware meets the qualitative requirements of the solution listed in the thesis and supports multi-role and multi-user BMS. In addition, the principles of relevant information delivery have been designed and delivered with this research work. The obtained results suggest feasibility of the proposed approach and serve as a basis for future research directions. Thus, the thesis contributes to the state of the art in technologies and practices for data and information management in control and monitoring systems within delimited scope.

References

[Adomavicius & Tuzhilin, 2005]

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

[Adomavicius & Tuzhilin, 2015]

Adomavicius, G., & Tuzhilin, A. (2015). Context-Aware Recommender Systems. In *Recommender Systems Handbook* (pp. 191–226). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4899-7637-6_6

[Alegre, Augusto, & Clark, 2016]

Alegre, U., Augusto, J. C., & Clark, T. (2016). Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, *117*, 55–83. https://doi.org/10.1016/j.jss.2016.02.010

[Alirezaie & Loutfi, 2015]

Alirezaie, M., & Loutfi, A. (2015). Reasoning for sensor data interpretation: An application to air quality monitoring. *Journal of Ambient Intelligence and Smart Environments*, *7*(4), 579–597. https://doi.org/10.3233/AIS-150323

[Almada-Lobo, 2017]

Almada-Lobo, F. (2017). Shop-Floor Decentralization: Shop Floor Marketplace Emerges. Retrieved June 29, 2017, from http://blog.mesa.org/2017/03/shop-floor-decentralization-shop-floor.html?m=1

[Aloulou, Mokhtari, Tiberghien, Endelin, & Biswas, 2015]

Aloulou, H., Mokhtari, M., Tiberghien, T., Endelin, R., & Biswas, J. (2015). Uncertainty handling in semantic reasoning for accurate context understanding. *Knowledge-Based Systems*, *77*, 16–28. https://doi.org/10.1016/j.knosys.2014.12.025

[Anicic, Fodor, Rudolph, & Stojanovic, 2011]

Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World wide web - WWW '11* (Vol. 43, pp. 635–644). https://doi.org/10.1145/1963405.1963495

[Arias Fisteus, Fernández García, Sánchez Fernández, & Fuentes-Lorenzo, 2014]

Arias Fisteus, J., Fernández García, N., Sánchez Fernández, L., & Fuentes-Lorenzo, D. (2014). Ztreamy: a middleware for publishing semantic streams on the Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *25*, 16–23. https://doi.org/10.1016/j.websem.2013.11.002

[Armand, Filliat, & Ibanez-Guzman, 2014]

Armand, A., Filliat, D., & Ibanez-Guzman, J. (2014). Ontology-based context awareness for driving assistance systems. In *2014 IEEE Intelligent Vehicles Symposium Proceedings* (pp. 227–233). IEEE. https://doi.org/10.1109/IVS.2014.6856509

[Bainbridge, 1983]
Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19*(6), 775–779. https://doi.org/10.1016/0005-1098(83)90046-8

[Bainbridge, 1997]
Bainbridge, L. (1997). The change in concepts needed to account for human behavior in complex dynamic tasks. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, *27*(3), 351–359. https://doi.org/10.1109/3468.568743

[Baldauf, Dustdar, & Rosenberg, 2007]
Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(4), 263–277. https://doi.org/10.1504/IJAHUC.2007.014070

[Banos, Damas, Pomares, Prieto, & Rojas, 2012]
Banos, O., Damas, M., Pomares, H., Prieto, A., & Rojas, I. (2012). Daily living activity recognition based on statistical feature quality group selection. *Expert Systems with Applications*, *39*(9), 8013–8021. https://doi.org/10.1016/j.eswa.2012.01.164

[Barbieri, Braga, Ceri, Valle, & Grossniklaus, 2010]
Barbieri, D. F., Braga, D., Ceri, S., Valle, E. Della, & Grossniklaus, M. (2010). C-SPARQL: a continuous query language for RDF data streams. *International Journal of Semantic Computing*, *4*(1), 3–25. https://doi.org/10.1142/S1793351X10000936

[Baumgartner, Gottesheim, Mitsch, Retschitzegger, & Schwinger, 2010]
Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., & Schwinger, W. (2010). BeAware! - Situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, *69*(11), 1181–1193. https://doi.org/10.1016/j.datak.2010.07.008

[Bettini et al., 2010]
Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, *6*(2), 161–180. https://doi.org/10.1016/j.pmcj.2009.06.002

[Bobadilla, Ortega, Hernando, & Gutiérrez, 2013]
Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

[Bolchini, Curino, Quintarelli, Schreiber, & Tanca, 2007]

Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., & Tanca, L. (2007). A data-oriented survey of context models. *ACM SIGMOD Record*, *36*(4), 19–26.

[Borgo, 2014]
Borgo, S. (2014). An ontological approach for reliable data integration in the industrial domain. *Computers in Industry*, *65*(9), 1242–1252. https://doi.org/10.1016/j.compind.2013.12.010

[Boubeta-Puig, Ortiz, & Medina-Bulo, 2014]
Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2014). A model-driven approach for facilitating user-friendly design of complex event patterns. *Expert Systems with Applications*, *41*(2), 445–456. https://doi.org/10.1016/j.eswa.2013.07.070

[Brewka, Eiter, & Truszczyński, 2011]
Brewka, G., Eiter, T., & Truszczyński, M. (2011). Answer set programming at a glance. *Communications of the ACM*, *54*(12), 92. https://doi.org/10.1145/2043174.2043195

[Cañas, Pacheco, Kemme, Kienzle, & Jacobsen, 2015]
Cañas, C., Pacheco, E., Kemme, B., Kienzle, J., & Jacobsen, H.-A. (2015). GraPS: a graph publish/sybscribe middleware. In *16th Annual Middleware Conference* (pp. 1–12). New York, New York, USA: ACM Press. https://doi.org/10.1145/2814576.2814812

[Chernbumroong, Cang, Atkins, & Yu, 2013]
Chernbumroong, S., Cang, S., Atkins, A., & Yu, H. (2013). Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, *40*(5), 1662–1674. https://doi.org/10.1016/j.eswa.2012.09.004

[Chiappe, Rorie, Morgan, & Vu, 2014]
Chiappe, D., Rorie, R. C., Morgan, C. A., & Vu, K.-P. L. (2014). A situated approach to the acquisition of shared SA in team contexts. *Theoretical Issues in Ergonomics Science*, *15*(1), 69–87. https://doi.org/10.1080/1463922X.2012.696739

[Colombo et al., 2014]
Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., … Martinez Lastra, J. L. (Eds.). (2014). *Industrial Cloud-Based Cyber-Physical Systems*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-05624-1

[Constantinou, Fenton, Marsh, & Radlinski, 2016]
Constantinou, A. C., Fenton, N., Marsh, W., & Radlinski, L. (2016). From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. *Artificial Intelligence in Medicine*, *67*, 75–93. https://doi.org/10.1016/j.artmed.2016.01.002

[Cugola & Margara, 2012]

Cugola, G., & Margara, A. (2012). Processing flows of information: from data stream to complex event processing. *ACM Computing Surveys*, *44*(3), 1–62. https://doi.org/10.1145/2187671.2187677

[Dejonghe, 2016]
Dejonghe, A. (2016). Towards Scalable Federated Context-Aware Stream Reasoning. In F. Gandon, M. Sabou, H. Sack, C. D'Amato, P. Cudré-Mauroux, & A. Zimmermann (Eds.) (Vol. 9088, pp. 803–812). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-34129-3_49

[Delamer, 2006]
Delamer, I. M. (2006). *Event-based middleware for reconfigurable manufacturing systems: a semantic Web services approach* (Tampereen teknillinen yliopisto. Julkaisu). Tampere University of Technology.

[Dey, 2001]
Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, *5*(1), 4–7. https://doi.org/10.1007/s007790170019

[Dey, Abowd, & Salber, 2001]
Dey, A. K., Abowd, G., & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, *16*(2), 97–166. https://doi.org/10.1207/S15327051HCI16234_02

[Díaz Rodríguez, Cuéllar, Lilius, & Delgado Calvo-Flores, 2014]
Díaz Rodríguez, N., Cuéllar, M. P., Lilius, J., & Delgado Calvo-Flores, M. (2014). A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, *66*, 46–60. https://doi.org/10.1016/j.knosys.2014.04.016

[Eckert, Bry, Brodt, Poppe, & Hausmann, 2011]
Eckert, M., Bry, F., Brodt, S., Poppe, O., & Hausmann, S. (2011). A CEP Babelfish: Languages for Complex Event Processing and Querying Surveyed. In *Reasoning in Event-Based Distributed Systems* (Vol. 347 SCI, pp. 47–70). https://doi.org/10.1007/978-3-642-19724-6_3

[Endsley, 1995]
Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

[Endsley, 2015]
Endsley, M. R. (2015). Situation Awareness Misconceptions and Misunderstandings. *Journal of Cognitive Engineering and Decision Making*, *9*(1), 4–32. https://doi.org/10.1177/1555343415572631

[Eppler & Mengis, 2004]
Eppler, M. J., & Mengis, J. (2004). The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS,

and Related Disciplines. *The Information Society*, *20*(5), 325–344. https://doi.org/10.1080/01972240490507974

[Esposito et al., 2008]
Esposito, A., Tarricone, L., Zappatore, M., Catarinucci, L., Colella, R., & Dibari, A. (2008). A framework for context-aware home-health monitoring. *Ubiquitous Intelligence and Computing*, *3*(1), 75–91. https://doi.org/10.1504/IJAACS.2010.030313

[Evchina, Dvoryanchikova, & Martinez Lastra, 2012]
Evchina, Y., Dvoryanchikova, A., & Martinez Lastra, J. L. (2012). Ontological framework of context-aware and reasoning middleware for smart homes with health and social services. In *IEEE International Conference on Systems, Man and Cybernetics* (pp. 985–990). Seoul. https://doi.org/10.1109/ICSMC.2012.6377857

[Evchina & Martinez Lastra, 2015]
Evchina, Y., & Martinez Lastra, J. L. (2015). Semantic-driven CEP for delivery of information streams in data-intensive monitoring systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)* (pp. 1251–1256). IEEE. https://doi.org/10.1109/INDIN.2015.7281914

[Fan, Xiao, & Yan, 2015]
Fan, C., Xiao, F., & Yan, C. (2015). A framework for knowledge discovery in massive building automation data and its application in building diagnostics. *Automation in Construction*, *50*(C), 81–90. https://doi.org/10.1016/j.autcon.2014.12.006

[Feng, Teng, & Tan, 2009]
Feng, Y.-H., Teng, T.-H., & Tan, A.-H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, *36*(1), 455–463. https://doi.org/10.1016/j.eswa.2007.09.061

[Fenza, Furno, & Loia, 2012]
Fenza, G., Furno, D., & Loia, V. (2012). Hybrid approach for context-aware service discovery in healthcare domain. *Journal of Computer and System Sciences*, *78*(4), 1232–1247. https://doi.org/10.1016/j.jcss.2011.10.011

[Fernández-López, Gómez-Pérez, & Suárez-Figueroa, 2013]
Fernández-López, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2013). Methodological guidelines for reusing general ontologies. *Data & Knowledge Engineering*, *86*, 242–275. https://doi.org/10.1016/j.datak.2013.03.006

[Forkan, Khalil, & Tari, 2014]
Forkan, A., Khalil, I., & Tari, Z. (2014). CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living. *Future Generation Computer Systems*, *35*, 114–127. https://doi.org/10.1016/j.future.2013.07.009

[Frost & Sullivan, 2011]
Frost & Sullivan. (2011). *European Building Automation systems market:*

paving the way for smart and energy efficient buildings. Retrieved from http://www.frost.com

[Frost & Sullivan, 2016a]
Frost & Sullivan. (2016a). *2016 Global Homes and Buildings Industry Outlook: Internet of Things in Smart Buildings to be the Biggest Driver of Growth*. Retrieved from http://www.frost.com

[Frost & Sullivan, 2016b]
Frost & Sullivan. (2016b). *Global Automation Industry Outlook, 2016: Growth to be Led by IIoT and New Service Models , Technology Focus on Managed Security Services and Advanced Predictive Analytics*. Retrieved from http://www.frost.com

[Furno & Zimeo, 2014]
Furno, A., & Zimeo, E. (2014). Context-aware composition of Semantic Web Services. *Mobile Networks and Applications*, *19*(2), 235–248. https://doi.org/10.1007/s11036-014-0494-y

[Gellrich, Lunkwitz, Dennert, & Kabitzsch, 2012]
Gellrich, A., Lunkwitz, D., Dennert, A., & Kabitzsch, K. (2012). Rule-driven manufacturing control based on ontologies. In *IEEE Emerging Technologies & Factory Automation* (pp. 1–8).

[Gonzalez, Chen, & Dahanayake, 2008]
Gonzalez, R. A., Chen, N., & Dahanayake, A. (Eds.). (2008). *Personalized Information Retrieval and Access: Concepts, Methods, and Practices*. IGI Global. Retrieved from http://common.books24x7.com/toc.aspx?bookid=18640

[Grasso, Iiritano, Lio, Ricca, & Scalise, 2010]
Grasso, G., Iiritano, S., Lio, V., Ricca, F., & Scalise, F. (2010). Team-building with answer set programming in the Gioia-Tauro seaport. In *25th Italian Conference on Computational Logic* (pp. 1–5).

[Guillame-Bert & Crowley, 2012]
Guillame-Bert, M., & Crowley, J. L. (2012). Learning temporal association rules on symbolic time sequences. In *Proceedings of the 4th Asian Conference on Machine Learning (ACML '12)* (Vol. 25, pp. 159–174). Retrieved from http://jmlr.csail.mit.edu/proceedings/papers/v25/guillame-bert12.html

[Gupta & Gadge, 2015]
Gupta, J., & Gadge, J. (2015). Performance analysis of recommendation system based on collaborative filtering and demographics. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCICT.2015.7045675

[Han, Lee, & Crespi, 2014]
Han, S. N., Lee, G. M., & Crespi, N. (2014). Semantic context-aware service composition for building automation system. *IEEE Transactions on Industrial Informatics*, *10*(1), 752–761.

[Hoareau & Satoh, 2009]

Hoareau, C., & Satoh, I. (2009). Modeling and processing information for context-aware computing: a survey. *New Generation Computing*, *27*(3), 177–196. https://doi.org/10.1007/s00354-009-0060-5

[Hobbs & Pan, 2004]

Hobbs, J. R., & Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP) - Special Issue on Temporal Information Processing*, *3*(1), 66–85. https://doi.org/10.1145/1017068.1017073

[Horrocks, Li, Turi, & Bechhofer, 2004]

Horrocks, I., Li, L., Turi, D., & Bechhofer, S. (2004). The Instance Store: DL reasoning with large numbers of individuals. In *Description Logic Workshop* (pp. 31–40). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.5326

[Jae, Boreom, & Kwang, 2011]

Jae, H. S., Boreom, L., & Kwang, S. P. (2011). Detection of Abnormal Living Patterns for Elderly Living Alone Using Support Vector Data Description. *IEEE Transactions on Information Technology in Biomedicine*, *15*(3), 438–448. https://doi.org/10.1109/TITB.2011.2113352

[Jardim-Goncalves, Coutinho, Cretan, da Silva, & Ghodous, 2014]

Jardim-Goncalves, R., Coutinho, C., Cretan, A., da Silva, C. F., & Ghodous, P. (2014). Collaborative negotiation for ontology-driven enterprise businesses. *Computers in Industry*, *65*(9), 1232–1241. https://doi.org/10.1016/j.compind.2014.01.001

[Johnson Controls, 2015]

Johnson Controls. (2015). *BAS delivers a new kind of smart*. Retrieved from http://www.johnsoncontrols.com/content/dam/WWW/jci/be/building_management/metasys_building_automation/MetasysProductPaper_010715.pdf

[Joho, Jatowt, & Blanco, 2015]

Joho, H., Jatowt, A., & Blanco, R. (2015). Temporal information searching behaviour and strategies. *Information Processing & Management*, *51*(6), 834–850. https://doi.org/10.1016/j.ipm.2015.03.006

[Josefiok, Krahn, & Sauer, 2015]

Josefiok, M., Krahn, T., & Sauer, J. (2015). A Survey on Expert Systems for Diagnosis Support in the Field of Neurology. In R. Neves-Silva, L. C. Jain, & R. J. Howlett (Eds.), *Smart Innovation, Systems and Technologies* (Vol. 39, pp. 291–300). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-19857-6_26

[Kaber, Riley, Tan, & Endsley, 2001]

Kaber, D. B., Riley, J. M., Tan, K.-W., & Endsley, M. R. (2001). On the Design of Adaptive Automation for Complex Systems. *International Journal of Cognitive Ergonomics*, *5*(1), 37–57. https://doi.org/10.1207/S15327566IJCE0501_3

[Kang, Li, & Krishnaswamy, 2012]
Kang, Y.-B., Li, Y.-F., & Krishnaswamy, S. (2012). Predicting reasoning performance using ontology metrics. In *The Semantic Web – ISWC 2012* (pp. 198–214). https://doi.org/10.1007/978-3-642-35176-1_13

[Khilwani, Harding, & Choudhary, 2009]
Khilwani, N., Harding, J. A., & Choudhary, A. K. (2009). Semantic web in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *223*(7), 905–924. https://doi.org/10.1243/09544054JEM1399

[Kokar, Matheus, & Baclawski, 2009]
Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. *Information Fusion*, *10*(1), 83–98. https://doi.org/10.1016/j.inffus.2007.01.004

[Kolbe, Zaslavsky, Kubler, Robert, & Le Traon, 2017]
Kolbe, N., Zaslavsky, A., Kubler, S., Robert, J., & Le Traon, Y. (2017). Enriching a Situation Awareness Framework for IoT with Knowledge Base and Reasoning Components. In *Lecture Notes in Computer Science* (Vol. 3554/2005, pp. 41–54). https://doi.org/10.1007/978-3-319-57837-8_4

[Komazec, Cerri, & Fensel, 2012]
Komazec, S., Cerri, D., & Fensel, D. (2012). Sparkwave. In *6th ACM International Conference on Distributed Event-Based Systems - DEBS '12* (pp. 58–68). New York, New York, USA: ACM Press. https://doi.org/10.1145/2335484.2335491

[Kranjc et al., 2015]
Kranjc, J., Smailovic, J., Podpecan, V., Grcar, M., Znidarsic, M., & Lavrac, N. (2015). Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the ClowdFlows platform. *Information Processing & Management*, *51*(2), 187–203. https://doi.org/10.1016/j.ipm.2014.04.001

[Krieger, 2010]
Krieger, H.-U. (2010). A general methodology for equipping ontologies with time. In *International conference on Language Resources and Evaluation* (pp. 3165–3172).

[Larson, 2014]
Larson, K. (2014). Operator effectiveness: the next frontier of process automation. Retrieved March 6, 2015, from https://www.yumpu.com/en/document/view/36262918/operator-effectiveness-the-next-frontier-of-control-global

[Le-Phuoc, Dao-Tran, et al., 2012]
Le-Phuoc, D., Dao-Tran, M., Pham, M., Boncz, P., Eiter, T., & Fink, M. (2012). Linked Stream Data Processing Engines: Facts and Figures. In *The Semantic Web – ISWC 2012. ISWC 2012. Lecture Notes in Computer Science* (Vol. 7650,

pp. 300–312). https://doi.org/10.1007/978-3-642-35173-0_20

[Le-Phuoc, Nguyen-Mau, Parreira, & Hauswirth, 2012]
Le-Phuoc, D., Nguyen-Mau, H. Q., Parreira, J. X., & Hauswirth, M. (2012). A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, *16*, 42–51. https://doi.org/10.1016/j.websem.2012.06.003

[Le-Phuoc, Parreira, & Hauswirth, 2012]
Le-Phuoc, D., Parreira, J. X., & Hauswirth, M. (2012). Linked Stream Data processing. *Reasoning Web. Semantic Technologies for Advanced Query Answering*, *7487 LNCS*, 245–289.

[J. Lee, Bagheri, & Kao, 2015]
Lee, J., Bagheri, B., & Kao, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

[J. D. Lee & Kirlik, 2013]
Lee, J. D., & Kirlik, A. (Eds.). (2013). *The Oxford Handbook of Cognitive Engineering*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199757183.001.0001

[J. H. Lee, Lee, Kim, Wang, & Love, 2014]
Lee, J. H., Lee, H., Kim, M. J., Wang, X., & Love, P. E. D. (2014). Context-aware inference in ubiquitous residential environments. *Computers in Industry*, *65*(1), 148–157. https://doi.org/10.1016/j.compind.2013.08.005

[Leitao, Colombo, & Karnouskos, 2016]
Leitao, P., Colombo, A. W., & Karnouskos, S. (2016). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, *81*, 11–25. https://doi.org/10.1016/j.compind.2015.08.004

[Lopera Gonzalez & Amft, 2016]
Lopera Gonzalez, L. I., & Amft, O. (2016). Mining hierarchical relations in building management variables. *Pervasive and Mobile Computing*, *26*, 91–101. https://doi.org/10.1016/j.pmcj.2015.10.009

[Lotfi, Langensiepen, Mahmoud, & Akhlaghinia, 2012]
Lotfi, A., Langensiepen, C., Mahmoud, S. M., & Akhlaghinia, M. J. (2012). Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of Ambient Intelligence and Humanized Computing*, *3*(3), 205–218. https://doi.org/10.1007/s12652-010-0043-x

[Luckham, 2002]
Luckham, D. (2002). The power of events: an introduction to Complex Event Processing in distributed enterprise systems. In *Rule Representation, Interchange and Reasoning on the Web* (p. 3). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-88808-6_2

[Luokkala & Virrantaus, 2014]
Luokkala, P., & Virrantaus, K. (2014). Developing information systems to support situational awareness and interaction in time-pressuring crisis situations. *Safety Science*, *63*, 191–203. https://doi.org/10.1016/j.ssci.2013.11.014

[Margara, Urbani, Van Harmelen, & Bal, 2014]
Margara, A., Urbani, J., Van Harmelen, F., & Bal, H. (2014). Streaming the Web: Reasoning over dynamic data. *Journal of Web Semantics*. Elsevier B.V. https://doi.org/10.1016/j.websem.2014.02.001

[Martinez Lastra, 2004]
Martinez Lastra, J. L. (2004). *Reference Mechatronic Architecture for Actor-based Assembly Systems* (Tampereen teknillinen yliopisto, Julkaisuja ; Vol. 484). Tampere University of Technology.

[Mayer, Verborgh, Kovatsch, & Mattern, 2016]
Mayer, S., Verborgh, R., Kovatsch, M., & Mattern, F. (2016). Smart Configuration of Smart Environments. *IEEE Transactions on Automation Science and Engineering*, *13*(3), 1247–1255. https://doi.org/10.1109/TASE.2016.2533321

[Meissen, Pfennigschmidt, Voisard, & Wahnfried, 2005]
Meissen, U., Pfennigschmidt, S., Voisard, A., & Wahnfried, T. (2005). Context- and Situation-Awareness in Information Logistics. In *Current Trends in Database Technology - EDBT 2004 Workshops* (pp. 335–344).

[Mileo, Abdelrahman, Policarpio, & Hauswirth, 2013]
Mileo, A., Abdelrahman, A., Policarpio, S., & Hauswirth, M. (2013). StreamRule: A Nonmonotonic Stream Reasoning System for the Semantic Web. In *Web Reasoning and Rule Systems* (pp. 247–252). https://doi.org/10.1007/978-3-642-39666-3_23

[Miller, Nagy, & Schlueter, 2015]
Miller, C., Nagy, Z., & Schlueter, A. (2015). Automated daily pattern filtering of measured building performance data. *Automation in Construction*, *49*(PA), 1–17. https://doi.org/10.1016/j.autcon.2014.09.004

[Morris & Kulvatunyou, 2017]
Morris, K., & Kulvatunyou, S. (2017). Working Towards an Industrial Ontology Foundry to Facilitate Interoperability. Retrieved September 28, 2017, from http://blog.mesa.org/2017/03/working-towards-industrial-ontology.html?m=1

[Mühl, Parzyjegla, & Prellwitz, 2015]
Mühl, G., Parzyjegla, H., & Prellwitz, M. (2015). Analyzing content-based publish/subscribe systems. In *9th ACM International Conference on Distributed Event-Based Systems* (pp. 128–139). New York, New York, USA: ACM Press. https://doi.org/10.1145/2675743.2771836

[Nacer & Aissani, 2014]

Nacer, H., & Aissani, D. (2014). Semantic web services: Standards, applications, challenges and solutions. *Journal of Network and Computer Applications*, *44*, 134–151. https://doi.org/10.1016/j.jnca.2014.04.015

[Nadoveza & Kiritsis, 2014]

Nadoveza, D., & Kiritsis, D. (2014). Ontology-based approach for context modeling in enterprise applications. *Computers in Industry*, *65*(9), 1218–1231. https://doi.org/10.1016/j.compind.2014.07.007

[Nazerfard & Cook, 2015]

Nazerfard, E., & Cook, D. J. (2015). CRAFFT: an activity prediction model based on Bayesian networks. *Journal of Ambient Intelligence and Humanized Computing*, *6*(2), 193–205. https://doi.org/10.1007/s12652-014-0219-x

[Nguyen & Aiello, 2013]

Nguyen, T. A., & Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. *Energy and Buildings*, *56*, 244–257. https://doi.org/10.1016/j.enbuild.2012.09.005

[Nieto Lee, Evchina, Dvoryanchikova, & Martinez Lastra, 2013]

Nieto Lee, A. N., Evchina, Y., Dvoryanchikova, A., & Martinez Lastra, J. L. (2013). Pro-active content managing system for efficient human machine interaction in data intensive environments. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)* (pp. 790–796). IEEE. https://doi.org/10.1109/INDIN.2013.6622985

[Nieto Lee, Gonzalez Moctezuma, & Martinez Lastra, 2013]

Nieto Lee, A. N., Gonzalez Moctezuma, L. E., & Martinez Lastra, J. L. (2013). Visualization of Information in a Service-Oriented Production Control System. In *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society* (pp. 4422–4428). IEEE. https://doi.org/10.1109/IECON.2013.6699847

[Noor, Salcic, & Wang, 2016]

Noor, M. H. M., Salcic, Z., & Wang, K. I. K. (2016). Enhancing ontological reasoning with uncertainty handling for activity recognition. *Knowledge-Based Systems*, *114*, 47–60. https://doi.org/10.1016/j.knosys.2016.09.028

[O’Connor & Das, 2011]

O’Connor, M. J., & Das, A. K. (2011). A Method for Representing and Querying Temporal Information in OWL. In *Biomedical Engineering Systems and Technologies* (Vol. 127 CCIS, pp. 97–110). https://doi.org/10.1007/978-3-642-18472-7_8

[O’Hara & Hall, 1991]

O’Hara, J. M., & Hall, R. E. (1991). Advanced control rooms and crew performance issues: implications for human reliability. *IEEE Transactions on Nuclear Science*, *39*(4), 919–923. https://doi.org/10.1109/NSSMIC.1991.258809

[Okeyo, Chen, & Wang, 2014]

Okeyo, G., Chen, L., & Wang, H. (2014). Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*, *39*, 29–43. https://doi.org/10.1016/j.future.2014.02.014

[Okeyo, Chen, Wang, & Sterritt, 2014]

Okeyo, G., Chen, L., Wang, H., & Sterritt, R. (2014). Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, *10*(PART B), 155–172. https://doi.org/10.1016/j.pmcj.2012.11.004

[Ongenae et al., 2013]

Ongenae, F., Claeys, M., Dupont, T., Kerckhove, W., Verhoeve, P., Dhaene, T., & De Turck, F. (2013). A probabilistic ontology-based platform for self-learning context-aware healthcare applications. *Expert Systems with Applications*, *40*(18), 7629–7646. https://doi.org/10.1016/j.eswa.2013.07.038

[Onisko, Lucas, & Druzdzel, 2001]

Onisko, A., Lucas, P. J. F., & Druzdzel, M. J. (2001). Comparison of Rule-Based and Bayesian Network Approaches in Medical Diagnostic Systems. In S. Quaglini, P. Barahona, & S. Andreassen (Eds.), *8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine* (Vol. LNAI 2101, pp. 283–292). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48229-6

[Perera, Zaslavsky, Christen, & Georgakopoulos, 2014]

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, *16*(1), 414–454. https://doi.org/10.1109/SURV.2013.042313.00197

[Pham, 2015]

Pham, T.-L. (2015). A Scalable Adaptive Method for Complex Reasoning Over Semantic Data Streams. In F. Gandon, M. Sabou, H. Sack, C. D'Amato, P. Cudré-Mauroux, & A. Zimmermann (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9088, pp. 751–759). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-18818-8_48

[Puttonen, 2014]

Puttonen, J. (2014). *A Semantically Enhanced Approach for Orchestration of Web Services in Factory Automation Systems* (Tampere University of Technology. Publication ; Vol. 1224). Tampere University of Technology.

[Puttonen, Lobov, Cavia Soto, & Martinez Lastra, 2010]

Puttonen, J., Lobov, A., Cavia Soto, M. A., & Martinez Lastra, J. L. (2010). A Semantic Web Services-based approach for production systems control. *Advanced Engineering Informatics*, *24*(3), 285–299. https://doi.org/10.1016/j.aei.2010.05.012

[Puttonen, Lobov, Cavia Soto, & Martinez Lastra, 2015]

Puttonen, J., Lobov, A., Cavia Soto, M. A., & Martinez Lastra, J. L. (2015). Planning-based semantic web service composition in factory automation. *Advanced Engineering Informatics*, *29*(4), 1041–1054. https://doi.org/10.1016/j.aei.2015.08.002

[Ray, 2013]

Ray, A. (2013). Autonomous perception and decision-making in cyber-physical systems. In *2013 8th International Conference on Computer Science & Education* (pp. 1–10). IEEE. https://doi.org/10.1109/ICCSE.2013.6554173

[Ren, Khrouf, Kazi-Aoul, Chabchoub, & Curé, 2016]

Ren, X., Khrouf, H., Kazi-Aoul, Z., Chabchoub, Y., & Curé, O. (2016). On measuring performances of C-SPARQL and CQELS. In *CEUR Workshop Proceedings* (Vol. 1783, pp. 1–12). Retrieved from http://arxiv.org/abs/1611.08269

[Ricci, Rokach, & Shapira, 2011]

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook* (Vol. 532, pp. 1–35). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-85820-3_1

[Rico, Caliusco, Chiotti, & Galli, 2014]

Rico, M., Caliusco, M. L., Chiotti, O., & Galli, M. R. (2014). OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems. *Computers in Industry*, *65*(9), 1291–1300. https://doi.org/10.1016/j.compind.2014.07.010

[Rodriguez, Bravo, & Guzman, 2012]

Rodriguez, J., Bravo, M., & Guzman, R. (2012). Multi-dimensional ontology model to support context-aware systems. In *International Conference on Internet and Web Applications and Services* (pp. 80–89).

[Roggen et al., 2013]

Roggen, D., Troster, G., Lukowicz, P., Ferscha, A., Millan, J., & Chavarriaga, R. (2013). Opportunistic human activity and context recognition. *Computer*, *46*(2), 36–45. https://doi.org/10.1109/MC.2012.393

[Russell & Norvig, 2010]

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (Third Edit). New Jersey: Pearson Education Inc.

[Russello, Mostarda, & Dulay, 2011]

Russello, G., Mostarda, L., & Dulay, N. (2011). A policy-based publish/subscribe middleware for sense-and-react applications. *Journal of Systems and Software*, *84*(4), 638–654. https://doi.org/10.1016/j.jss.2010.10.023

[Ruta, Scioscia, Loseto, & Di Sciascio, 2014]

Ruta, M., Scioscia, F., Loseto, G., & Di Sciascio, E. (2014). Semantic-based resource discovery and orchestration in Home and Building Automation: a multi-agent approach. *IEEE Transactions on Industrial Informatics*, *10*(1), 730–741. https://doi.org/10.1109/TII.2013.2273433

[Salmon et al., 2009]
Salmon, P. M., Stanton, N. A., Walker, G. H., Jenkins, D., Ladva, D., Rafferty, L., & Young, M. (2009). Measuring Situation Awareness in complex systems: Comparison of measures study. *International Journal of Industrial Ergonomics*, *39*(3), 490–500. https://doi.org/10.1016/j.ergon.2008.10.010

[Santofimia, Fahlman, del Toro, Moya, & Lopez, 2011]
Santofimia, M. J., Fahlman, S. E., del Toro, X., Moya, F., & Lopez, J. C. (2011). A semantic model for actions and events in ambient intelligence. *Engineering Applications of Artificial Intelligence*, *24*(8), 1432–1445. https://doi.org/10.1016/j.engappai.2011.05.008

[Sauter, Soucek, Kastner, & Dietrich, 2011]
Sauter, T., Soucek, S., Kastner, W., & Dietrich, D. (2011). The evolution of factory and building automation. *IEEE Industrial Electronics Magazine*, *5*(3), 35–48. https://doi.org/10.1109/MIE.2011.942175

[Schmidt, 2014]
Schmidt, A. (2014). Context-aware computing: context-awareness, context-aware user interfaces, and implicit interaction. In M. Soegaard & R. F. Dam (Eds.), *The Encyclopedia of Human-Computer Interaction* (2nd ed.). The Interaction Design Foundation. Retrieved from http://www.interaction-design.org/encyclopedia/context-aware_computing.html

[Seeger, Laerhoven, Sauer, & Buchmann, 2013]
Seeger, C., Laerhoven, K. Van, Sauer, J., & Buchmann, A. (2013). A Publish/Subscribe middleware for body and ambient sensor networks that mediates between sensors and applications. In *2013 IEEE International Conference on Healthcare Informatics* (pp. 199–208). IEEE. https://doi.org/10.1109/ICHI.2013.30

[Seong, 2009]
Seong, P. H. (Ed.). (2009). *Reliability and Risk Issues in Large Scale Safety-critical Digital Control Systems*. Springer. https://doi.org/10.1007/978-1-84800-384-2

[Sequeda & Corcho, 2009]
Sequeda, J. F., & Corcho, O. (2009). Linked Stream Data: a position paper. In *2nd International Conference on Semantic Sensor Networks* (Vol. 522, pp. 148–157).

[Shi, Deng, & Qin, 2011]
Shi, K., Deng, Z., & Qin, X. (2011). TinyMQ : a content-based publish/subscribe middleware for Wireless Sensor Networks. In *Sensor Technologies and Applications* (pp. 12–17).

[Skoczylas, 2014]
Skoczylas, N. (2014). Estimating gas and rock outburst risk on the basis of knowledge and experience - the expert system based on fuzzy logic. *Archives of Mining Sciences*, *59*(1), 41–52. https://doi.org/10.2478/amsc-2014-0003

[Smith & Bianchini, 2014]
Smith, F., & Bianchini, D. (2014). Selection, ranking and composition of semantically enriched business processes. *Computers in Industry*, *65*(9), 1253–1263. https://doi.org/10.1016/j.compind.2014.07.009

[Soylu, De Causmaecker, & Desmet, 2009]
Soylu, A., De Causmaecker, P., & Desmet, P. (2009). Context and adaptivity in pervasive computing environments: links with software engineering and ontological engineering. *Journal of Software*, *4*(9), 992–1013. https://doi.org/10.4304/jsw.4.9.921-1013

[Steffen & Schweizer, 2014]
Steffen, H., & Schweizer, L. (2014). Answer Set Programming and CLASP A Tutorial. In *Young Scientists' International Workshop on Trends in Information Processing (YSIP)* (pp. 77–95).

[Strang & Linnhoff-Popien, 2004]
Strang, T., & Linnhoff-Popien, C. (2004). A Context Modeling Survey. In *First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*.

[Sun, Wu, & Pan, 2009]
Sun, J., Wu, Z., & Pan, G. (2009). Context-aware smart car: from model to prototype. *Journal of Zhejiang University-SCIENCE A*, *10*(7), 1049–1059. https://doi.org/10.1631/jzus.A0820154

[Symeonidis, Ntempos, & Manolopoulos, 2014]
Symeonidis, P., Ntempos, D., & Manolopoulos, Y. (2014). *Recommender Systems for Location-based Social Networks*. Springer. Retrieved from http://common.books24x7.com/toc.aspx?bookid=76691

[Taylor & Leidinger, 2011]
Taylor, K., & Leidinger, L. (2011). Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. In *The Semantic Web: Research and Applications* (Vol. 6644 LNCS, pp. 285–299). https://doi.org/10.1007/978-3-642-21064-8_20

[Terroso-Saenz, Valdes-Vela, Campuzano, Botia, & Skarmeta-Gomez, 2015]
Terroso-Saenz, F., Valdes-Vela, M., Campuzano, F., Botia, J. A., & Skarmeta-Gomez, A. F. (2015). A complex event processing approach to perceive the vehicular context. *Information Fusion*, *21*(1), 187–209. https://doi.org/10.1016/j.inffus.2012.08.008

[Uddin, Puttonen, Scholze, Dvoryanchikova, & Martinez Lastra, 2012]
Uddin, M. K., Puttonen, J., Scholze, S., Dvoryanchikova, A., & Martinez Lastra, J.

L. (2012). Ontology-based context-sensitive computing for FMS optimization. *Assembly Automation*, *32*(2), 163–174. https://doi.org/10.1108/01445151211212316

[Urbieta, González-Beltrán, Ben Mokhtar, Anwar Hossain, & Capra, 2017]
Urbieta, A., González-Beltrán, A., Ben Mokhtar, S., Anwar Hossain, M., & Capra, L. (2017). Adaptive and context-aware service composition for IoT-based smart cities. *Future Generation Computer Systems*, *76*, 262–274. https://doi.org/10.1016/j.future.2016.12.038

[Wan, O'Grady, & O'Hare, 2015]
Wan, J., O'Grady, M. J., & O'Hare, G. M. P. (2015). Dynamic sensor event segmentation for real-time activity recognition in a smart home context. *Personal and Ubiquitous Computing*, *19*(2), 287–301. https://doi.org/10.1007/s00779-014-0824-x

[Weiss, Mandl, & Schill, 2011]
Weiss, J., Mandl, P., & Schill, A. (2011). Functional testing of Complex Event Processing applications. In *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing* (pp. 407–413). IEEE. https://doi.org/10.1109/ICCP.2011.6047906

[Wu, Kao, Wu, & Huang, 2015]
Wu, C., Kao, S.-C., Wu, C.-C., & Huang, S. (2015). Location-aware service applied to mobile short message advertising: Design, development, and evaluation. *Information Processing & Management*, *51*(5), 625–642. https://doi.org/10.1016/j.ipm.2015.06.001

[Xiao & Fan, 2014]
Xiao, F., & Fan, C. (2014). Data mining in building automation system for improving building operational performance. *Energy and Buildings*, *75*, 109–118. https://doi.org/10.1016/j.enbuild.2014.02.005

[Xin & Zhang, 2014]
Xin, J., & Zhang, J. (2014). Application of Complex Event Processing in the Express Business Automation. *International Journal of U- and E-Service, Science and Technology*, *7*(2), 275–286. https://doi.org/10.14257/ijunesst.2014.7.2.25

[Xue, Chang, & Liu, 2014]
Xue, X., Chang, J., & Liu, Z. (2014). Context-aware intelligent service system for coal mine industry. *Computers in Industry*, *65*(2), 291–305. https://doi.org/10.1016/j.compind.2013.11.010

[Yang, Fu, Chen, Xu, & Yang, 2016]
Yang, Y., Fu, C., Chen, Y.-W., Xu, D.-L., & Yang, S.-L. (2016). A belief rule based expert system for predicting consumer preference in new product development. *Knowledge-Based Systems*, *94*, 105–113. https://doi.org/10.1016/j.knosys.2015.11.012

[Yao, Chu, & Li, 2011]

Yao, W., Chu, C.-H., & Li, Z. (2011). Leveraging complex event processing for smart hospitals using RFID. *Journal of Network and Computer Applications*, *34*(3), 799–810. https://doi.org/10.1016/j.jnca.2010.04.020

[Ye, Dasiopoulou, et al., 2015]

Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Kontopoulos, E., Kompatsiaris, I., & Dobson, S. (2015). Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing*, *23*, 1–25. https://doi.org/10.1016/j.pmcj.2014.12.009

[Ye, Stevenson, & Dobson, 2015]

Ye, J., Stevenson, G., & Dobson, S. (2015). KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, *19*, 47–70. https://doi.org/10.1016/j.pmcj.2014.02.003

[Yu, Haghighat, Fung, Morofsky, & Yoshino, 2011]

Yu, Z. (Jerry), Haghighat, F., Fung, B. C. M., Morofsky, E., & Yoshino, H. (2011). A methodology for identifying and improving occupant behavior in residential buildings. *Energy*, *36*(11), 6596–6608. https://doi.org/10.1016/j.energy.2011.09.002

[Zappia, Paganelli, & Parlanti, 2012]

Zappia, I., Paganelli, F., & Parlanti, D. (2012). A lightweight and extensible Complex Event Processing system for sense and respond applications. *Expert Systems with Applications*, *39*(12), 10408–10419. https://doi.org/10.1016/j.eswa.2012.01.197

[Zhao, Liu, & Lin, 2012]

Zhao, X., Liu, C., & Lin, T. (2012). Incorporating business logics into RFID-enabled applications. *Information Processing & Management*, *48*(1), 47–62. https://doi.org/10.1016/j.ipm.2011.02.004

Original Papers

Publication I

Evchina, Y., & Martinez Lastra, J. L. (2016). An ICT-driven hybrid automation system for elderly care support: a rehabilitation facility study case. *Journal of Housing For the Elderly*, *30*(1), 52–74. https://doi.org/10.1080/02763893.2015.1129382

An ICT-driven Hybrid Automation System for Elderly Care Support: A Rehabilitation Facility Study Case

The demographic statistics in developed countries suggest growth in the elderly segment of the population. At the same time, other studies forecast a shortage of nurses, increasing the pressure on hospitals to provide treatment for longer periods of time. These trends suggest a need for new ways of taking care of elderly population that support safe, comfortable and independent living. Meanwhile, prominent advances in information and communications technology (ICT) have enabled new systems that address various needs of the elderly. This paper presents an automation system combining ambient assisted living (AAL) and building automation (BA) system functionalities. The paper introduces a case study of a rehabilitation facility situated in Tampere, Finland. It details a field study summarizing the needs of the users, describes functional scenarios supported by the system, and reports the results of the first usability tests suggesting acceptance of a new technology by residents and care-giving personnel. The unique combination of ambient assisted living and building automation offers a safer and more comfortable environment for the elderly as well as helps caregivers on site in managing their workloads.

KEYWORDS building automation systems, ambient assisted living, information and communications technology (ICT), elderly care, rehabilitation facility

INTRODUCTION

The projections of demographic statistics indicate a growth in the elderly segment of the population throughout all developed countries (Andersson, Lindahl, & Malmqvist, 2011), ("Frost & Sullivan," 2012). In Finland, the percentage of population over 65 years of age is forecast to increase from 17.5% in 2010 to 28.2% in 2060 ("Official Statistics of Finland (OSF)," 2012). These numbers indicate an increased need for elderly care in the near future, resulting in higher demands for care services such as residential care homes or home assistance. On the other hand, the total number of caregivers has remained at almost the same level during the last decade ("National Institute for Health and Welfare," 2010), resulting in a "*huge shortage of nurses by 2020*" ("Frost & Sullivan," 2012). The gap between the amount of required services and capabilities of service providers creates a need for new ways of taking care of the elderly population that provide both (1) high-quality care in a safe environment and (2) a reduction of unnecessary workload for the caregivers on the site. This paper addresses the problem by proposing a hybrid automation system that combines the functionalities of ambient assisted living (AAL) and building automation (BA) systems. This combination of functionalities facilitates management of the facility for caregivers and provides a safe and comfortable environment for elderly.

Ambient assisted living (AAL) and building automation (BA) systems are traditionally seen as having different application areas and scope. AAL addresses the special needs of the elderly population by providing a safer and more comfortable environment through the use of ICT (Ruyter & Pelgrim, 2007). The AAL concept incorporates a large set of systems, including smart homes (Sun, Florio, Gui, & Blondia, 2009), (Demiris & Hensel, 2009), remote patient monitoring ("Frost & Sullivan," 2013), and telecare ("Frost & Sullivan," 2014). The aim of AAL systems is to stimulate and extend an active, independent lifestyle for elderly people.

On the other hand, BA systems are concerned with automation and monitoring of the environment in commercial, industrial, or institutional facilities. The purpose is to ensure operational performance, reduce energy consumption of buildings, and increase the comfort and safety of building occupants (Sauter, Soucek, Kastner, & Dietrich, 2011). The functionalities of BA systems include control and monitoring of aspects of the building environment, such as lighting, HVAC (heating, ventilation, and air conditioning), humidity control, energy management, fire and flood safety, and security ("Frost & Sullivan," 2011).

While AAL systems address the needs of senior citizens, increasing their comfort and level of independence, BA solutions are able to deliver products facilitating management of the whole facility, thus ensuring a safe environment and cheaper maintenance costs. This paper presents a hybrid automation system that combines functionalities of AAL and BA solutions. The solution relies on modern information and communication technologies (ICT) enabling new approaches for care support. The major benefit of this approach is in addressing specific needs of different user groups of care facilities: care-giving and administrative personnel as well as the elderly clients. The prototype has gone through several phases of development, beginning with a field study and continuing through three iterations of preliminary usability tests in the considered rehabilitation facility. The main contributions of the paper are twofold: (1) to introduce the reader to the advances of ICT in relation to AAL and BA systems; and (2) to present by means of a developed prototype the possible usage of ICT in elderly care support. The benefits of the hybrid automation system are demonstrated through supported functionality and the results of the preliminary usability tests, which suggest acceptance of the technology by residents and caregivers of the facility.

The paper is organized as follows. First, the main tendencies of contemporary AAL and BA systems are analyzed with an emphasis on the possibility of integration and the resulting advantages of the hybrid automation system. Second, approaches and methods taken during the prototype development are presented. Third, the rehabilitation facility is described and information needs of user groups from the facility are presented, showing that these needs could not be addressed by one type of system (AAL or BA) alone. Next, scenarios involving support by means of the hybrid automation system demonstrating a combination of AAL and BA functionalities are described. Finally, findings from preliminary usability tests are presented.

ICT-DRIVEN TRENDS IN ELDERLY CARE AND BUILDING AUTOMATION

Recent ICT advances have affected almost every aspect of our lives. Now there is the possibility of interconnection between devices, people, and various objects anywhere and at any time. Technical systems have become large, distributed, and open to extension and integration between each other. IP-based networks have enabled remote control and monitoring of large facilities and industrial sites (Sauter et al., 2011). New paradigms such as the Internet of Things (IoT) have emerged and claim to enable connection, cooperation and accessibility to any objects throughout the globe (Atzori, Iera, & Morabito, 2010).

Within the scope of elderly care, ICT solutions have improved many aspects, including health care support through remote monitoring and consultations, possibilities of social interaction with family and friends, and the enhanced safety and comfort of the residential environment (Bobillier Chaumon, Michel, Tarpin Bernard, & Croisile, 2013), (Helal et al., 2005). Further advances are seen in providing integrated services and infrastructure by means of combined systems (Coomans, Smet, & Heylighen, 2011; O'Grady, Muldoon, Dragone, Tynan, & O'Hare, 2009; Rocha et al., 2013). This section analyzes trends in AAL and BA systems from a technological point of view. The main purposes of the section are to introduce the reader to recent ICT advances in relation to these systems, present opportunities created by ICT, and highlight the potential benefits of the proposed hybrid automation system.

Ambient Assisted Living Systems

Assisted Living (AL) has been a hot topic of research during the last decade, driven by the aging population, the rising demand for personalized care, and the resulting new market opportunities ("Frost & Sullivan," 2010). Usage of ICT has played a crucial role in the evolution of AL systems. The observed shift is from face-to-face care to support in which the elderly are given more independence, a shift enabled by remote services such as telehealth and telecare (While & Dewsbury, 2011). A present trend is the usage of Ambient Intelligence (AmI) concept enabling environments sensitive to the presence of people and specific needs of the elderly (Sadri, 2011). A major milestone for Europe is the creation of the Ambient Assisted Living Joint Programme (AAL JP[1]), which stimulates research and development of ICT solutions that create better conditions for active and healthy aging. This program has funded a number of projects, with the focus ranging from remote elderly monitoring (REMOTE[2]) to smart homes (HOPE[3]) and social interaction for the elderly (FoSIBLE[4]).

The advances in ICT have enabled significant improvement in activity recognition, which is one of the main enablers of the AAL concept. Numerous prototypes and even commercial solutions already exist and are being studied. For example, Chernbumroong et al. (Chernbumroong, Cang, Atkins, & Yu, 2013) tested multiple sensors worn by elderly subjects on their wrists (the devices were approximately the size of wristwatches) in order to track their activities, such as

[1] Ambient Assisted Living Joint Programme (AAL JP): http://www.aal-europe.eu/, accessed April 15, 2014

[2] Remote health and social care for independent living of isolated elderly with chronic conditions (REMOTE) project: http://www.remote-project.eu/, accessed April 21, 2014

[3] Smart HOme for the elderly PEople (HOPE) project: http://www.hope-project.eu/, accessed April 25, 2014

[4] Fostering Social Interactions for a Better Life of the Elderly (FoSIBLE) project: http://fosible.eu/, accessed April 10, 2014

walking, sleeping, ironing, brushing teeth, and washing dishes, with an accuracy of over 90%. Interesting result in this area emerged from the 2012 EvAAL (Evaluating AAL Systems through Competitive Benchmarking) competition (Alvarez-garcia, Barsocchi, Chessa, & Salvi, 2013). The competition focuses on evaluation of AAL solutions; however, the organizers acknowledge the complexity of such systems, taking a bottom-up approach to their evaluation (starting from components and services of the systems to entire AAL platforms). Thus, the 2012 competition addressed localization and activity recognition systems. In total, 11 systems were competing in conditions close to real environments. Although the results were promising and some of the solutions had relatively high levels of accuracy, the competition also revealed areas for improvement, especially regarding user acceptance (i.e., how much the system disturbs a user) and interoperability (i.e., how easily the system is installed and is possibly integrated into AAL platforms).

There are also prototypes that address several challenges in the AAL concept other than advanced activity recognition. For example, an experimental apartment was created at Fraunhofer IESE[5] inside the institute's facilities for testing and developing different AAL solutions. An AAL approach studied as part of the ProAssist4Life[6] project involved multi-sensor nodes, which capture not only residents' movement patterns, but also some environmental conditions, triggering emergency notifications in case of sudden falls or other abnormal situations. Another example of a space to test prototypes is the CareLab situated in Eindhoven, Netherlands. This research space consists of a one-bedroom apartment equipped with different sensor networks. The apartment is a test bed for different research projects, including research into daily assistance, cognitive stimulation and social interaction for the elderly (Ruyter & Pelgrim, 2007). Botia et al. (Botia, Villa, & Palma, 2012) also combine various environment sensors, such as pressure sensors in a bed and chairs, door sensors, and infrared sensors, with user state sensors for recognizing possible user activities and abnormal situations. The system has evolved into a commercial product available in Spain.

The common trend in these projects is the shift towards usage of embedded devices along with simple "sense and send value" devices. New devices are capable of performing computations, are reprogrammable and support various communication protocols. This trend gives new possibilities for AAL systems, although it also poses new challenges in terms of interoperability and integration. One of the approaches to overcome these challenges is service-oriented architecture (SOA). The approach involves encapsulating functionalities of the system into services provided by devices and system software components. For example, the Gato Tech Smart House aims to deliver an assistive environment through integration and cooperation of different devices at home that function independently and communicate via services (Helal et al., 2005). The Amigo[7] project has a similar service-oriented view of AAL systems and targets the integration of multiple service providers at home to create a coordinated system adaptable to different user needs and situations. Another approach (Sun, Florio, Gui, & Blondia, 2010) even aims at a system to abstract user needs into services and establish a social AAL platform. In this platform, users are able to publish their needs, for example to go for a walk in the park or chat with someone, and their capabilities, for example if they are free to help with cleaning or ready to meet for walk, and the system will match published needs and capabilities, suggesting appropriate help to users.

All in all, ICT advances have enabled integration of heterogeneous devices and a combination of different functionalities in systems, thereby opening new potential for AAL solutions.

Building Automation Systems

The development of Building Automation Systems (BAS) started in the 1980s with the automation of simple building functions such as HVAC and lights ("Frost & Sullivan," 2011). The motivation behind first developments in this area is still relevant with regard to current systems: to minimize the lifecycle costs of buildings, to increase energy efficiency, and to improve the comfort of building occupants ("Frost & Sullivan," 2011), (R. Yang & Wang, 2012). The general trend of BAS evolution enabled by ICT development is a shift from restricted access and local control to remote control and monitoring of building facilities (Sauter et al., 2011). The key ICT enabler of this transition are the emerged standards of communication that have enabled IP-based control and monitoring and have facilitated device integration and system interoperability on different levels of system architecture (Figueiredo & Martins, 2010; Hoffmann, 2011; Kastner, Neugschwandtner, Soucek, & Newmann, 2005; Y. Yang, Zhu, Maasoumy, & Sangiovanni-Vincentelli, 2012).

Contemporary BA systems attempt to cross-integrate different building functions under one system, reducing energy consumption by considering multiple factors inside and outside of the building, such as weather and residents' activity (Nguyen & Aiello, 2013). For example, a BA system by Johnson Controls Inc. (Glendale, Wisconsin) integrates

[5] Fraunhofer Institute of Experimental Software Engineering (Fraunhofer IESE): http://www.iese.fraunhofer.de/en.html, accessed May 15, 2014

[6] Proactive assistance for critical situations – Emergency detection for senior citizens (ProAssist4Life): http://www.iese.fraunhofer.de/en/customers_industries/healthcare/referenzprojekt_proassist4life.html, accessed May 02, 2014

[7] Ambient intelligence for the networked home environment (Amigo): http://www.hitech-projects.com/euprojects/amigo/index.htm, accessed May 18, 2014

multiple energy sources, such as solar panels and heat pumps, in addition to energy coming from an electricity supplier; it then uses internal algorithms for controlling energy consumers inside the building, such as lights, heating and air conditioning with respect to the outside weather and office occupancy patterns (Hoffmann, 2011).

Usage of embedded devices and SOA adoption are also quite common in the Building Automation domain (Perumal, Ramli, & Leong, 2014), (Sleman & Moeller, 2011). Although promising in the research and prototypes, the adoption is limited in real systems due to legacy systems (systems, which are heavily rely on older technologies) and the high cost of renovations. There are several solutions, which address problem of legacy systems. For example, Cimetrics Inc. offers analytical support in tracing energy consumption patterns. It uses data generated by legacy systems of the buildings, which is sent to the Cimetrics analytics center. Then the Cimetrics' platform analyzes this data by using internal algorithms and develops customized strategies for saving energy in the building (Lee, 2011). Another approach is to install additional (quite cheap) equipment, which serves as intermediate layer between old devices and new communication protocols (Jarvinen, Litvinov, & Vuorimaa, 2011). Both solutions allow keeping legacy systems and involve only minor investments in introducing additional devices and services.

To summarize, modern BA systems are large scale and are moving towards incorporation of various, traditionally separate, systems into a single one, such as the integration of heating, air conditioning, safety, and security. Integration of heterogeneous devices and interoperability between system components and services are presently the key challenges in building automation (Y. Yang et al., 2012), (Marinakis, Doukas, Karakosta, & Psarras, 2013). The future of BA systems may extend to providing multiple services tailored to the needs of building occupants and facility managers. In other words, the latest trend in building automation is to consider not only the energy consumption of the building but also to maximize the comfort of building occupants as well as to facilitate the maintenance of facility ("Frost & Sullivan," 2011) while personalizing the delivered services and human-machine interfaces (HMIs) (Mairs et al., 2012). By considering specific residents' needs, BA systems become more friendly and user-oriented (Agarwal et al., 2010), (Wicaksono, Rogalski, & Kusnady, 2010).

Benefits of a Hybrid Automation System

Care facilities could benefit much from the usage of AAL and BA systems, even if those systems function separately on the site. AAL is able to not only deliver solutions for the aging population that facilitate "aging in place", but also to promote safer and more independent living in care facilities, such as retirement or rehabilitation homes, where residents are quite independent and active. The added value of AAL for such care facilities is in providing a comfortable living environment that addresses the specific needs of the elderly while stimulating them to engage in more independent activities. On the other hand, BA solutions are able to address the specific needs of caregivers, such as in facilitating the management and maintenance of a facility, along with providing a safe environment for residents and reduction of energy consumption.

While each system has a number of benefits, the independence of AAL and BA solutions in the facility could lead to conflicts in objectives, thus resulting in a lower degree of comfort for all system users. For example, attempts to save energy in a facility without considering the needs of the elderly occupants could lead to a lower satisfaction level with the living conditions. This issue was identified during the usability tests of care facilities in Sweden, where staff participated in an energy savings program, leaving corridors and common rooms in relative darkness and thus reducing the level of comfort for residents (Andersson et al., 2011). From the caregivers point of view, having more than one system on the site requires more effort to use and manage both, rather than help lighten the daily workload.

The hybrid approach to AAL and BA systems opens up new possibilities in elderly care support. The hybrid system enables new functionalities that consider multiple aspects of the care facilities and the specific needs of different user groups.

MATERIALS AND METHODS

The main objectives of this work are twofold: (1) to perform a preliminary assessment of whether a combination of functionalities of AAL and BA systems could bring benefits to users (residents and personnel of facilities); and (2) to demonstrate the capabilities of modern automation systems and their possible usage in elderly care support. To achieve these objectives, the developed prototype has gone through an entire development cycle starting from the gathering of requirements and design phase to the preliminary assessment via iterative usability tests (Figure 1). This section gives an overview of the general workflow procedure followed in this work during the development of the prototype.

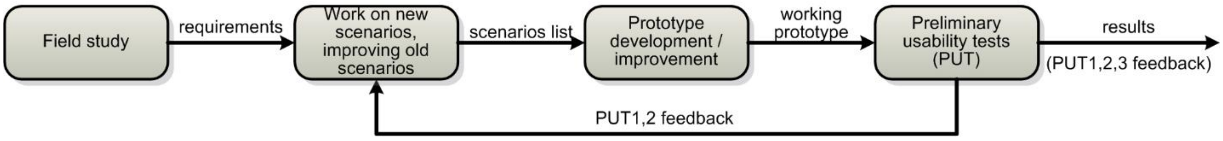

Field study
requirements
Work on new scenarios, improving old scenarios
scenarios list
Prototype development / improvement
working prototype
Preliminary usability tests (PUT)
results
(PUT1,2,3 feedback)
PUT1,2 feedback

Figure 1 The workflow procedure followed in presented work

Field Study

Some studies have found the personnel of various health care facilities to be reluctant to accept new technologies and emphasize the importance of involvement of nurses in the software design process of supportive systems (de Veer & Francke, 2010; Stevenson, Nilsson, Petersson, & Johansson, 2010; While & Dewsbury, 2011). In order to design a system that satisfies the needs of end users and facilitates completion of daily tasks and activities, an extensive field study was performed to identify the main user groups of considered rehabilitation facility and their information needs. The field study was carried out over three days. Each participant was interviewed once for approximately one hour. The interviews were audiotaped and further documentation was made in the form of handwritten notes. Afterwards, all the collected material was analyzed and grouped based on the activity types identified on the site, the information needs and the requirements for and the concerns about the planned system. More information on the field study and its results is presented in the section User Groups and Their Information Needs.

Prototype Development

The prototype represents two boxes with a set of real and simulated sensors from the facility, which we refer to as demoBoxes. The interaction with the boxes is done via HMIs developed for tablets and smartphones. The functionalities supported by the prototype are developed according to the needs of users in the facility and are reported in the Scenarios Supported by the Hybrid Automation System section.

The elderly care domain is challenging in terms of user acceptance of new technologies and services (Heart & Kalderon, 2013), (While & Dewsbury, 2011). Muller et al. (Müller, Neufeldt, Randall, & Wulf, 2012) suggest that collaboration between researchers/developers and residents/personnel is the key factor in determining acceptance of new technologies by users. Thus, the developed prototype has gone through three development iterations, starting from limited functionality to a fully functional prototype in a simulated environment. Two of the iterations addressed feedback from preliminary usability tests, involving end users into development process.

Preliminary Usability Tests

The preliminary usability tests were organized in an iterative manner in order to consider feedback to developed prototype in the early development stages. Due to the iterative nature of the usability tests and the orientation to the development process, the tests were organized with a limited number of users, up to five users for each group, thus posing some limitations on interpretation of the quantitative results and possible generalization to similar cases. Nevertheless, participants were real residents and staff of the rehabilitation facility and provided valuable feedback on requirements and constraints with respect to the specific use case. The age of the participants varied from 35 to 55 years and none of the participants of the residents group had any prior experience with smartphones or any kind of home automation systems.

The tests were performed in one of the common rooms of the facility. The room had a large table, where equipment was placed, including boxes with real and simulated sensors, a smartphone and a laptop for participants. The participants agreed to be audio and video recorded, with the exception of one resident who gave permission only to written notes. The video camera was placed on the table and pointed at the screen of the device (smartphone or laptop) to capture the interaction of the participant with the device and reduce disturbance of the camera. Tests were performed by two organizers: a moderator interacting with the participant and directing the test flow, and a logger making notes during the tests and capturing information that could not be taped directly by camera, such as facial expressions, gestures, and emotions. All the interfaces were translated into the Finnish language, which was used to conduct the usability tests. The duration of a test for one participant, including a post-test interview, was approximately 45 minutes. The next sections give an overview of the case study and present the most salient intermediate results obtained during the work.

A REHABILITATION FACILITY CASE STUDY

Facility Description

The rehabilitation house in our study is situated in Tampere, Finland. It provides different rehabilitation services ranging from physical and psychological therapy to chemical dependency treatment. The facility is designed to offer temporary residence for different clients, including elderly people, people with physical disabilities and alcoholics. The main purpose of the facility is to provide services which support the recovery process after health distorting events, and prepare residents for independent living in their households after the therapy period.

The facility consists of five buildings, three of which have ten residential apartments each and the other two of which contain office spaces and common areas (Figure 2). The common areas include a dining room, a kitchen and common rooms, which might be used for various activities, such as meetings, seminars and discussions stimulating social interaction between residents of the facility.

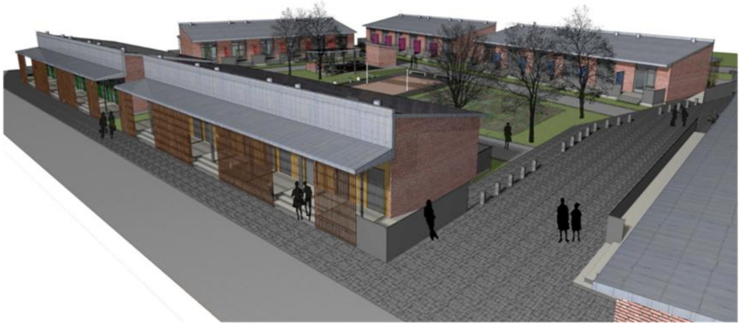


Figure 2 General view of rehabilitation facility

The services provided aim to assist residents in development of everyday life skills, depending on a specific person's constraints, in improvement of health and social interactions, and in receiving personalized guidance for successful independent living. More specifically, the personnel are responsible for scheduling residents' daily activities, for health monitoring and support, and for building daily habits for independent living, such as cooking, doing laundry, and cleaning, with respect to individual constraints.

Although the resident groups of the described rehabilitation facility are not restricted to elderly clients, this case study provides relevant scenarios for elderly care and is easily extendable to other care facilities such as care and retirement homes. The case study provides scenarios requiring residents' health monitoring, daily assistance and support in common activities as well as the challenges faced by caregivers in the form of diverse tasks and activities within the facility.

User Groups and Their Information Needs

Based on the field study, the following groups of users have been identified: maintenance, nurses, the facility owner, and residents.

Maintenance

The maintenance personnel on the site are outsourced from a property management company. The tasks and work hours of workers on the site are defined in a contract, but most of the work hours are flexible, meaning that somebody from the maintenance company is in the facility only if there is something wrong. The main responsibility of the maintenance personnel on the site is operational support for all devices in the buildings: status monitoring, set points adjustment, and various measures to prevent equipment faults.

The first-priority information needed by personnel is the status of all devices and details of notification messages, which include information on time, place, and cause of the notification. The challenge here is to present massive information flows in a convenient manner and give hints to personnel about the cause of a problem and the location in a way that he/she feels better informed instead of overwhelmed. Thus, the main expectations for the system from a maintenance point of view is the logical structure of functions, convenient information representation, easy access from anywhere at any time, ease of use and support of the main daily activities related to device monitoring, for example reminders of scheduled device checkpoints.

Nurses

The healthcare personnel on the site are represented by two nurses, who have morning (08:00-16:00) and evening (14:00-20:00) shifts. The nurses are the most engaged group of users that have diverse responsibilities in the facility. They are in charge of planning of residents' daily activities, health monitoring, and organizing of various thematic group meetings. Moreover, having maintenance personnel as part-time support, they are also responsible for overall maintenance of facilities, including filling in of official forms, reporting accidents to maintenance, contacting most of the outside authorities, and handling various residents' requests.

The information needs of nurses are quite diverse and range from the status of residents, such as information about health care and the activities schedule, to the state of the overall facility, including device statuses and warning/alarm notifications, schedules of workers and maintenance plans. Given the wide range of responsibilities and tasks, it becomes even more important to provide simple ways of handling the system, which were also indicated as main requirements by the nurses. The system is also expected to facilitate daily tasks, to provide better situation awareness – to be able to see the status of the situation in one glance – to support scheduling of daily activities, and to improve overall communication between the different user groups in the facility.

Facility Owner

The owner of the facility represents a separate user group. He does not engage in daily activities in the facility and is mostly responsible for the functioning of the administrative part of facility, for example determining the amount of required services from outside companies based on needs reported by maintenance and caregivers. The owner's information needs are similar to the nurses': evaluate the situation at a glance. However, his needs are more management-oriented and do not include specific information on the health status of residents or a detailed daily plan. The main requirements from the facility owner are easy access to the system (e.g., if he travels or is at another facility), logical and reliable functioning.

Residents

Although the residents are the most heterogeneous group in the facility, the field study has indicated common requirements and needs expressed toward the system. First, residents would like to be informed and to have some reminders about the common activities in the facility, for example doctor visitations and the plan for the day. Secondly, most of the residents wished to have information about the status of their own apartments while, for example, being outside, including information about safety and security issues such as the lights, doors, and electric stoves. This group of users has the least experience with technologies in everyday life; thus the main concern was related to the possible complexity of the system and necessity to spend a long period of time learning it.

Information Exchange between User Groups

The primary flow of information within the rehabilitation facility occurs between three groups of users: nurses, residents and maintenance. The residents may address specific issues to either maintenance services or nurses (Figure 3). It is supposed that residents request device-specific services from maintenance personnel, while health-care related issues are requested from nurses; yet in reality, residents feel more comfortable communicating with nurses and most of the requests are addressed to them. This creates additional workload for nurses, emphasizing the need for better scheduling and communication tools for caregivers in the facility. The maintenance personnel and nurses also communicate with the owner of the rehabilitation house. They report on the status of the facility, general issues and needs from external companies, if these needs are beyond existing contract terms.

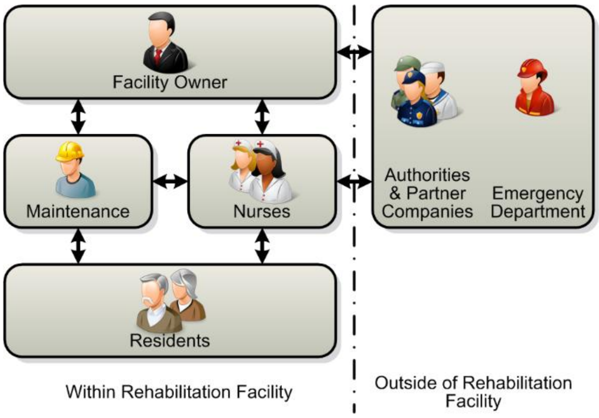

Facility Owner
Maintenance
Nurses
Residents
Authorities & Partner Companies
Emergency Department
Within Rehabilitation Facility
Outside of Rehabilitation Facility

Figure 3 Model of information exchange between different user groups of the rehabilitation facility

The personnel of the rehabilitation facility also communicate with the other companies providing services from outside. Most of the communications are handled by nurses and maintenance staff. They report on issues in the facility, prepare needed forms and schedule visits. The owner contracts services for the facility, discusses contract terms and conditions.

The field study demonstrated that communication between user groups and their information needs are quite complex; some of the tasks are delegated to the "wrong" user group, leading to an additional workload for this group. An example would be nurses receiving maintenance related requests from residents. The diversity of tasks and information needs creates requirements that could only be addressed with a combination of AAL and BA functionalities, such as nurses both monitoring of the health of residents and the overall situation in the facility).

Description of an Automation System in the Facility

The facility is equipped with an automation system that monitors and controls different environmental conditions. The system has a number of sensors and actuators enabling different functionalities. For example, each one-room apartment has at least the following list of sensors: a temperature sensor, an infrared (IR) sensor, a water leak sensor, fire detectors (smoke and sprinklers), heating controls (with the levels defined based on outside temperature and internal temperature set points), door locks (which might be accessed and controlled by personnel of the facility), nurse alarms and acknowledgement buttons, and air ventilation controls (Figure 4). The number and type of devices in the apartment could be tailored to specific user needs; for example it is possible to integrate user-state monitoring sensors, fall detectors or additional alarm buttons to increase the safety and comfort of the residents.

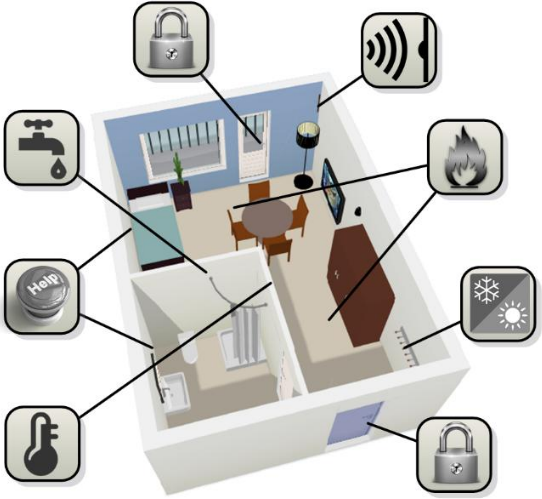


Figure 4 Standard installation of devices in the apartment of the rehabilitation facility

The system operates by means of the Cromi SCADA (supervisory control and data acquisition) software, which integrates data from all devices and allows building customized HMIs and application logics. The first version of the application was developed and installed under the ASTUTE[8] project, with the main purpose being to integrate all available devices and provide access to personnel user groups. This version is more BA oriented; it monitors and automates the main, building-related functions of the facility (air ventilation, heating system, security) and is accessible only to personnel of the facility. Moreover, it does not consider specific needs of different user groups. Maintenance workers, nurses and the owner of the building are exposed to the same HMIs and functionalities of the system while residents of the facility are not allowed to use the system at all due to security reasons.

In order to develop a more usable system addressing specific needs of different user groups in the rehabilitation facility, a test bed was built simulating the physical environment of the real system and allowing safe testing with different technologies and system functionalities. The test bed is represented by two demoBoxes, which have programmable controllers connected to devices and switches. Each demoBox simulates the same amount and type of sensors as in a single real apartment of the facility. The rest of the devices in the other apartments are simulated with software. It is also possible to connect additional sensors to the controller inside the demoBox for testing purposes. The integration of sensors and HMI development was also done with the usage of SCADA software. The test bed has enabled a combination of BA and AAL functionalities and was used during the usability tests with the residents and staff of the facility. The next section describes the functionalities of the test bed solution.

SCENARIOS SUPPORTED BY THE HYBRID AUTOMATION SYSTEM

The underlying system infrastructure and software allow usage of mobile devices displaying all essential information and having access to the system at any time and place. The personnel of the facility have tablets as the main interaction tool with the system, while residents of the facility are more likely to have personal smartphones, which were considered to be the displaying device for the residents' user group.

Facility Monitoring by Personnel

The rehabilitation facility is equipped with numerous devices and sensors controlling and monitoring different environmental conditions in the facility. The challenge for personnel HMIs is in presenting this information in a convenient manner. In order to support personnel in understanding the situation at one glance, a schematic map representation of the facility is used in the central part of the HMI (Figure 5).

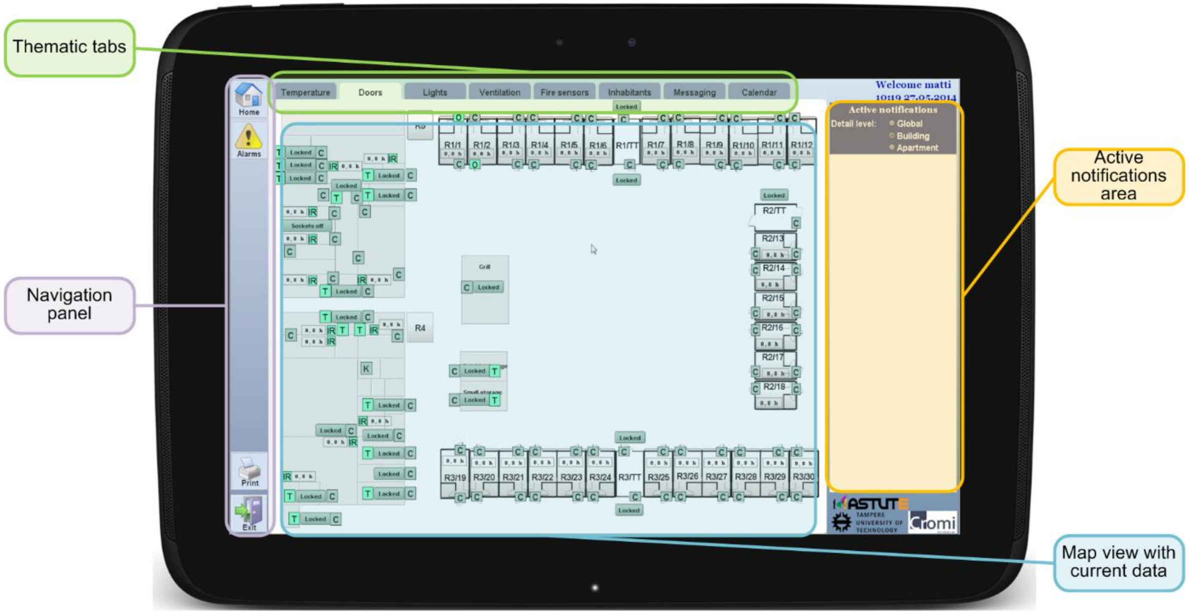


Figure 5 An example view of the facility monitoring interface delivered to personnel user groups (maintenance, nurses, facility owner)

The field study determined the information needed and its priority for different groups of personnel. Based on this information, thematic tabs of the HMI further categorize available data (Figure 5). The number of tabs is tailored to the

[8] Pro-active decision support for data-intensive environments (ASTUTE): http://www.astute-project.eu/, accessed June 15, 2014

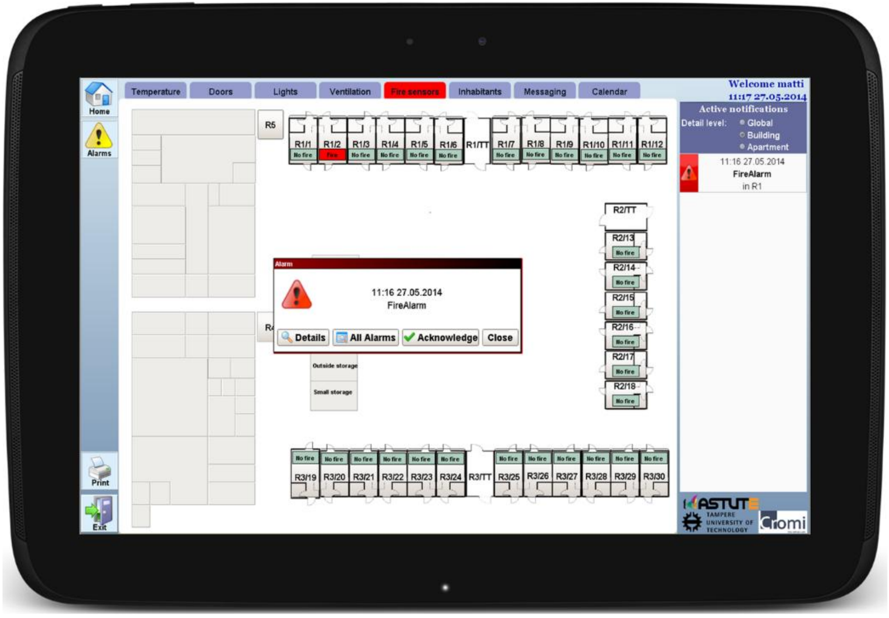
needs of the specific groups of personnel. For example, in addition to tabs for devices, nurses have tabs displaying health-related information of the residents and the calendar of events in the facility. In contrast, the manager is exposed to an HMI with fewer device tabs and no detailed information about residents of the facility.

The active notifications area is used to further simplify interaction with the system. The area displays messages specific to each user of the system. For example, the maintenance personnel are not exposed to nurse-related notifications. The area could be adjusted by each user in order to group information based on locations. For instance, the facility owner would be more interested to know whether there are any problems on the site at the global detail level, while maintenance personnel would like to have a more detailed information display at the building or apartment level. The messages in the area appear according to their priority with highest priorities first.

The messages are divided into three priority groups: low, medium, and high (notifications, warnings and alarms respectively). Each priority group has its own pattern of appearance, summarized in Table 1. For instance, high priority messages, called alarms, are accompanied by a dialog box which appears periodically while the alarm is active, a sound specific to each alarm type, blinking of labels of devices that cause the alarm, and a message in the active notifications area.

Table 1 – HMI patterns for system messages of different priorities

The example of a high priority message pattern is presented in Figure 6. The alarm source, a sprinkler sensor, could be easily located from the *"Fire sensors"* tab because of the blinking of a related tab and label. Further information might be found from a dialog box under the *"Details"* button. The alarm is also accompanied by a specific sound, simplifying interpretation of the alarm type at first appearance.



Welcome matti
11:17 27.05.2014

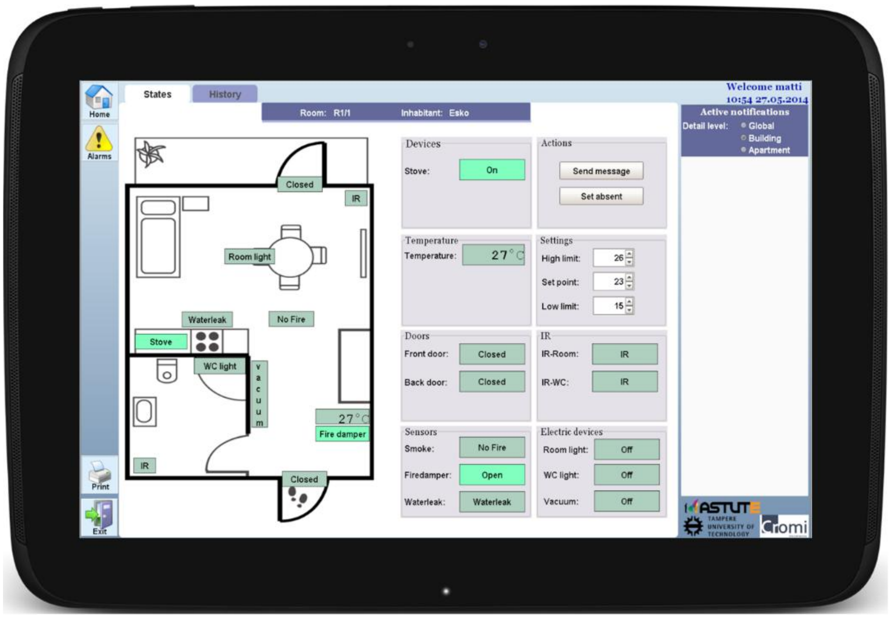
Active notifications
Detail level: ● Global ○ Building ● Apartment

11:16 27.05.2014
FireAlarm
in R1

Alarm
11:16 27.05.2014
FireAlarm

Figure 6 An example of an appearance of a high priority message. The corresponding thematic tab (*"Fire sensors"*) and the label of the sensor are blinking in red; an alarm is indicated in the active notifications area; the dialog box with available actions on the alarm appears periodically; the specific sound rings on the first alarm indication.

Each apartment also has a map representation, which could be accessed by clicking on the corresponding apartment on the map. This view contains the current status of all sensors in the apartment and specifies their locations on the map (Figure 7). The sensor history data could be accessed from the history tab, where information is presented in the form of trends and bars simplifying its interpretation.



Welcome matti
10:54 27.05.2014

Room: R1/1 Inhabitant: Esko

Devices
Stove: On

Temperature
Temperature: 27°C

Settings
High limit: 26
Set point: 23
Low limit: 15

Doors
Front door: Closed
Back door: Closed

IR
IR-Room: IR
IR-WC: IR

Sensors
Smoke: No Fire
Firedamper: Open
Waterleak: Waterleak

Electric devices
Room light: Off
WC light: Off
Vacuum: Off

Active notifications
Detail level: ○ Global ○ Building ● Apartment

Figure 7 An example of apartment status monitoring delivered to personnel user groups

Apartment Monitoring by Residents

Residents of the facility have the right to monitor and control the devices in their own apartments that do not influence functioning of the facility. For example, they are allowed to control lights and doors, but they cannot control air ventilation or heating valves. The average size of the smartphone screens and lack of technical experience of the residents put simplicity of interaction between the system and the residents as a central requirement. Therefore, the designed HMI has large thematic buttons with self-descriptive pictures in the main window (Figure 8, a). These buttons navigate to the detailed information on the status of selected devices, available in the form of a list or a map (Figure 8, b, c). The user is able to change the device state, for example turn on and off lights, by simply clicking on the corresponding button. The non-device buttons ("*Support*" and "*Events*") are used for interaction with personnel and will be explained in the Collaboration between User Groups scenario.

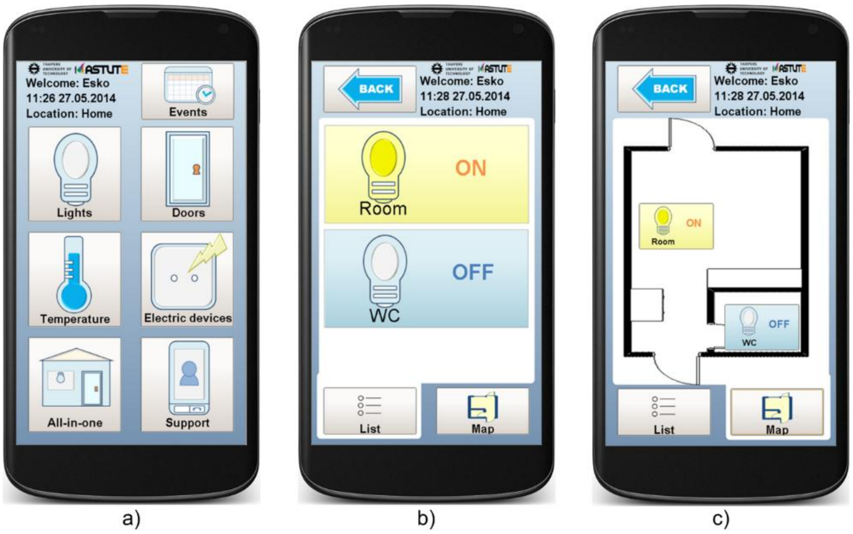


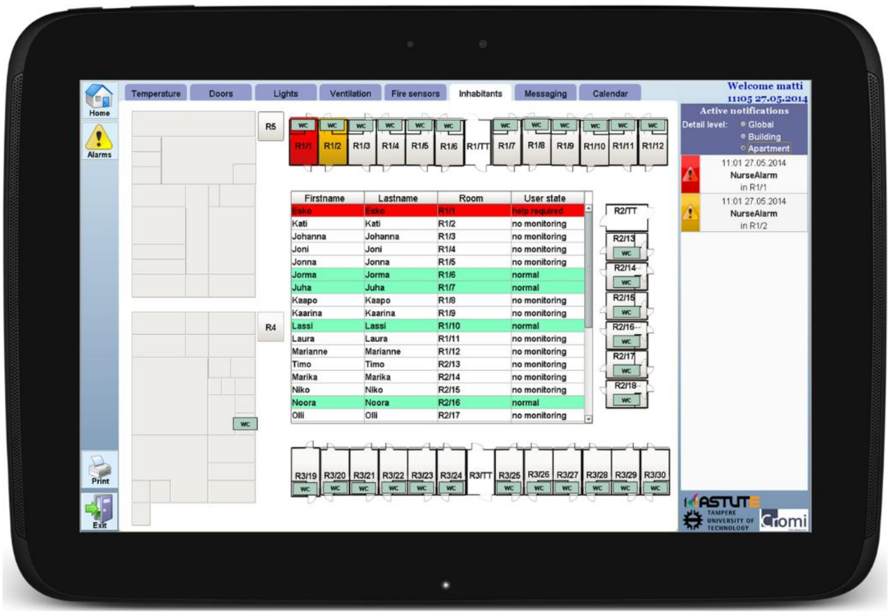
Figure 8 An example of monitoring screens delivered to the residents user group: a) the main view of the application; b) a detailed list view of devices; c) a detailed map view of devices

Health Monitoring of Residents by Nurses

Depending on the resident's wellbeing, he/she could be monitored with additional user-state sensors to detect, for example heart rate, ECG, blood pressure, and falling. Data from these devices is summarized in one residents' status table, which could be accessed by nurses (Figure 9). The detailed information from user-state devices is accessed by double-clicking on the corresponding row in the table.

The notifications are marked with different priorities based on the type and number of triggered devices. For example, if a resident triggers a nurse call button from his/her apartment, the nurse receives a warning notification, which is marked with yellow in Figure 9). However, if a user-state sensor triggers a notification which is considered to be of higher priority than a self-triggered nurse call, the nurse receives a high priority alarm (marked with red in Figure 9).

All notifications are grouped by location and type. Therefore, if nurse calls and user-state sensor notifications are from the same apartment, only one combined message of the highest priority is delivered to the nurse. In this way, the nurse is exposed to less information flow that needs immediate processing, but she is still able to access all the required details, such as device data, descriptions and specific instructions through the detailed notification view. The localization of alarms is also visually supported by blinking of apartment labels, which further improves understanding of the situation.



Welcome matti
11:05 27.05.2014

Active notifications

Detail level: ● Global ● Building ○ Apartment

11:01 27.05.2014
NurseAlarm
in R1/1

11:01 27.05.2014
NurseAlarm
in R1/2

Firstname	Lastname	Room	User state
Esko	Esko	R1/1	help required
Kati	Kati	R1/2	no monitoring
Johanna	Johanna	R1/3	no monitoring
Joni	Joni	R1/4	no monitoring
Jonna	Jonna	R1/5	no monitoring
Jorma	Jorma	R1/6	normal
Juha	Juha	R1/7	normal
Kaapo	Kaapo	R1/8	no monitoring
Kaarina	Kaarina	R1/9	no monitoring
Lassi	Lassi	R1/10	normal
Laura	Laura	R1/11	no monitoring
Marianne	Marianne	R1/12	no monitoring
Timo	Timo	R2/13	no monitoring
Marika	Marika	R2/14	no monitoring
Niko	Niko	R2/15	no monitoring
Noora	Noora	R2/16	normal
Olli	Olli	R2/17	no monitoring

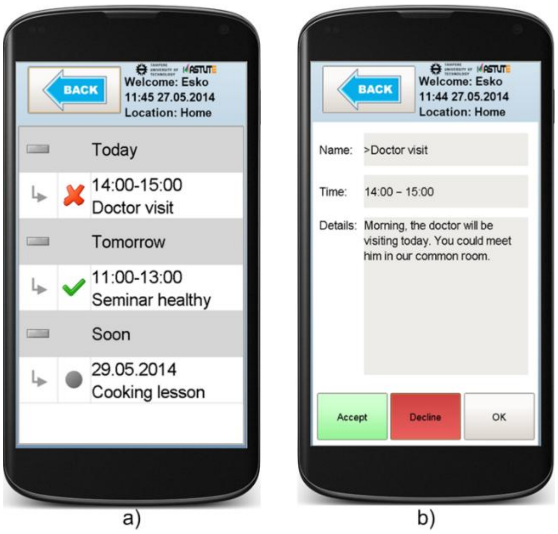
Figure 9 An example of health monitoring of residents by the nurses user group. The HMI also depicts alarm (red) and warning (yellow) message patterns.

Collaboration between User Groups

Better means of communication were needed by all user groups in the facility. The improvements were expected in the scheduling of daily activities and the handling of notifications from residents. The developed prototype supports two means of communication for all user groups: messaging and calendar events.

Messaging is the simplest way of exchanging information between users of the system. Personnel have this function in a separate tab of the HMI, where it is possible to choose recipients and enter free-form messages that will be received by all the selected users. A simplified version of messaging is also accessible for residents of the facility from smartphones. In this version, residents are able to choose a nurse or a maintenance worker as one of the recipients and send one of the predefined messages. The topics and general content of the predefined messages were determined based on the results of the field study. The advantages of having predefined messages for residents are twofold: (1) it allows better sorting of requests for personnel (different topics are divided according to recipients: maintenance or nurses) and (2) residents are able to send messages in two clicks without having to type it in the smartphone.

The calendar functionality supports users of the system with scheduling of daily activities and notifications. The personnel of the facility are able to create various events in the separate calendar tab, and invite selected users, personnel and residents, to participate. Residents may access their personal schedule through the events button (Figure 10, a) and mark their participation through the detailed event view (Figure 10, b). Based on this feedback, nurses are able to better organize common activities, estimate the number of participants and estimate the resources needed.



Welcome: Esko
11:45 27.05.2014
Location: Home

Today

14:00-15:00
Doctor visit

Tomorrow

11:00-13:00
Seminar healthy

Soon

29.05.2014
Cooking lesson

Welcome: Esko
11:44 27.05.2014
Location: Home

Name: >Doctor visit

Time: 14:00 – 15:00

Details: Morning, the doctor will be visiting today. You could meet him in our common room.

Figure 10 (a) An example of the calendar view delivered to the residents user group; (b) an example of detailed information about a calendar event delivered to the residents user group

FINDINGS FROM THE PRELIMINARY USABILITY TESTS

During the usability tests, participants were asked to do a number of tasks using the provided device. The tasks were related to the daily activities of the participants and were intended to test how a participant understands the situation in a simulated environment, whether he/she is able to act according to the situation (e.g., switch on/off corresponding devices, understand notifications and alerts), and how difficult it is to handle the system without prior experience and knowledge. The results were assessed with different usability metrics, including time on tasks, error and completion rates, SUS and satisfaction questionnaires.

The preliminary results have shown positive acceptance of the system by participants. The personnel of the facility have indicated a clear understanding of system messages such as notifications, warnings, alarms, easy access to the information from, for example, thematic tabs and the apartment view, and have appreciated the possibility to send messages to different users of the system. Participants from residents group have found the proposed interface simple to use and easy to learn, which correlates with the results of the completion and error rates per task and have expressed a hope to see this application in the rehabilitation facility in the future.

During the development process, the feedback from the iterative usability tests have pointed to some issues, which were addressed in the next development rounds. For example, in the original system design each warning and alarm notification in the personnel HMI was accompanied by a specific sound, with the assumption that it simplifies understanding of the nature of the notification. However, the personnel of the facility felt rather overwhelmed with the multiple sounds instead of better oriented, so it was decided to have distinguished sounds only for the highest priority alarms, which was accepted much better. Another interesting outcome was the increased amount of suggestions from participants for system functionality and suggestions about the “look and feel” with every usability test. The direct involvement of users in the system design process and subsequent observation of the results liberated users to express expectations for the system and allowed them to formulate more wish to have functions than were collected in the field study. This is reflected in the average SUS score that increased from 79.5 in the first usability test round to 86 in the third usability test round, indicating actual improvement in the system and positive development overall.

In summary, preliminary usability tests have shown a willingness of the users, both personnel and residents, to use the developed system and to contribute to the development process. The first two rounds of usability tests in which a combination of basic BA and AAL functionalities were tested and in which users witnessed how their previous requirements have been addressed, allowed users to verbalize their desired functionalities and better collaborate in the development process.

CONCLUSION

Growth in the elderly segment of the population in developed countries and the growing shortage of nurses and other caregivers requires new approaches for elderly care and support. This paper proposes a hybrid automation system for elderly care support addressing needs of different user groups in care facilities. The approach is enabled by the present state of ICT, allowing simpler integration and communication between various system components and hence having a potential to extend system applications and functionalities.

The work considers a rehabilitation facility as an illustrative case study, demonstrating the possibilities of the approach. The facility served as a source of requirements and usability test feedback for development of the prototype. The field study indicated a variety of tasks and information needs of the facility personnel that could only be addressed by a combination of functionalities of different systems. The prototype developed has the same physical infrastructure as a real system in a facility and combines functionalities of BA and AAL systems, such as control and monitoring of facilities, health care support, handling of alarms and messages, calendar events. The preliminary usability tests indicated positive acceptance of the prototype and an increasing SUS score over three test iterations.

One limitation of the work is the small number of usability test participants, up to five in each user group, which was sufficient for development purposes but should be carefully considered when generalizing the results for similar use cases. For example, results in other studies (While & Dewsbury, 2011), (Heart & Kalderon, 2013) indicate a reluctance on the part of nurses and the older people to accept new technologies, which was not observed in this work. This could be explained by the fact that the development process in this work was highly influenced by the end users of the system through the field study and usability test feedback, allowing customization for a specific use case design; however, further research on this phenomenon is required.

The presented case study has demonstrated the possibilities of a hybrid automation system. While combined functionalities result in growing complexity of the systems, the present day technologies allow the hiding of this complexity from end-users under convenient and user-friendly interfaces, which are adaptable to different situations in the environment. The benefits of this approach are seen in the addressing of specific information needs of various user groups, the reduction of daily work loads for personnel and the provision of more comfortable and safe environments for residents. Future work will focus on organizing more extensive usability tests with participants from related facilities, such as retirement homes and assisted living facilities.

ACKNOWLEDGEMENTS

This work is partially supported by the ARTEMIS First Call 2010 Program under Grant Agreement number 269334, corresponding to the project ASTUTE (Pro-active decision support for data-intensive environments) and the Doctoral Programme of the President of the Tampere University of Technology. The authors also would like to thank Ville Hämäläinen for support in preparing and conducting of usability tests as well as direct involvement in the work on the prototype.

REFERENCES

Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., & Weng, T. (2010). Occupancy-driven energy management for smart building automation. In *2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building* (pp. 1–6). New York, New York, USA: ACM Press. doi:10.1145/1878431.1878433

Alvarez-garcia, J. A., Barsocchi, P., Chessa, S., & Salvi, D. (2013). Evaluation of localization and activity recognition systems for ambient assisted living: the experience of the 2012 EvAAL competition. *Journal of Ambient Intelligence and Smart Environments*, *5*(1), 119–132. doi:10.3233/AIS-120192

Andersson, M., Lindahl, G., & Malmqvist, I. (2011). Use and usability of assisted living facilities for the elderly: an observation study in Gothenburg Sweden. *Journal of Housing For the Elderly*, *25*(4), 380–400. doi:10.1080/02763893.2011.621860

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010

Bobillier Chaumon, M.-E., Michel, C., Tarpin Bernard, F., & Croisile, B. (2013). Can ICT improve the quality of life of elderly adults living in residential home care units? From actual impacts to hidden artefacts. *Behaviour & Information Technology*, *33*(6), 574–590. doi:10.1080/0144929X.2013.832382

Botia, J. A., Villa, A., & Palma, J. (2012). Ambient Assisted Living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications*, *39*(9), 8136–8148. doi:10.1016/j.eswa.2012.01.153

Chernbumroong, S., Cang, S., Atkins, A., & Yu, H. (2013). Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, *40*(5), 1662–1674. doi:10.1016/j.eswa.2012.09.004

Coomans, K., Smet, H. De, & Heylighen, A. (2011). In search of a future for large-scale care homes in Flanders. *Journal of Housing For the Elderly*, *25*(4), 329–351. doi:10.1080/02763893.2011.621858

De Veer, A. J. E., & Francke, A. L. (2010). Attitudes of nursing staff towards electronic patient records: a questionnaire survey. *International Journal of Nursing Studies*, *47*(7), 846–854. doi:10.1016/j.ijnurstu.2009.11.016

Demiris, G., & Hensel, B. (2009). "Smart Homes" for patients at the end of life. *Journal of Housing For the Elderly*, *23*(1-2), 106–115. doi:10.1080/02763890802665049

Figueiredo, J., & Martins, J. (2010). Energy Production System Management – renewable energy power supply integration with Building Automation System. *Energy Conversion and Management*, *51*(6), 1120–1126. doi:10.1016/j.enconman.2009.12.020

Frost & Sullivan. (2010, November). *European markets for assisted living technologies*. Retrieved April 14, 2014, from http://www.frost.com

Frost & Sullivan. (2011, August). *European Building Automation Systems Market: Paving the Way for Smart and Energy Efficient Buildings*. Retrieved March 20, 2014, from http://www.frost.com

Frost & Sullivan. (2012, May). *Long-Term Care in Germany*. Retrieved March 20, 2014, from http://www.frost.com

Frost & Sullivan. (2013, January). *European Market for Remote Patient Monitoring Applications*. Retrieved March 21, 2014, from http://www.frost.com

Frost & Sullivan. (2014, February). *Connected Health and eGovernment for smarter cities: exploring the current landscape and future opportunities*. Retrieved April 08, 2014, from http://www.frost.com

Heart, T., & Kalderon, E. (2013). Older adults: are they ready to adopt health-related ICT? *International Journal of Medical Informatics*, *82*(11), e209–e231. doi:10.1016/j.ijmedinf.2011.03.002

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., & Jansen, E. (2005). The Gator Tech Smart House: a programmable pervasive space. *Computer*, *38*(3), 50–60. doi:10.1109/MC.2005.107

Hoffmann, T. (2011). Building Automation and Control Systems Case Study: Johnson Controls Headquarters, Glendale Wisconsin. In B. L. Capehart & T. Middelkoop (Eds.), *Handbook of Web Based Energy Information and Control Systems* (pp. 521–527).

Jarvinen, H., Litvinov, A., & Vuorimaa, P. (2011). Integration platform for home and building automation systems. In *Consumer Communications and Networking Conference* (pp. 292–296). IEEE. doi:10.1109/CCNC.2011.5766475

Kastner, W., Neugschwandtner, G., Soucek, S., & Newmann, H. M. (2005). Communication Systems for Building Automation and Control. *Proceedings of the IEEE*, *93*(6), 1178–1203. doi:10.1109/JPROC.2005.849726

Lee, J. (2011). Brining Building Automation Systems under control. In B. L. Capehart & T. Middelkoop (Eds.), *Handbook of Web Based Energy Information and Control Systems* (pp. 529–538).

Mairs, S. M., McCoy, S. M., Richards, D. M., Eiynk, B., Seewald, J. S., Brink, J., … Kistler-Robinson, J. K. (2012, January 17). Building automation system facilitating user customization. Retrieved from http://www.google.com/patents/US8099178

Marinakis, V., Doukas, H., Karakosta, C., & Psarras, J. (2013). An integrated system for buildings' energy-efficient automation: application in the tertiary sector. *Applied Energy*, *101*, 6–14. doi:10.1016/j.apenergy.2012.05.032

Müller, C., Neufeldt, C., Randall, D., & Wulf, V. (2012). ICT-development in residential care settings: sensitzing design to the life curcumstances of the residents of a care home. In *Human Factors in Computing Systems* (pp. 2639–2648). New York, New York, USA, USA: ACM Press. doi:10.1145/2207676.2208655

National Institute for Health and Welfare. (2010). *Social welfare and health care personnel 2010*. Retrieved April 04, 2014, from http://www.thl.fi/en_US/web/en/statistics/topics/personnel/social_welfare_and_health_care_personnel

Nguyen, T. A., & Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. *Energy and Buildings*, *56*, 244–257. doi:10.1016/j.enbuild.2012.09.005

O'Grady, M. J., Muldoon, C., Dragone, M., Tynan, R., & O'Hare, G. M. P. (2009). Towards evolutionary ambient assisted living systems. *Journal of Ambient Intelligence and Humanized Computing*, *1*(1), 15–29. doi:10.1007/s12652-009-0003-5

Official Statistics of Finland (OSF). (2012, September). *Population projection*. Retrieved April 01, 2014, from http://www.stat.fi/til/vaenn/2012/vaenn_2012_2012-09-28_tie_001_en.html

Perumal, T., Ramli, A. R., & Leong, C. Y. (2014). SOA-Based Framework for Home and Building Automation Systems (HBAS). *International Journal of Smart Home*, *8*(5), 197–206. doi:10.14257/ijsh.2014.8.5.18

Rocha, A., Martins, A., Freire Junior, J. C., Kamel Boulos, M. N., Vicente, M. E., Feld, R., … Rodríguez-Molinero, A. (2013). Innovations in health care services: the CAALYX system. *International Journal of Medical Informatics*, *82*(11), e307–e320. doi:10.1016/j.ijmedinf.2011.03.003

Ruyter, B., & Pelgrim, E. (2007). Ambient assisted-living research in CareLab. *Magazine Interactions - Designing for Seniors: Innovations for Graying Times*, 30–33. doi:10.1145/1273961.1273981

Sadri, F. (2011). Ambient intelligence: a survey. *ACM Computing Surveys*, *43*(4). doi:10.1145/1978802.1978815

Sauter, T., Soucek, S., Kastner, W., & Dietrich, D. (2011). The Evolution of Factory and Building Automation. *IEEE Industrial Electronics Magazine*, *5*(3), 35–48. doi:10.1109/MIE.2011.942175

Sleman, A., & Moeller, R. (2011). SOA distributed operating system for managing embedded devices in home and building automation. *IEEE Transactions on Consumer Electronics*, *57*(2), 945–952. doi:10.1109/TCE.2011.5955244

Stevenson, J. E., Nilsson, G. C., Petersson, G. I., & Johansson, P. E. (2010). Nurses' experience of using electronic patient records in everyday practice in acute/inpatient ward settings: a literature review. *Health Informatics Journal*, *16*(1), 63–72. doi:10.1177/1460458209345901

Sun, H., Florio, V. De, Gui, N., & Blondia, C. (2009). Promises and Challenges of Ambient Assisted Living Systems. In *Sixth International Conference on Information Technology: New Generations* (pp. 1201–1207). IEEE. doi:10.1109/ITNG.2009.169

Sun, H., Florio, V. De, Gui, N., & Blondia, C. (2010). The Missing Ones: Key Ingredients Towards Effective Ambient Assisted Living Systems. *Journal of Ambient Intelligence and Smart Environments*, *2*(2), 109–120. Computers and Society; Artificial Intelligence. Retrieved from http://arxiv.org/abs/1401.2657

While, A., & Dewsbury, G. (2011). Nursing and information and communication technology (ICT): a discussion of trends and future directions. *International Journal of Nursing Studies*, *48*(10), 1302–1310. doi:10.1016/j.ijnurstu.2011.02.020

Wicaksono, H., Rogalski, S., & Kusnady, E. (2010). Knowledge-based intelligent energy management using building automation system. In *IPEC 2010* (pp. 1140–1145). Ieee. doi:10.1109/IPECON.2010.5696994

Yang, R., & Wang, L. (2012). Multi-objective optimization for decision-making of energy and comfort management in building automation and control. *Sustainable Cities and Society*, *2*(1), 1–7. doi:10.1016/j.scs.2011.09.001

Yang, Y., Zhu, Q., Maasoumy, M., & Sangiovanni-Vincentelli, A. (2012). Development of Building Automation and Control Systems. *IEEE Design & Test of Computers*, *29*(4), 45–55. doi:10.1109/MDT.2012.2201130

Publication II

Evchina, Y., Dvoryanchikova, A., & Martinez Lastra, J. L. (2013). Semantic information management for user and context aware smart home with social services. In *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support* (pp. 262–268). Ieee. https://doi.org/10.1109/CogSIMA.2013.6523856

This is an *Accepted Manuscript*, the final version could be found at https://doi.org/10.1109/CogSIMA.2013.6523856

Semantic Information Management for User and Context Aware Smart Home with Social Services

Yulia Evchina, Aleksandra Dvoryanchikova, José L. Martinez Lastra

***Abstract*— Emerging network technologies and growing variety of available devices open new smart home systems perspective for data capturing and analyzing. Systems tend to become wider in scale and are able to capture different aspects of living conditions inside and outside buildings. The variety of available information makes those systems attractive to various user groups with different roles and responsibilities in Smart Home domain. Yet current solutions for monitoring systems do not consider personal user needs, which lead to overwhelming user with excessive information flow. This paper proposes information management system with usage of Semantic Web technologies (ontologies and queries), which takes into account personal user needs in the system and current situation in the environment and reduces information load to the user by providing personal data. It is expected that proposed information management approach will make general monitoring systems user friendly and personal oriented and thus safer during the operation. The detailed description of designed ontology is provided. Two basic scenarios for Smart Homes with social services are considered and developed prototype is described. Current implementation of the proposed architecture shows feasibility of the approach and prompts further fields of research.**

***Index Terms*—Context-awareness, information management, smart homes**

I. Introduction

Modern control and monitoring systems for buildings could incorporate large number of various devices. Those systems have wide scale and usually are distributed over several buildings and sometimes even over cities. The amount of devices creates bulky informational flow, which has to be processed and followed by correspondent actions. At the same time the systems are used by different groups of users with different responsibilities and needs (e.g. sensor information from the apartments could be used by maintenance personnel to analyze the situation in the whole building and by inhabitants to monitor conditions inside their apartments). However, this information is delivered to the user without adapting it to the personal needs. For instance, systems usually have one common interface for displaying all available information. In this case the usage of the system is restricted to authenticated personnel, while common users (like inhabitants), who are interested in partial information, do not have access to the system or would be too overwhelmed by presented data. These problems raise the question of information management in the system.

Traditional monitoring systems meet difficulties in data management, since data are strongly connected with application logic. The meaning of data entities and their relationships with each other are distributed across programming instructions and table relations inside databases. As a result, massive if/else structures complicate programming logic and make it difficult to extend and maintain applications. Presenting new user roles in the systems with their access policy could be a cumbersome task, which involve vast reprogramming across related applications. Moreover, even inside one group with similar access rights users could require different information (e.g. inhabitants are interested in the information about their own apartments). Thus, there is a need in intelligent information filtering, which will be sensitive to the user role in the system and current situation in the environment. The aim of the paper is to propose approach for reducing informational load to the user, thus making systems user friendly and safer during the operation. Informational load could be reduced by filtering out data, which is irrelevant to the particular user or not important in the current context.

It is supposed that Semantic Web technologies (like ontologies and queries) will help to facilitate problem of information management by enriching existing data sources with data meaning (or semantic). The domain of origin for Semantic Web is World Wide Web, which deals with vast amount of heterogeneous data. To reduce information overwhelming and make connections between different data sources Semantic Web offer new data models and techniques to work with these models thus making data self-consistent and coherent. Adding semantic to the data in Smart Home domain will allow information management, which is separate from application logic. This separation will add flexibility to the data, which could be used by different applications, and to the system applications, which could focus on the specific tasks instead of dealing with data management.

This paper describes ontology for smart homes with social services, which is used for information management within the monitoring system. The idea behind the usage of ontology is in providing semantic connections for available data, thus allowing information filtering for specific users by analyzing

Manuscript received November 1, 2012. This work was supported in part by the ARTEMIS JU through the ASTUTE project.

Evchina Y., Dvoryanchikova A. and Martinez Lastra J.L. are with Tampere University of technology, Department of Production Engineering, Tampere, Finland (correspondent contacts are: yulia.evchina@tut.fi, aleksandra.dvoryanchikova@tut.fi, jose.lastra@tut.fi).

user role in the system, his responsibilities and current situation. The scenarios with illustrative examples of ontological usage are considered, and implementation prototype with enabling technologies is proposed.

The rest of the paper is organized as follows. Section II contains brief description of related work in the domain. Section III presents domain requirements, which should be considered during the system development. In section IV general architecture of the information management system is explained. Section V describes main structure of ontology and provides illustrative examples of its usage. Implementation approach is presented in section VI, which is followed by conclusions.

II. Related work

Delivering personalized information to the user depending on the current situation and user itself, implies system to be aware of its context. The research area of context-aware systems is broad, so is the widely accepted definition of the context provided by Abowd and Dey [1]: context is any information that can be used to characterize the situation of an entity, where entity could be an application, physical environment or person. There are numerous surveys related to different aspects of context-aware systems ([2], [3], [4], [5]). Smart homes is one of the widely used use cases for research of context-aware systems, since it includes all mentioned above context entities. The goals of incorporating context-awareness into the system could vary from energy efficiency of smart homes [6] to personalized service providing [2]. Since information management system should provide user with the information depending on current context, context-aware systems and proposed solutions were reviewed in the field of smart home environments and health care domain.

According to [5], where key-value, markup, graphical, object-oriented models and ontologies were compared as knowledge models for context-aware systems, ontologies were proposed to be the most expressive models with fulfillment of most of the indicated by authors requirements. Various implementations of context-aware systems with ontologies exist. For example, in [7] ontology for smart home is proposed, which includes models for location, activity, preferences of the user to understand current context. However this ontology does not include any inference with dynamic data, and could be used only for static relationships description. In [8] Semantic Web models are used for expressing relationships between situation and actions in the hospital use case in order to provide important information to the doctors. However the system does not consider other groups of users (like patients) and do not have any mechanisms to handle those groups. In [9] authors use ontologies to match user preferences to current location of the user and establish correspondent environment (lights, sound). Another example with ontology usage is presented in [10], where authors attempt to describe fire alarm situation in the model and define user's health level. Ontological approaches are also popular for other domains, for example, in [11] authors propose the use of ontologies for enabling rapid reconfiguration of manufacturing systems; in [12] authors analyze context by using ontology-based model on top of flexible manufacturing system control platform. Some of the works use hybrid approaches to define current context. For example, in [13] context-aware activity recognition was implemented with usage of semantic technologies and machine learning techniques.

All of the mentioned above works claim the feasibility of their approaches with respect to formulated objectives. Our contribution to the context-aware systems is in the proposed ontology for flexible information management, which respects user context and his role in the system as well as performs simple reasoning about the current context. The proposed solution is illustrated with explanatory examples and is supported by implementation approach described in further sections. Next section explains smart home domain requirements, which should be considered for information management solutions.

III. Domain requirements

This section presents in brief requirements, which should be considered during the development of information management system for smart homes domain with social and health care services. The main features of such domain are heterogeneous and dynamic data from various data sources, which are incorporated into distributed system, and different user groups with specific interest and information needs, who are using this system. The following requirements of context-aware information management system could be formulated.

Heterogeneous data: the system should be able to deal with different data sources. Mechanisms for seamless integration of different data sources into information management system should be provided.

Data connections: the system should provide logical connections between data sources in order to understand relevancy of the information to particular user and situation.

Time notion: to analyze *current* situation, system should incorporate the notion of time in the knowledge model. Moreover, the sequence of events could also compose and describe situation, which also shows a need in the time.

User profiles: in order to keep the knowledge about the user and enhance information management, the system should be aware of its users.

Configuration capabilities: the considered domain is smart homes with social services. Thus, new users appear more often with comparison to usual smart home (the service staff category tends to change more frequently than ordinary inhabitants of smart homes). Thus, the platform of the system should provide means for system configuration, e.g. for introducing new users with access policies and user roles.

These requirements form the basis for choosing appropriate methods for system realization. Ontological approach is seen as suitable solution since it provides means for data management with focus on data itself instead of the dealing with application specific logic. Ontologies are seen as most promising data models with comparison to database schemas in terms of flexibility of data relationships. Moreover, ontologies could be easily concealed under the user interface and managed by general installer with comparison to machine learning techniques, where specific knowledge is needed for adjustment and usage.

IV. Information Management System: General Architecture

This section explains usage of ontologies in proposed information management system. The main purpose of this system is supply user HMI with right information depending on the current situation and user itself. In order to understand this current situation, the model of the system is needed, which would combine available data sources and make meaningful connections between them. However, model along could not provide information filtering. Relevant data selection is done with a set of queries, which are run over the model.

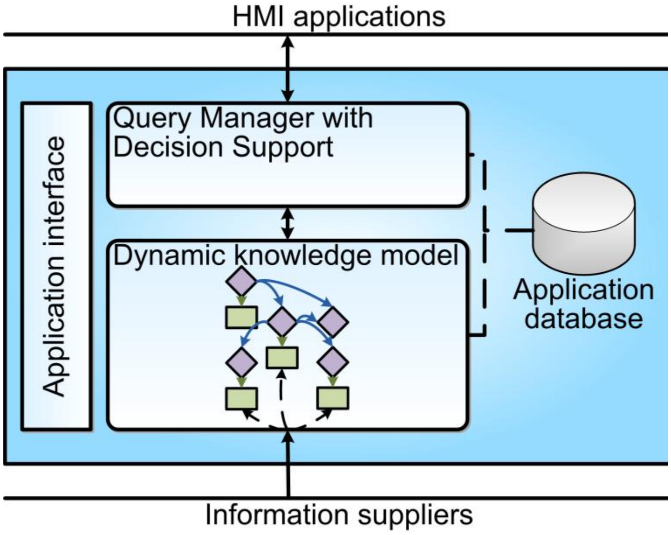


Figure 1: Information management system: general architecture

Figure 1 represents general architecture of information management system. Ontologies are used as a knowledge model of the system. They are supplied by different sources of data (sensors, databases, the Internet, external systems). Combination of dynamic data from *information suppliers* and ontologies with extensions in the form of SWRL rules [14] results in appearing new relationships in ontology, which are relevant to current context. Moreover, initial structure of ontology could incorporate relationships, which explain the relevance of specific information to certain users. Ontologies with data from information suppliers, and SWRL rules form the *dynamic knowledge model* of the system.

To filter relevant information from available knowledge model *query manager* should be implemented. Query manager indicates active user of the system, then performs a set of SPARQL queries [15] for defining currently needed information, and supplies it to the HMI for displaying.

Application database holds ontological schema with necessary individuals and predefined set of queries. This database is used during the launching of the system in configuration process.

Application interface is used by system installer during the integration of information management system with HMI applications. The interface makes configuration process of the system available to general system installer, who is unfamiliar with ontologies. More information on function blocks of the system and enabled technologies could be found in [16].

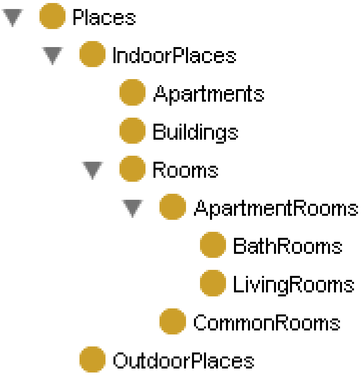
The main component of the described system is dynamic knowledge model with ontology as a central element. Next section presents developed ontology with explanatory examples.

V. Ontology for Smart Home with Social Services

As it was mentioned above, ontologies provide meaningful connections between available data, which facilitate data management process in the system. This section describes the developed ontology with explanatory examples on the usage of this ontology in information management system. It is worth to notice, that in Semantic Web it is highly recommended to adopt developed ontologies for new applications in order to support cross domain and cross application knowledge sharing. However vast adoption of existing ontologies in most cases leads to the problem of ontology integration due to the conflicts in entities and relationships naming and meaning. Since smart home applications are quite closed in nature (it is not supposed, that knowledge model of the application will be used by external programs of third parties) and available data from environment heavily depends on the actual implementation of smart homes, it was decided to develop new application ontology with some parts adopted from existing solutions.

A. Main classes and relationships

To answer the questions of what is the right information and who is the right user, the model should incorporate the notion of users and available data from sensors. Moreover, to analyze current context in terms of information importance and relevance to the certain users the concept of alarms in the system and users/devices location is needed. Thus, the main taxonomies of ontology for information management would be: devices, places, users and alarms.



- Places
  - IndoorPlaces
    - Apartments
    - Buildings
    - Rooms
      - ApartmentRooms
        - BathRooms
        - LivingRooms
      - CommonRooms
  - OutdoorPlaces

Figure 2: Places taxonomy for smart home domain ontology

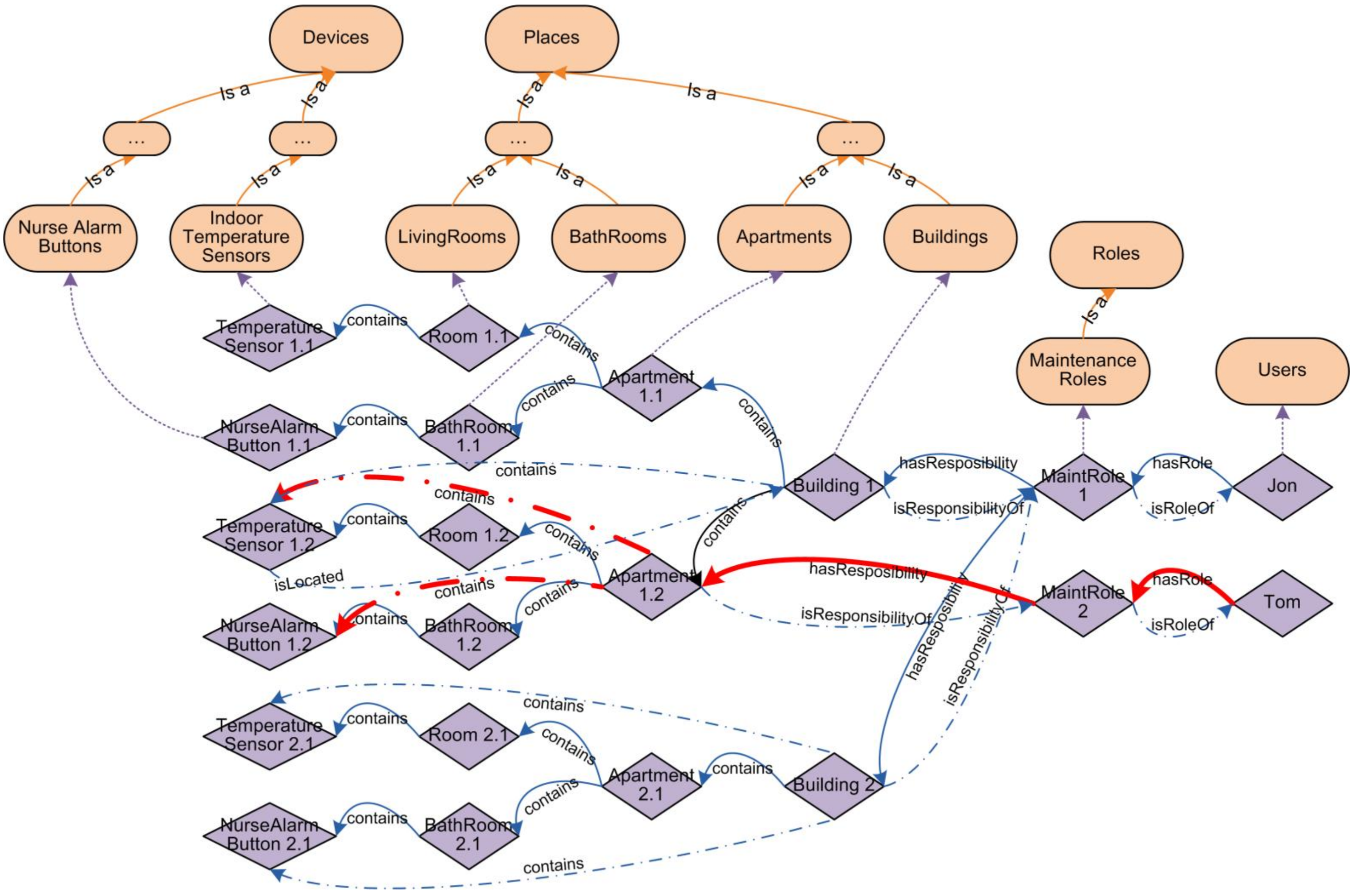


Figure 3: Part of ontology explaining information filtering scenario

The *device* taxonomy includes notion of all available devices in the system (including wearable by the user sensors) with relationships describing the current location of device in the building (or on the user). Another type of important relationships for device taxonomy is the datatype relationship, containing the current measured value from device (e.g. temperature, door lock state, IR sensor state), which could be used to analyze current context.

Places taxonomy defines locations in and outside of the smart home with possible extensions depending on the certain implementations. Two main properties, which connect places with each other, are transitive properties *isLocated* and *contains*, which are inverse properties at the same time. These properties enable binding of data from devices to certain locations. The example of places taxonomy is presented in Figure 2.

User taxonomy incorporates the notion of the users of the system. It is also accompanied by the *roles* taxonomy with relationships addressing main responsibilities and field of interest for certain users.

Alarms taxonomy represents the notion of possible alarms in the system. When condition is met, the alarm is activated by providing new relationship in the ontology (more information is provided with fire alarm scenario).

Instantiating mentioned above taxonomies with individuals and their relationships results in knowledge model of the system with meaningful connections between available data and presented users. To analyze current context and trigger correspondent alarms the processing of the data is needed, which could be done with SWRL rules. Simple explanatory scenarios are provided in the next subsection to illustrate the proposed approach with ontologies.

B. *Scenarios*

The use case, which is considered for information management system, is a senior house with social services. It consists of several buildings with twelve apartments in each building. The users of the systems are divided in three main groups: inhabitants, nurses, maintenance personnel. Inhabitants should receive information relevant to their own apartments while nurses and maintenance personnel should be aware of the situation in all buildings with the focus on wellbeing of inhabitants (nurses) and device status (maintenance). The following scenarios will illustrate main principles of information management approach in senior house.

1) *Information filtering scenario*

This scenario addresses the capability of the system to filter relevant to the user information. Thus, ontology should capture available information with the connections to certain users. The partial illustration of developed ontology is represented in Figure 3. Taxonomy structure (classes of ontology) is represented in rounded rectangles. Individuals of classes are represented in diamonds. For simplicity it was assumed that there are two buildings in the field (*Building 1*, *Building 2*) and they have two and one apartments correspondently. Each living room inside the apartments has temperature sensor and each bathroom has nurse alarm button. *Contains* object property describes relative locations of all objects in the model (devices, buildings, apartments). It is also a transitive property; it means that with application of inference engine additional *contains* properties will be inferred (part of them are shown with dashed lines), which are assumed implicitly. Users of the system are individuals of the *Users* class of the ontology, and they have roles in the system, which are connected to the users via *hasRole* and *isRoleOf* inverse properties. Each role individual assumes having responsibility via *hasResponsibility* and *isResponsibilityOf* inverse properties. Organizing connections in this way make it possible to deliver personalized information to the user. For example, we could use SPARQL queries in order to ask the model for devices, which are under the responsibility of user Tom, who is maintenance personnel in the field. The query for this example is:

```
SELECT ?device
WHERE
 { foaf:Tom foaf:hasRole ?role .
  ?role foaf:hasResponsibility ?responsibility .
  ?responsibility foaf:contains ?device .
  ?device rdf:type foaf:Devices . }
```

This query will result in patterns matching, which are indicated in bold in Figure 3; and the list of devices, which are in *Apartment 1.2* will be returned. If some new device will be installed in *Apartment 1.2* and be presented in knowledge model, then the same query would return also this new device without additional query modifications.

Thus, making ontology with presented structure and performing SPARQL queries over this structure it is possible to personalize information and reduce informational load by filtering out irrelevant to particular user data.

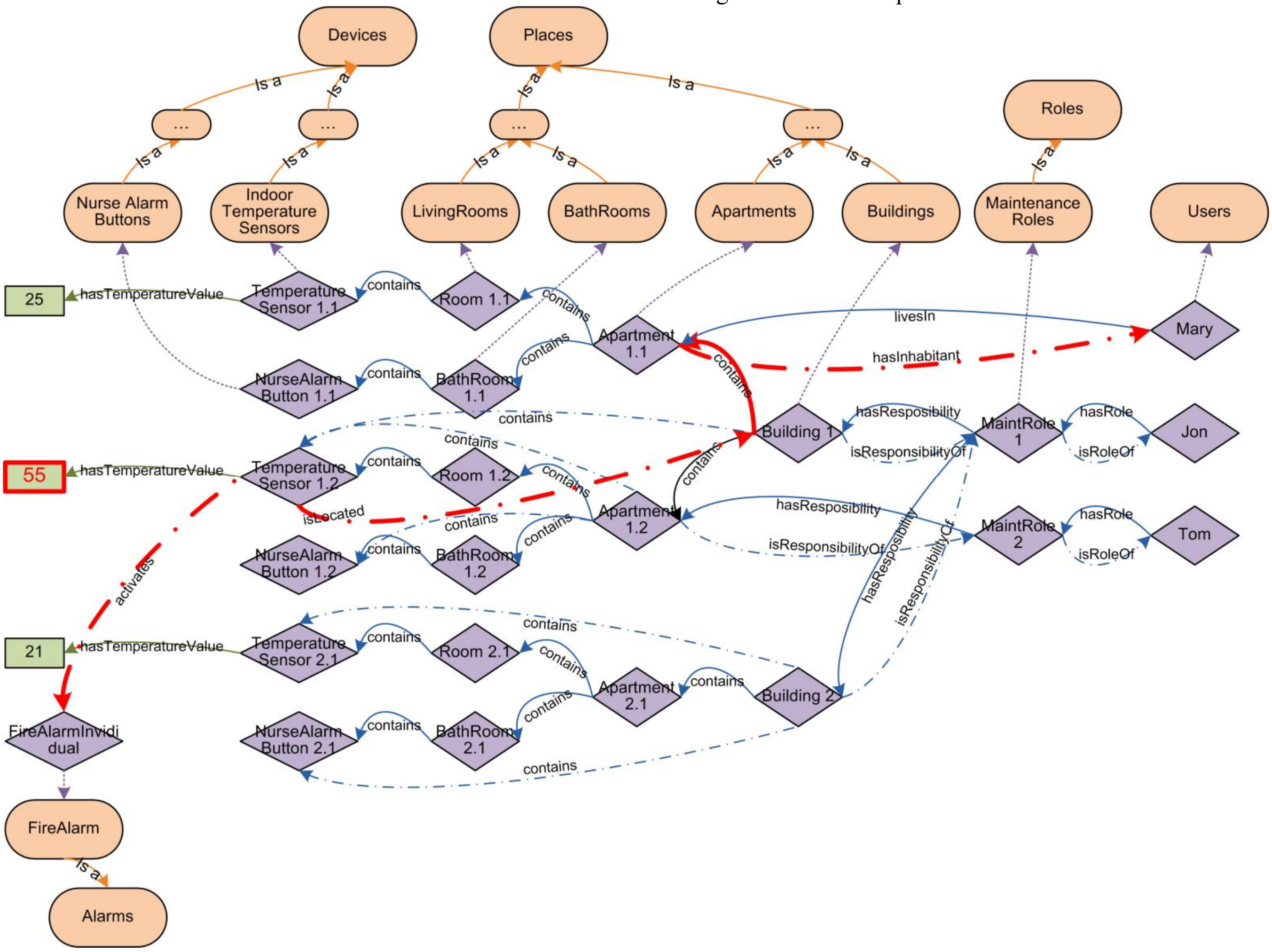


Figure 4: Part of ontology explaining fire alarm scenario

2) Fire alarm scenario

This scenario addresses context-aware capability of the system, when information from sensors is analyzed, and conclusions about the current situation are made in the form of warning and alarm messages, which should be delivered to the user. To illustrate context-awareness, the scenario simulating fire alarm from temperature sensors is considered. Previous example was extended with minor additions. Those additions include datatype properties of *IndoorTemperatureSensors* class individuals connected via *hasTemperatureValue* relationship; new *Alarms* class in the ontology; and additional instance representing inhabitant (*Mary* individual), who lives in the *Apartment 1.1* (Figure 4). To analyze current situation in the environment, the model should be able to react on the changes from environment sensors. For this purpose, sensor data from the temperature sensors is mapped into the literal values of the ontology (*hasTemperatureValue* datatype property of ontology). Furthermore, to react on the higher temperatures and create fire alarm in the system, the following SWRL rule is implemented, which produces *activates* property from temperature sensor to *FireAlarmIndividual* if temperature is higher than 40 degrees:

```
IndoorTemperatureSensors(?sensor) ∧
hasTemperatureValue(?sensor, ?value) ∧
swrlb:greaterThan(?value, 40) →
→ activates(?sensor, FireAlarmIndividual)
```

In the example, represented in Figure 4, *Apartment 1.2* has temperature 55 degrees. Applying the reasoner to this ontology causes *activates* relationship appearing from *TemperatureSensor 1.2* to *FireAlarmIndividual*. Thus, ontology reflects current situation in the environment. To produce alarm messages in the system, SPARQL queries are run over the described ontology to check active alarms in the system. For example, next query returns list of users, who live in the building, where currently the fire alarm is active (thus making it possible to notify correspondent users). If there is no any active alarms in the system (no *activates* relationships between *FireAlarm* class individuals and devices) then query will not be satisfied, and no users will be returned as a result of the query (no fire alarm notifications should be made).

```
SELECT ?inhabitants
WHERE
 { ?device foaf:activates ?alarm .
  ?alarm rdf:type foaf:FireAlarm .
  ?device foaf:isLocated ?building .
  ?building rdf:type foaf:Buildings .
  ?building foaf:contains ?apartment .
  ?apartment foaf:hasInhabitant ?inhabitants .}
```

This query matches the patterns indicated in bold in Figure 4, and will return Mary instance, indicating that she should be notified about the situation. The similar queries could be implemented asking the handler of fire alarm.

Thus, applying SWRL rules to the model, and performing correspondent SPARQL queries it is possible to make system context aware and reacting to the changes in the environment on the knowledge model level without coding all mentioned above relationships and dependences into the application. The next section describes an implementation approach of proposed information filtering system.

VI. Implementation

According to Figure 1 proposed information management system has two main blocks: *Dynamic Knowledge Model*, and *Query Manager with Decision Support*. To keep the structure of the system modular and open for implementations in other domains, it was considered to develop two separate applications for these blocks with communication via Web Services. The proposed architecture of information management system allows reusing some of components developed earlier for production system control [17]. Thus, *Dynamic Knowledge Model* block is implemented as *OntologyService* application, which holds the model of the system, and *OntologyManager* was adopted for mapping data from sensors into the ontology model [17]. The general implementation concept is represented in Figure 5. All applications are implemented on the server side (different servers could be used to increase computational capability of the system). Data could be gathered from different sources of information, for example, wireless sensors with WS capabilities and databases. *OntologyManager* subscribes to WS events from required sources of data and invokes SPARQL/Update [18] services of *OntologyService* for updating information in the ontology. *OntologyService* contains the ontology of the system, performs reasoning and provides services for external applications for enabling SPARQL and SPARQL/Update queries. At last, *query manager* uses services of *OntologyService* and retrieves data from ontology, which should be displayed in the HMI. All applications are developed in Java with usage of Jena framework. The Pellet engine was used in order to perform ontological inference in the *OntologyService*. The proposed architecture uses open standard technologies, which ensure feasibility of the solution.

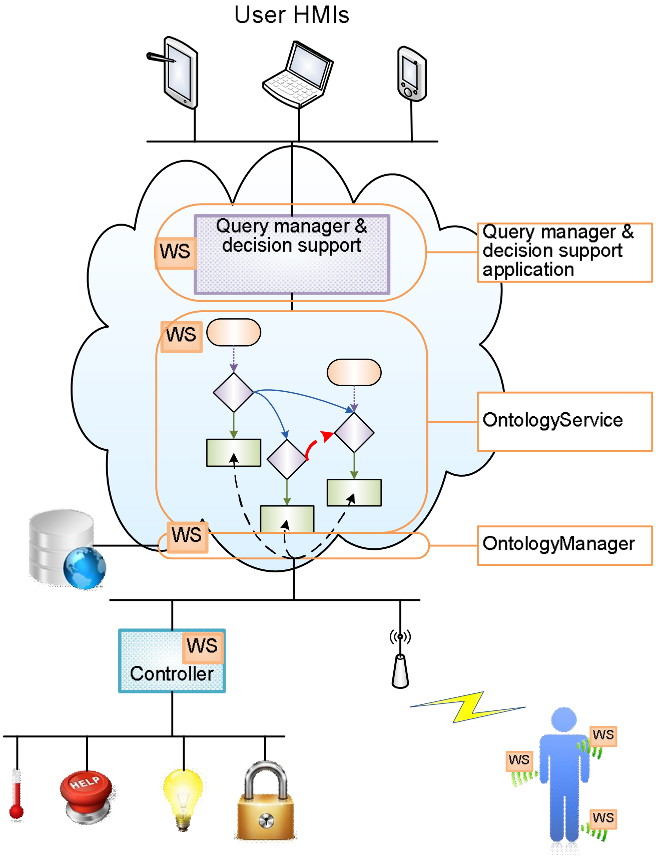

User HMIs
WS
Query manager & decision support
Query manager & decision support application
WS
OntologyService
WS
OntologyManager
WS
Controller
HELP
WS
WS
WS

Figure 5: Implementation concept of information management system

VII. Conclusions

This paper proposes information management system, which delivers personal information to the user based on the user role in the system and current context. The system takes advantage of Semantic Web technologies and decouples data from application specific logic. This decoupling leads to the flexible information management solution with possibility to extend system and introduce new user roles, responsibilities and context analysis. Current information management approach is seen as domain independent although the domain in this paper is restricted due to the testing purposes. The simple scenarios with implementation prototype are considered proving the feasibility of the approach. Future research will focus on extension of considered scenarios and applying approach in other information intensive domains.

REFERENCES

[1] G. Abowd and A. Dey, "Towards a Better Understanding of Context and Context-Awareness," *Lecture Notes in Computer Science,* vol. 1707, pp. 304-307, 1999.

[2] G. M. Kapitsaki, G. N. Prezerakos, N. D. Tselikas, and I. S. Venieris, “Context-aware service engineering: A survey,” *Journal of Systems and Software*, vol. 82, no. 8, pp. 1285–1297, Aug. 2009.

[3] M. Miraoui, C. Tadj, and C. Amar, “Architectural Survey of Context-Aware Systems in Pervasive Computing Environment,” *Ubiquitous Computing and Communication Journal*, vol. 3, no. 3, pp. 68–76, 2008.

[4] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.

[5] C. Hoareau and I. Satoh, “Modeling and Processing Information for Context-Aware Computing: A Survey,” *New Generation Computing*, vol. 27, no. 3, pp. 177–196, Aug. 2009.

[6] Y. Cheong, Y.-J. Kim, S. Y. Yoo, H. Lee, S. Lee, S. C. Chae, and H.-J. Choi, “An Ontology-Based Reasoning Approach Towards Energy-Aware Smart Homes,” in *IEEE Consumer Communications and Networking Conference*, 2011, pp. 850–854.

[7] T. Van Nguyen, W. Lim, H. A. Nguyen, D. Choi, and C. Lee, “Context Ontology Implementation for Smart Home,” in *2nd International Conference on Ubiquitous Information Technologies & Applications*, 2007.

[8] A. Esposito, L. Tarricone, M. Zappatore, L. Catarinucci, R. Colella, and A. Dibari, “A framework for context-aware home-health monitoring,” *Ubiquitous Intelligence and Computing*, vol. 3, no. 1, pp. 75–91, 2008.

[9] M. Sutterer, O. Droegehorn, and K. David, “User Profile Selection by Means of Ontology Reasoning,” in *IEEE Fourth Advanced International Conference on Telecommunications*, 2008, pp. 299–304.

[10] V. Huang and M. K. Javed, “Semantic Sensor Information Description and Processing,” in *IEEE Second International Conference on Sensor Technologies and Applications*, 2008, pp. 456–461.

[11] J. L. Martinez Lastra and I. M. Delamer, “Ontologies for Production Automation,” *Advances in Web Semantics I*, vol. LNCS 4891, pp. 276–289, 2008.

[12] M. K. Uddin, J. Puttonen, S. Scholze, A. Dvoryanchikova, and J. L. M. Lastra, “Ontology-based context-sensitive computing for FMS optimization,” *Assembly Automation*, vol. 32, no. 2, pp. 163–174, 2012.

[13] D. Riboni and C. Bettini, “Context-Aware Activity Recognition through a Combination of Ontological and Statistical Reasoning,” *Ubiquitous Intelligence and Computing*, vol. 5585 LNCS, pp. 39–53, 2009.

[14] J. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML,” 2004. [Online]. Available: http://www.w3.org/Submission/SWRL/. [Accessed: 12-Dec-2012].

[15] “SPARQL Query Language for RDF,” 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/. [Accessed: 12-Dec-2012].

[16] Y. Evchina, A. Dvoryanchikova, and J. L. Martinez Lastra, “Ontological Framework of Context-Aware and Reasoning Middleware for Smart Homes with Health and Social Services,” in *IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 985–990.

[17] J. Puttonen, A. Lobov, M. A. Soto Cavia, and J. L. Martinez Lastra, “A Semantic Web Services-based approach for production systems control,” *Advanced Engineering Informatics*, vol. 24, no. 3, pp. 285–299, Aug. 2010.

[18] A. Seaborn, G. Manjunath, C. Bizer, J. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo, and B. Nowack, “SPARQL Update A language for updating RDF graphs,” 2008. [Online]. Available: http://www.w3.org/Submission/SPARQL-Update/. [Accessed: 12-Dec-2012].

Citation and link to access:

Evchina, Y., Dvoryanchikova, A. & Martinez Lastra, J.L., 2013. Semantic information management for user and context aware smart home with social services. In *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*. Ieee, pp. 262–268. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6523856.

Publication III

Evchina, Y., Puttonen, J., Dvoryanchikova, A., & Martinez Lastra, J. L. (2015). Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. *Engineering Applications of Artificial Intelligence*, *43*, 111–126. https://doi.org/10.1016/j.engappai.2015.04.008

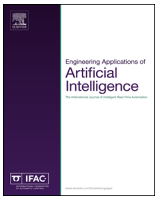
The original print of this publication is available online (DOI: 10.1016/j.engappai.2015.04.008)

Engineering Applications of Artificial Intelligence 43 (2015) 111–126

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems

Yulia Evchina*, Juha Puttonen, Aleksandra Dvoryanchikova, José Luis Martinez Lastra

Tampere University of Technology, FAST Laboratory, Korkeakoulunkatu 5L, FIN-33720 Tampere, Finland

ARTICLE INFO

Article history:
Received 16 January 2015
Received in revised form
10 April 2015
Accepted 13 April 2015

Keywords:
Context-aware systems
Information filtering
Semantic Web technologies
Ontologies
Building automation

ABSTRACT

Multiple embedded devices in modern control and monitoring systems are able to sense different aspects of the current context such as environmental conditions, current processes in the system and user state. The number of captured situations in the environment and quantity and variety of devices in the system produce considerable amounts of data, which should be processed, understood and followed by corresponding actions. However, fully delivered to the user regardless of their role in the system and needs, data flows cause cognitive overload and thus may compromise the safety of the system depending on the timely response of the operators. This paper addresses the problem of selective information delivery with respect to the user's role in the system, his needs and responsibilities, by proposing context-aware information management middleware. The system utilizes Semantic Web technologies by capturing relevant information in the knowledge model of the system, which decouples data from the application logics. A clear division of data and application logics enables context-awareness and facilitates the reconfiguration process, when new information should be added into the system. The chosen approach is justified with an analysis of main trends in context-aware solutions. The engineering principles of the knowledge model are described and illustrated with simple scenarios from the building automation domain. The prototype developed proves the feasibility of the approach via performance evaluation and demonstrates the reconfiguration capabilities of information flows in the system. Further work assumes the extension of the knowledge model and integration of the system with adaptive human–machine interfaces for multi-role and multi-user environments.

1. Introduction

The evolution of control and monitoring systems has enabled information accessibility for almost any aspect of the environment (Sauter et al., 2011). In modern systems it is possible to integrate large numbers of various sensors to capture environmental conditions (e.g. temperature, humidity, and light), particular features of the process of interest (e.g. the product status in the production line) and the state of the users involved in the process or system operation. Potentially, it could offer advantages for timely situational awareness of the all system users. In reality growing information complexity challenges the usability of the system, human performance, and safety overwhelming the cognitive capacities of a user. This paper addresses a problem of selective information delivery to different users of the system by proposing context-aware information management middleware (IMM), which is able to analyze the run-time data of the system and deliver the information which is relevant to the user given the current situation and user needs/responsibilities.

The aim is to bring the right information to the right user at the right time, thus information management middleware should be able to determine the "rightness" according to the situation. The introduction of context-awareness is claimed to enable the desired flexibility of system behavior (Baumgartner et al., 2010; Fischer, 2012). The context can be defined as "*any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object*" (Abowd and Dey, 1999), and then the system is context-aware "*if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*" (Dey, 2001). Both definitions are quite generic and allow different technological solutions for context-aware systems providing the right services (Emmanouilidis et al., 2013), actions (Feng et al., 2009) or system adaptation.

This paper proposes context-aware information management middleware for data intensive monitoring systems, where user is exposed to numerous sources of data coming from heterogeneous devices of the system and various situations of the environment.

* Corresponding author. Tel.: +358 408490281.
E-mail addresses: yulia.evchina@tut.fi (Y. Evchina), juha.puttonen@tut.fi (J. Puttonen), jose.lastra@tut.fi (J.L.M. Lastra).

http://dx.doi.org/10.1016/j.engappai.2015.04.008
0952-1976/© 2015 Elsevier Ltd. All rights reserved.

The middleware provides the user with the relevant information depending on the current situation, user role in the system, his needs and responsibilities. The main technological trends of achieving context-awareness in the systems are presented and analyzed, and the chosen approach using Semantic Web technologies (specifically OWL ontologies, SPARQL and SPARQL/Update[1] queries) is justified. The system utilizes a reconfigurable knowledge model (ontologies) which decouples data from the application logic. The clear division of data and application logic enables context-awareness and (1) facilitates the system reconfiguration process when new information is introduced into the system; (2) enables the implementation of domain independent tools which facilitate the process of adapting IMM in new domain. The proposed approach is supported by ontological engineering principles, a description of a prototype already implemented, and describing reference cases from the building automation domain.

The rest of the paper is organized as follows. Section 2 overviews the state of the art for context-aware systems and justifies the choice of Semantic Web technologies for the solution proposed. Section 3 describes the functional architecture of the system and the enabling technologies for the main architectural blocks. Section 4 states the main design principles of the knowledge model of the system, the query templates enabling reconfiguration and the domain independent capabilities of the solution. Section 5 describes the tools for implementing IMM in detail and is followed by section 6 with simple examples of scenarios from the building automation domain exemplifying the knowledge model and demonstrating in function the main principles of the proposed solution. Section 7 presents performance evaluation of developed prototype; and finally, Section 8 concludes the paper and highlights directions for future work.

2. State of the art

Context-aware systems are a fairly active research topic resulting in numerous solutions and application areas. In order to limit the scope during the literature analysis, the focus was placed on the solutions with use cases similar to data intensive monitoring systems (i.e. with active and numerous incoming data flows) and selective reaction to the present context (i.e. delivery of personalized services or performing particular actions rather than activity recognition). Thus, other features of context-aware systems, such as learning mechanisms and analyzing of historic data are beyond the scope of the paper. On the basis of these criteria three main trends in technological solutions of context-aware systems were identified: Complex Event Processing (CEP), Semantic Web technologies, and Linked Stream Data. This section analyzes the advantages and disadvantages of the approaches in depth, and justifies the choice of the Semantic Web approach for the development of the context-aware information management middleware.

2.1. CEP technologies

CEP is a method for analyzing information flows (events) and matching them with predefined patterns (rules). Context-aware systems with CEP capture relevant information in the rules, that is, they model the context via the rule condition part and provide context-sensitive reactions via the rule action part (Chung et al., 2013; Ottenwälder et al., 2014). Usually the matching is realized via standalone engines, which are configured via the engine specific rules and perform run-time event analysis (Gao and Bhiri, 2012). The existing engines are quite numerous, and many are freely available on the web (e.g. Drools Fusion,[2] Esper,[3] Storm,[4] Oracle CEP,[5] SASE,[6] etc.) and could be easily adapted to applications via provided API. The engines provide scalable infrastructure for analyzing streams of different events, and this ability helps to decouple the application logic from the business rules. The CEP context-aware applications are much used, especially in “sense and respond” systems (Zappia et al., 2012). For example, Drools was applied to understand the current context on the basis of readings from RFID marked devices in a hospital in (Yao et al., 2011). The data from the sensors were analyzed by the CEP engine to identify critical situations in health monitoring. CEP engines enable the collecting of the row and hard interpretable sensors data with RFID tags, and the interpretation and extraction of meaningful events pertaining to the patient. In automotive domain the CEP engine was used for data fusion from different sensors in a car (Terroso-Saenz et al., 2015). The approach made it possible to interpret the context of the route, driver, and travel conditions corresponding to the traveler's needs.

The CEP approach has several drawbacks. For example, the large number of rules in the applications requires supplementary rule consistency check (Weiss et al., 2011). The ontologies were used for consistency checking of rule databases (Gellrich et al., 2012), however the approach is still under investigation and the reported prototype was not yet clearly exemplified. Another problem is in the software interoperability, which impedes the reconfiguration process as far as the rules for CEP engines can be expressed in different languages (Eckert et al., 2011). A lightweight solution to CEP was applied to minimize the dependencies on external libraries and applications in order to focus on ease-of-use, extensibility and scalability; nevertheless the solution still uses engine-specific language for rule expressions (Zappia et al., 2012). There have been attempts to produce a description of CEP rules using ontologies. For example, CEP was enriched with an ontological description of events (Taylor and Leidinger, 2011), however the model does not consider other context information such as places and users and it is not clear how the system will react to changes. To summarize, the CEP approach is a powerful way of analyzing intensive streams of data, but it still has its difficulties: CEP engines have different languages for rule expressions, which require additional effort in engine adoption and result in complex reconfiguration processes during the system adjustment, especially if the knowledge base of rules is large and difficult to understand.

2.2. Semantic Web technologies

The matching of complex event patterns with CEP in an event cloud is able to anticipate only the context defined in the rule. If a pattern is not described with a rule, it is beyond further consideration. In other words, with CEP it is not possible to derive implicit knowledge from the explicit statements. Another breed of applications attempts to bridge this gap by capturing the current context in the knowledge model of the system, which helps to reason over the present situation in the environment. There are various technologies supporting context modeling such as key-value, markup, graphical, object-oriented and ontological models (Hoareau and Satoh, 2009; Bolchini et al., 2007). Context modeling with ontologies is seen as the most expressive way in terms of ability to capture knowledge of the environment,

[1] SPARQL Update: http://www.w3.org/Submission/SPARQL-Update/.

[2] Drools Fusion: http://www.jboss.org/drools/drools-fusion.html.

[3] Esper: http://esper.codehaus.org/.

[4] Storm: http://storm-project.net/.

[5] Oracle CEP: http://docs.oracle.com/cd/E13157_01/wlevs/docs30/get_started/overview.html.

[6] Stream-based And Shared Event processing (SASE): http://sase.cs.umass.edu/.

formalism, reusability, and understandability by humans and machines (Hoareau and Satoh, 2009). The standardization activities on OWL ontologies and related query techniques have triggered an active development of new applications with the usage of Semantic Web technologies including context-aware systems.

Semantic Web technologies were applied to context-aware systems in many application domains: road traffic management (Baumgartner et al., 2010), business meeting assistance (Her et al., 2010), tourist services (Moreno et al., 2013), home health monitoring (Esposito et al., 2008), smart homes (Nguyen et al., 2007; Santofimia et al., 2011), manufacturing (Khilwani et al., 2009; Puttonen et al., 2010; Uddin et al., 2012), and personalized services (Barros et al., 2011; Chernbumroong et al., 2013; Furno and Zimeo, 2014; Hong et al., 2009; Rodriguez et al., 2012; Viktoratos et al., 2014) to name a few. In general, all the mentioned applications use ontologies to capture the relevant context in a model. Next the model is queried to advise on actions to be performed (Baumgartner et al., 2010; Esposito et al., 2008; Puttonen et al., 2010), or on what service should be delivered in respect to the current context (Chen et al., 2007; Her et al., 2010; Hong et al., 2009). Often a combination of several ontologies is used to describe the application specific context (Rodriguez et al., 2012; Miguelanez et al., 2011). For example, the needs of a meeting assistance application were described in the ontological models of users, devices and locations (Her et al., 2010). Certain applications require additional ontologies which capture the features of activities, preferences (Nguyen et al., 2007; Rodriguez et al., 2012) or tasks (e.g. with Task Model Ontology[7] or application specific solutions).

Ontologies are flexible to model almost any information and relationships among the entities of a domain. However, ontologies were originally developed for the expression of mostly static information in the Web, and they are challenged in their ability to express the temporal relationships between the ontological elements (Baumgartner et al., 2010). The problem of expressions of temporal relationships was approached from different positions. Some of the solutions propose an introduction of ontologies for the notion of time into the knowledge model (Krieger, 2010; Hobbs and Pan, 2004). These solutions are heavy in representations and they rather describe the relationships than analyze and interpret them. Another approach is to develop a lightweight temporal model with a set of tools enabling temporal reasoning among the concepts (O'Connor and Das, 2011), which binds the application to certain type of temporal ontology and supported concepts in it. Alternatively the simple temporal relationships can be supported with the Semantic Web Rule Language (SWRL[8]), which offers an opportunity to express custom relationships and rules and to analyze temporal relationships with built-ins for comparison.[9]

2.3. Linked Stream Data

The lack of sufficient temporal reasoning over the highly intensive streams of semantic data led to an emergence of the Linked Stream Data concept to meet the needs of highly dynamic applications (Sequeda and Corcho, 2009; Teymourian and Paschke, 2009). The technology is based on assigning URIs to data streams represented via RDF and linking the URIs to corresponding semantic models. Linked Stream Data is supported in the form of engines and query languages enabling continuous querying[10] and complex temporal reasoning over the stream data. In recent years several engines have been developed with Linked Stream Data processing support (Hasan et al., 2011; Komazec et al., 2012; Le-Phuoc et al., 2012a, 2012b). These engines make it possible to analyze data in time slots, to aggregate data within a certain time interval, and to support negation operations (e.g. to check whether these types of instances are not presented in data streams). The communication with engines is realized via C-SPARQL, which is a newly developed continuous semantic language. In comparison to the CEP approach Linked Stream Data engines are capable of combining data processing and knowledge model analysis, which claims to derive implicit knowledge from explicit statements. However the technology is still immature and the tools mainly support the temporal reasoning over RDF models without support for OWL expressivity (Le-Phuoc et al., 2012a; Hasan et al., 2011). The need for deriving implicit knowledge over the streams of data was recognized by Komazec and Cerri (2011) and was addressed by developing the Sparkwave engine (Komazec et al., 2012) with the support of RDFS entailments and a few OWL constructs (*owl:inverseOf* and *owl:SymmetricProperty*). The engine has demonstrated the possibility of combining semantic reasoning with temporal data on a limited set of reasoning constructs.

2.4. Assessment of the background

A solution for personalized information delivery should comply with the following requirements. First, it should enable flexible modeling of the data relevant to the user's information needs. Second, due to the system scale and possible changes the solution should allow means for convenient (i.e. easily understandable by humans) system (re)configuration. The modern CEP approach enables processing the intensive streams of data indicating application dependent patterns of interest. However the rule-based context modeling is not capable of inferring implicit data and in general the approach requires engine dependent skills for the reconfiguration process. Linked Stream Data is concerned with context modeling and data streams processing; yet the technology is still immature and currently supports only temporal data processing without semantic reasoning. Semantic Web technologies allow expressive context modeling via ontologies and SWRL rules. Ontologies are also both human and machine understandable and could therefore be used to facilitate the reconfiguration process of the system. The lack of sophisticated temporal reasoning can be compensated in a manner sufficient for the scope with application of the SWRL comparison built-ins and SPARQL/Update language (the latter is to provide the sensor data for the ontological model of the system). These features are the bases for the choice of Semantic Web technologies for the context-aware information management middleware, and have enabled the reconfiguration capabilities of the solution.

The next section presents the conceptual architecture and lists the principal technologies enabling the proposed solution.

3. Building blocks of IMM and associated technologies

IMM is seen as intermediate layer between information sources of the system (such as physical sensors, databases, and applications) and information consumers (such as HMI builders, history

[7] Task Model Ontology (TMO): http://www.semanticdesktop.org/ontologies/2008/05/20/tmo/.

[8] Semantic Web Rule Language (SWRL): http://www.daml.org/rules/proposal/.

[9] SWRL built-ins for comparison: http://www.daml.org/rules/proposal/builtins.html#8.1.

[10] Continuous querying is the mechanism of constant evaluation of query pattern when new data is added to the system and updating the corresponding result set as an opposite to traditional queries when a query should be invoked explicitly by the caller and the result set remains the same regardless of the new data in the model.

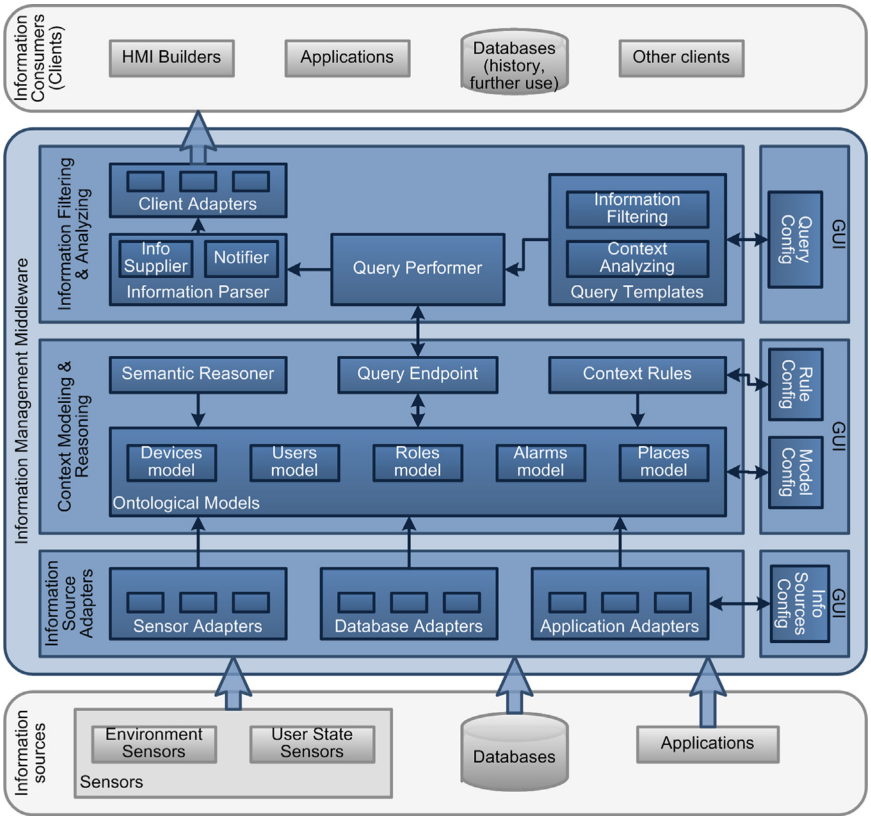


Fig. 1. Functional blocks of information management middleware.

databases, and other client applications) (Fig. 1). The objectives of information management middleware are (1) to analyze data from information sources in order to deliver the information to the end users in a personalized manner during run-time, (2) to send the event notification messages to the different groups of users with respect to their informational needs; and (3) to facilitate the reconfiguration process for the technical administrators of the system. The middleware has a layered structure, which is quite a common approach in designing context-aware systems (Ailisto et al., 2002; Schmidt, 2014).

The key component of the proposed information management middleware is an ontological model, which captures the current context in the domain and is simple to reconfigure by non-technical personnel (*Context Modeling and Reasoning* layer in Fig. 1). Information sources supply the ontological model with new data via various adapters (*Information Source Adapters* layer). The model is queried with generalized query templates, which provide information filtering and personalized information delivery in the form of notification messages for different groups of users (*Information Filtering and Analyzing* layer). Each layer is supported with Graphical User Interface (GUI), which facilitates the reconfiguration process of the system.

3.1. *Information source adapters*

This layer serves as a data provider to the ontological model of the middleware. The middleware receives data from numerous data sources via different interfaces. The interface support is realized via dedicated data adapters. Each data adapter transforms data from specific interface to the unified format understandable by the application. The unified format for updating ontological model is the SPARQL/Update expressions, which are configured via dedicated in this layer GUI. Thus the technical administrator of IMM has two ways of configuring the *Information Source Adapters* layer: (1) plugging adapters of the information sources determining which type of interfaces will be understandable by the middleware; (2) adjusting SPARQL/Update expressions, which define how the ontological model should be updated for each type of sensors.

In Fig. 1 adapters are grouped by physical sources; however the implementation depends on the data source interface abstraction. For example, nowadays the Service Oriented Architecture (SOA) paradigm is quite a popular solution for highly distributed and decentralized systems. The solution assumes that data and some of the functionalities (blocks) of the systems are provided as a service. SOA is becoming adoptable solution in different domains from Internet of Things (IoT) (Atzori et al., 2010) to the control and monitoring systems of buildings and manufacturing (Martinez Lastra and Delamer, 2006). With the SOA approach it is possible to gain access to the applications and hardware devices via the same Web Service (WS) interface. For example, it is possible to equip legacy systems with smart RTUs (e.g. Inico S1000[11]), which enable access to the data from devices as Web Service (WS) via standard TCP/IP protocols. In that way, the same WS adapter could be used for accessing data from hardware devices and software applications.

3.2. *Context modeling and reasoning*

The next layer of the architecture is *Context Modeling and Reasoning*. The overall responsibility of the layer is to reflect the current state of the environment in a knowledge model of the system and to provide query support for the upper levels of the architecture.

[11] Inico S1000:http://www.inicotech.com/s1000_overview.html.

The knowledge model captures relevant information for personalized data delivery (*Ontological Models*): device statuses, user profiles and activities, places and their relation on the site, etc. The model is updated with the current device values from the *Information Source Adapters* layer; thus it reflects not only the concepts and their relationships in the domain, but also incorporates up-to-date data on the environment. The knowledge model is represented with OWL-DL[12] language, which allows reach expressiveness with guaranteed inferring of all presented entailments. Since the ontological models are human–machine readable, they can be quite easily understood by humans and are directly interpreted by machines. There are several open-source projects like Protégé,[13] which provide convenient interface for ontology modifications. Combining conceptual modeling with real-time data from devices makes ontological models dynamic and allows more profound analysis of the current context.

The ontological models are further enriched with *Context Rules*, which allow the expression of application specific patterns of interest (e.g. “if the temperature rises above 40°, then it is fire alarm”, “if a nurse alarm button is pressed, then it is nurse alarm”). The OWL-DL ontologies allow this extension with SWRL rules. The context rules are domain-dependent and are configured by technical administrator of the system.

The ontological inference is enabled with *Semantic Reasoner* block, which allows deriving of hidden relationships and classes for individuals of ontological models. In order to combine the analysis of ontological models with context rules the semantic reasoner should be able to process OWL-DL entailments and SWRL rules. Examples of such reasoners are Pellet,[14] OntoBroker,[15] HermiT,[16] KAON2.[17]

The *Query Endpoint* is the enabling access to the context model. The endpoint allows other applications to ask the knowledge model for specific information. The most common query language for ontologies, which is also the W3C recommendation, is SPARQL; thus, it was decided to use SPARQL query endpoint to enable communications between *Context Modeling and Reasoning* and *Information Filtering and Analyzing* layers.

3.3. *Information filtering and analyzing*

The purpose of the layer is twofold: (1) it filters information to different users of the system; (2) it sorts existing notifications in the system and composes corresponding messages to users. The source of the information is the knowledge model, which reflects the present state in the environment. The tools for information filtering and analyzing are *Query Templates*, which are run over the knowledge model and return the results depending on the current context and model of the system.

The query templates can be configured by the technical administrator of the system via GUI. There are two types of query templates: *Information Filtering* and *Context Analyzing*. Information filtering templates select data, which should be displayed to the user in the normal mode; and the query returns a list of devices with their values and locations to be displayed. Context analyzing templates ask the knowledge model about the active alarms in the system. If alarms are active, then the query returns a non-empty result with additional information about the handler of the alarm. The query templates are designed to be generic, thus they are reusable and can be applied to many domains with minor modifications. The approach makes it possible to avoid major recompilation while modifying only the knowledge model with domain specific data and unique situation in the environment.

The *Query Performer* block uses *Query Endpoint* services to ask knowledge models for the information. After receiving the results the *Query Performer* directs them to the *Information Parser* to be split into corresponding messages. *Client Adapters* receive messages and wrap them in the corresponding communication interface for information consumers.

As a summary of proposed architecture, decoupling of data from application logic is done via domain-specific knowledge models and query templates. The decoupling allows architecture reusability since the same tools can be used with the specific knowledge models for different domains. The architecture utilizes Semantic Web technologies, which are human–machine readable, and provides GUI on different levels of the architecture to facilitate reconfiguration process. The next section presents the ontological design principles in detail and the query templates adopted for the solution.

4. Design principles of ontological context modeling and queries

The proposed IMM architecture utilizes Semantic Web technologies with the focus on ontologies and queries enabling the reconfiguration capabilities of the system. In order to make generic query templates reusable in different domains, ontologies should follow certain design principles. This section serves to explain the ontological design principles and query templates adopted for the proposed solution.

4.1. *Ontological design principles*

Ontology is a knowledge representation technique which captures domain concepts and their relationships. Ontologies enable information sharing and reuse between different communicating entities. In relation to context-aware applications, ontologies are expressive enough to model different entities of the domain, their relationships and situations in human–machine readable form (Hoareau and Satoh, 2009; Baldauf et al., 2007). The commonly accepted language for modeling ontologies is OWL (Web Ontology Language). OWL is an extension of RDF (Resource Description Framework) with added expressivity and additional properties of relationships. There are three sublanguages of OWL: OWL Light, OWL DL, and OWL Full which differ in expressivity and computational completeness (Bechhofer et al., 2004). The OWL DL sublanguage, which is used in this work, is more expressive in comparison to OWL Light and is guaranteed to be computable during finite time period unlike OWL Full.

The knowledge captured in the ontology is expressed in concepts, relationships and individuals (instances of concepts). A concept represents entities of the domain. A set of entities and their relationships forms terminological component of the ontology or TBox. An example of TBox statement is “*All temperature sensors are environment sensors*”, which defines relationships between two concepts: temperature sensors and environment sensors. A set of facts associated with terminology forms assertion component or ABox. An example of ABox statement is “*Temperature sensor TS3 is located in room 1*”, which defines relationships between two instances of ontology: TS3 and room1.

In principle, any entity could be concept or fact in the ontology, and it is up to ontology engineer how to model knowledge of the domain. The modeling freedom has a two-sided effect. On one

[12] Ontology Web Language Description Logic: http://www.w3.org/TR/owl-semantics/syntax.html#2.3.2.

[13] Protégé ontology editor: http://protege.stanford.edu/.

[14] Pellet reasoner: http://clarkparsia.com/pellet/.

[15] OntoBroker Semantic Web middleware: http://www.w3.org/2001/sw/wiki/OntoBroker.

[16] HermiT reasoner: http://www.hermit-reasoner.com/

[17] KAON2 infrastructure for ontology management: http://kaon2.semanticweb.org/.

hand, it ensures the possibility to model any aspect of interest for different applications and users. On the other hand, it could lead to overcomplicated ontological models in the attempt to capture every possible detail of the domain. These models are hard to reuse, maintain, or merge with other ontologies for different applications (Simperl, 2009). In order to focus on the application scope and improve the readability of ontology by humans, it is important to carefully consider ontology taxonomies and relationships, which are both: (1) expressive enough to capture information relevant to the application in the knowledge model, and (2) simple and available for further extensions and reuse.

According to common engineering principles of ontologies, it is advisable to adopt existing ontological models from different applications in order to ensure their interoperability (Gomez-Perez et al., 2010; Ploennigst et al., 2012). However, the tools so far developed perform only a few of the specific tasks, such as translation between different languages (Simperl, 2009), calculation of semantic similarity between different concepts (Hadj Taieb et al., 2014) or semantic accuracy of ontologies (Sánchez et al., 2015), in an automatic manner. Some researchers address the problem of ontological reusability by creating extensive methodologies for ontological reuse (Fernández-López et al., 2013), others propose feasibility study methodologies in order to estimate how costly the ontological adoption is likely to be compared to developing ontologies from scratch (Simperl, 2009). As such, the process of ontological adaptation is still mostly manual and usually requires additional efforts for merging ontological semantics and class hierarchies.

The focus of this article is to prove a concept for the proposed architecture; and the adaptation of existing ontologies for purposes of information management middleware is a matter for further research. Thus, it was decided to develop ontology from scratch and use inductive ontology engineering methodology, which assumes development of ontology by generalizing specific use case(s) in the related domain (Holsapple and Joshi, 2002). The use case considered in this work is a rehabilitation facility, which is described in Section 6. The following subsections describe the terminological concepts of resulting ontology (TBox high level taxonomies and relationships).

4.1.1. Taxonomies

In order to be both simple and expressive, an ontology should reflect the necessary and sufficient information for the purposes of an application. Monitoring systems are characterized by numerous data sources and involvement of the various *users* and *user groups* with particular informational needs. Devices produce data, which should be analyzed; and if the data reports a relevant event, then an alarm or notification corresponding message should be delivered to a related user and/or a group of users. The group of devices located in the same place forms the context data of the place, thus the devices could be logically linked by *locations*.

In order to describe the environment, it was found necessary and sufficient to build five taxonomies on users, roles, devices, alarms, and places. The following is the description of TBox concepts:

- *Users* concept serves to describe general information about the users of the system. The individuals of this taxonomy contain information like names, age, contact details, etc.
- The *Roles* concept is separated from *Users*, since the same user could have several roles in the system (e.g. s/he could be assigned to both operator and managerial tasks). The division of roles and users facilitates the reconfiguration process if a user has to temporarily take over a new role while, for instance, substituting for a co-worker on sick leave. The taxonomy is domain-specific. E.g. in the case of smart home with assisted living it may contain *Inhabitants*, *Nurses*, *Maintenance* and *Managers* as the main classes (Evchina et al., 2013).
- *Devices* concept incorporates hierarchy of devices presented on the site.
- *Alarms* concept presents the hierarchy of notifications about on-going events in the system, which could be triggered by the devices.
- Finally, the *Places* concept reflects the hierarchy of the places presented in the domain. For example, in the manufacturing domain hierarchy of *Places* could contain *Buildings*, *Floors*, *ManufacturingLines*, *Cells*, *Robots*; for smart home domains these classes may be *Buildings*, *Floors*, *Apartments*, *Rooms*, *CommonPlaces* etc.

It is assumed that each taxonomy consists of the domain-specific classes. For example, the device taxonomy for a group of buildings with assisted living services might group sensors according to their purpose of measurement such as user state monitoring or environment conditions (Fig. 2). Within one domain ontologies remain the same. In this case the individuals of the ontologies and their inter-relationships reflect the differences of the various systems (e.g. the difference in ontology between systems of the same domain is only in ABox statements). Examples of taxonomies with individuals and their relationships are described in Section 6.

4.1.2. Relationships

Relationships in ontology reflect the connections between the concepts and their properties. The OWL expressivity allows them

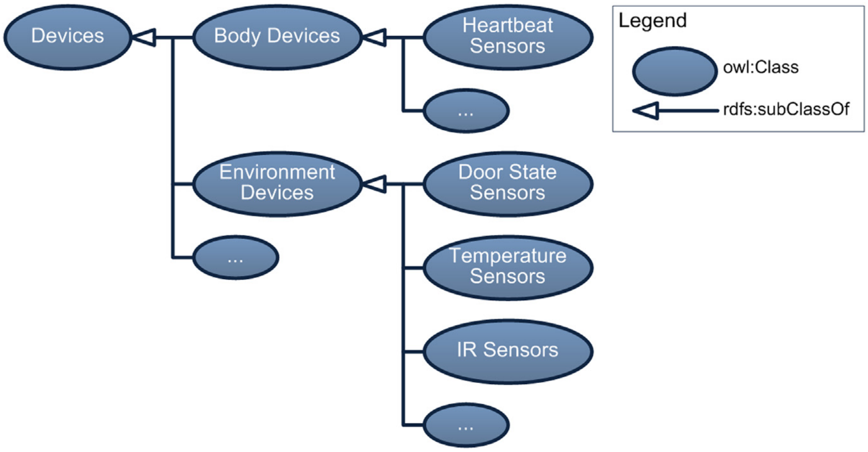


Fig. 2. Example of device taxonomy for houses with assisted living services.

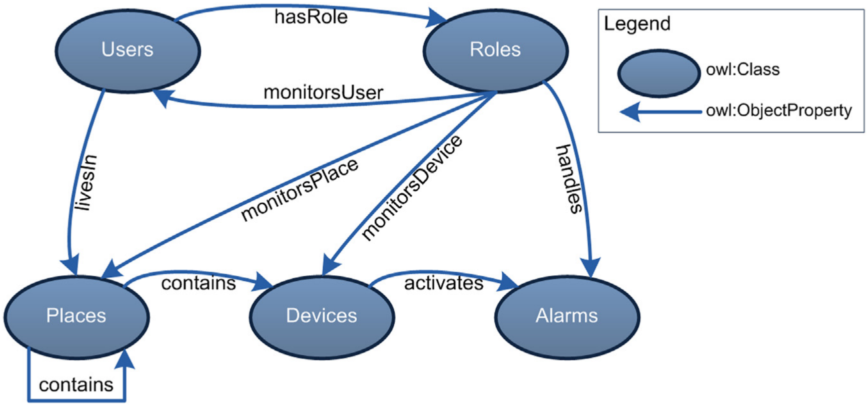


Fig. 3. Domain-independent high level ontology.

to be functional (*owl:FunctionalProperty*), transitive (*owl:TransitiveProperty*), inverse (*owl:InverseOf*) etc. thus capturing characteristics of real relationships between the objects of the system. Combination of high-level taxonomies with relationships gives domain-independent ontology, which is used in information management middleware (Fig. 3). The relationships connect the main taxonomies of the system and are applicable to all elements of underlying taxonomies adopted from different domains.

The *hasRole* relationship connects *Users* and *Roles* taxonomies; a user could be related to many roles and may thus have multiple *hasRole* relationships reflecting the whole range of his/her responsibilities in the system. The *Roles* taxonomy describes the responsibilities of the users via corresponding relationships with other entities of the system. The *monitorsUser* connects *Roles* with *Users* taxonomies in the case of health monitoring of residents in assisted living applications and stands for a link of the nursing role and an individual patient. The *monitorsPlace* relationship reflects that a user monitors all the devices in a particular location. The *monitorsDevice* relationship stands for the monitoring of a particular device or class of devices regardless of their location (e.g. an electrician may exclusively monitor the electricity supply meters). Finally, *handles* relationship reflects the responsibilities of certain users to handle the alarms in the system.

Places are connected with each other and with *Devices* via a *contains* relationship, which is transitive. Thus if it is stated that a device is contained in a particular room, the statement that the device is also located in the apartment with the room and in the building with the apartment both become explicit after the semantic reasoner has been applied to the knowledge model. Consequently, when some of the users are responsible for the buildings (via *monitorsPlace* relationship) the system will automatically “know” how many and what devices are located in those buildings. Currently the reasoning on transitive relationships is not supported in Linked Stream Data reasoners (Le-Phuoc et al., 2012b; Komazec et al., 2012); however this property is important in the proposed approach and is one of the reasons leading to the use of Semantic Web technologies.

The *activates* relationship dynamically links *Devices* with active *Alarms*. If there are no active alarms in the system, then individuals of these two taxonomies are not connected with each other. Conditions for the alarm activations are expressed via SWRL rules. The semantic reasoner analyzes data from devices causing *activates* relationships to appear in case of alarms. More examples of SWRL rules and alarm handling are described in Section 6.2.

4.2. Query templates

As mentioned above, proposed high-level taxonomies and the relationships between them are domain independent. The peculiarity of a domain is reflected via different kinds of taxonomies adopted on the lower levels. The defined high-level structure of the relationships and taxonomy makes it possible to use the same query templates for various domains. The queries return different results depending on the underlying featured models. This subsection explains what kind of query templates are used in IMM and what kind of information they return. Query templates are divided into two groups: information filtering and context analyzing (Fig. 1) with the description below.

4.2.1. Information filtering

Information filtering templates deal with selecting data for the user during normal system operation, i.e. they return information which should be presented to the user based on his/her role in the system and responsibilities. The responsibilities are defined via different *monitors* relationships (Fig. 3), which are used in query templates. The queries are run in two stages. The first stage occurs during the starting up of the middleware and is performed once; and as a result of this stage, the system retrieves the list of registered users:

```
SELECT ?user
WHERE {
  ?user rdf:type imm:Users.
}
```

Next, the model is queried for retrieving the list of devices with their values and locations which are under responsibility of each user. The example of the *monitorsPlace* relationship and environment devices is given below (the template values are *imm:userName* and *imm:hasTValue*; *imm:userName* is replaced with users according to the system list available from previous query, *imm:hasTValue* is replaced with the corresponding type of device values, i.e. double, integer, etc. supported in the system):

```
SELECT ?device ?value ?apartment
WHERE {
  imm:userName imm:hasRole ?role.
  ?role imm:monitorsPlace ?place.
  ?place imm:contains ?device.
  ?device rdf:type imm:EnvironmentDevices.
  ?device imm:hasTValue ?value.
  ?apartmentInd imm:contains ?device.
  ?apartmentInd rdf:type imm:Apartments.
  ?apartmentInd imm:hasName ?apartment.
}
ORDER BY ?apartment
```

The queries are run for each type of *monitors* relationship (i.e. *Places*, *Devices*, *Users*) and for each user. The result is a personalized list of monitored devices for each user depending on his/her role in the system and responsibilities.

4.2.2. *Context analyzing*

This type of query templates forms the basis for notification messages. The queries ask for active notifications in the system and for corresponding handlers of those notifications (i.e. responsible users of the system) to compose a message. Context analyzing query templates also work in two stages. In the first stage (middleware start up), the middleware asks for all types of notification messages, which are presented in the system. Then the model is asked periodically for active notifications and their handlers. The following is an example of a template for notification types caused by environment devices (*imm:alarmType* is replaced with the corresponding type of notifications available in the system, *imm:hasTValue* is replaced with the corresponding type of device values, i.e. double, integer, etc.):#

```
SELECT ?device ?value ?location ?handler
WHERE {
   ?device imm:activates imm:alarmType.
   ?device rdf:type imm:EnvironmentDevices.
   ?device imm:hasTValue?value.
   ?apartmentInd imm:contains ?device.
   ?apartmentInd rdf:type imm:Apartments.
   ?apartmentInd imm:hasName ?location.
   ?role imm:handles imm:alarmType.
   ?handler imm:hasRole ?role.
}
```

It is noteworthy that the context rules are domain specific and are expressed via SWRL. However, if the rules comply with the design principles described, i.e. when a condition is met, a rule defines the *activates* relationship between a device and corresponding alarm individuals, then it is possible to use a domain independent query template stated above. The next section describes the implementation prototype of the proposed architecture and message flows in the system.

5. Software tools for implementing IMM blocks functionality

The proposed functional blocks of IMM have modular structure with specific functionalities (Fig. 1), which could be quite easily divided between different tools implementing IMM. The distinct functionality and distributed nature of modern data intensive environments led to the adoption of the SOA approach for IMM implementation. The architecture layers are realized as separate web applications with the communication via Web Services (WS); these applications are *OntologyManager*, *OntologyService,* and *OntologyServiceClient* (Fig. 4). Each application launches its own embedded Jetty[18] web server and publishes its (WSDL[19]) description document in order to enable communication via WS. Applications use DPWS4J[20] stack, which provides an implementation of Devices Profile for Web Services (DPWS[21]) specification (version 1.1).

5.1. *OntologyManager*

In the current implementation *OntologyManager* is able to acquire data from other WS capable devices (and/or applications) and from MySQL databases using DPWS4J stack and Spring JdbcTemplate[22] classes respectively. The received data is mapped with the conditions determining whether the update rule will be triggered or not. If the condition is met, then a corresponding update statement (in the form of a SPARQL/Update expression) is sent to the *OntologyService* via invocation of *ExecuteUpdate*() WS. *OntologyManager* provides GUI for information sources configuration (e.g., subscription to WS, database connections) and modifications of SPARQL/Update expressions.

5.2. *OntologyService*

OntologyService provides access to the knowledge model of the system. It utilizes Apache Jena[23] framework for handling ontologies in the system and Pellet reasoner for inference. At present, *OntologyService* does not provide any GUI for end users. It is assumed, that an ontological model is created and edited using external tools such as Protégé, and the model could be later loaded into *OntologyService* using ontology URL. Access to the knowledge model is provided via two WSs: *ExecuteQuery*(), which has SPARQL query as a parameter and is used by *OntologyServiceClient* application; and *ExecuteUpdate*() with SPARQL/Update expression as a parameter. Other services are provided for convenient configuration, e.g., loading ontology, setting the reasoner (*SetReasoner*()).

5.3. *OntologyServiceClient*

OntologyServiceClient uses *ExecuteQuery*() service of *OntologyService* in order to request the knowledge model of the information according to query templates. In the current implementation *OntologyServiceClient* also provides simple HMI for displaying textual data from the knowledge model of the system. It is assumed that in later implementation *OntologyServiceClient* will provide data to other information consumers like adaptive HMIs (Nieto Lee et al., 2013) or other client applications via agreed interfaces.

5.4. *General message flow in the IMM*

The general information flow in the system during the runtime is illustrated in Fig. 5. The example is based on the data coming from the device via WS. *OntologyManager* receives notification of the changed device status (*statusChanged*), compares it internally with the conditions for update triggering; and, if the condition is met, sends a corresponding *ExecuteUpdate*() request with SPARQL/Update statement. *OntologyService* processes the update statement, performs reasoning over the updated ontological model, and sends *ontologyUpdated* notification to the subscribers. When *ontologyUpdated* notification is received by *OntologyServiceClient*, it initiates a loop to execute template queries in order to update information to the end user.

The implemented tools are model independent; they could operate the same way regardless of the underlying knowledge. Thus the same tools could be used in different domains if the knowledge model complies with the proposed design principles described in Section 4. The next section presents examples of

[18] Jetty web server and servlet container: http://www.eclipse.org/jetty/.

[19] Web Service Description Language (WSDL): http://www.w3.org/TR/wsdl.

[20] Devices Profile for Web Services for J2ME CDC: https://forge.soa4d.org/projects/dpws4j/.

[21] Devices Profile for Web Services (DPWS): http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01.

[22] Spring JdbcTemplate data access class: http://static.springsource.org/spring/docs/2.5.x/api/org/springframework/jdbc/core/JdbcTemplate.html.

[23] Apache Jena framework: http://jena.apache.org/.

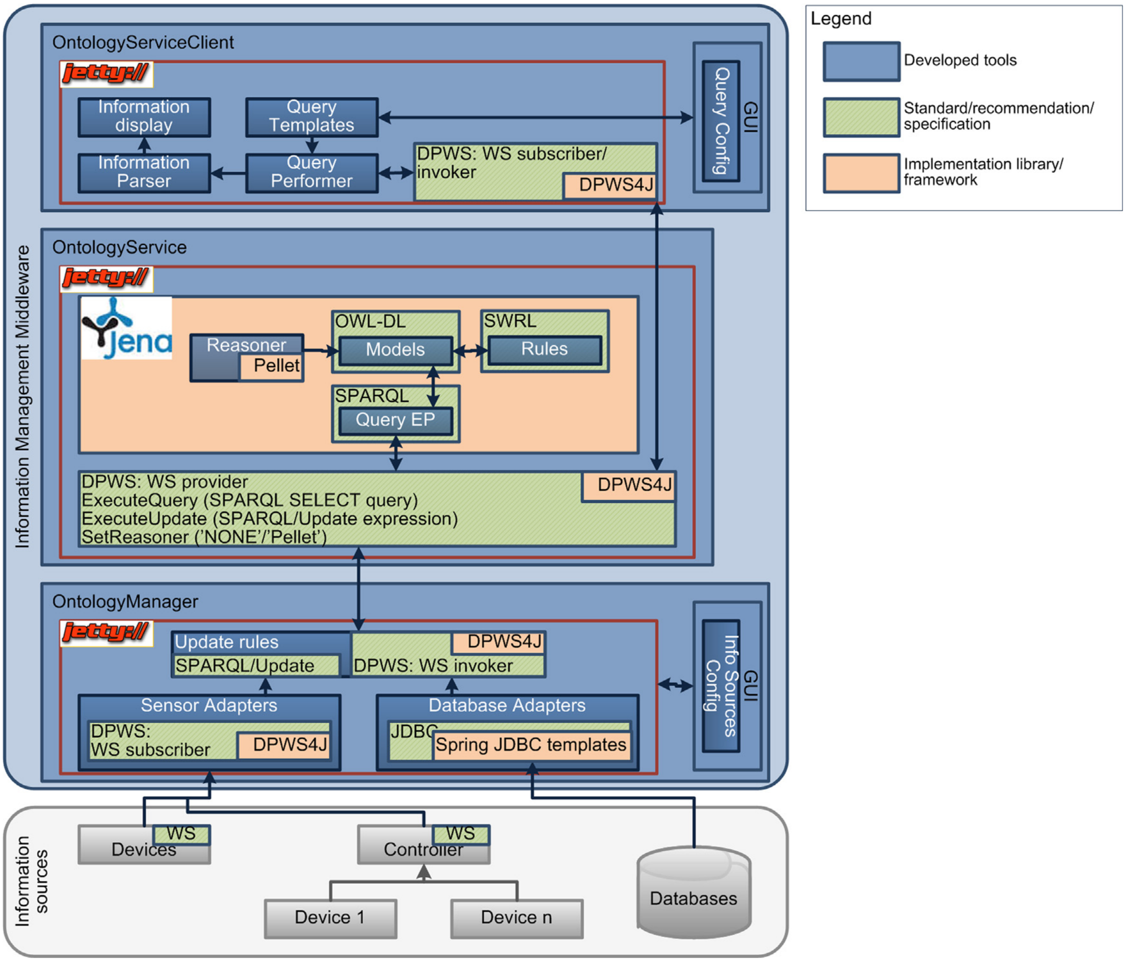


Fig. 4. Software tools implementing IMM.

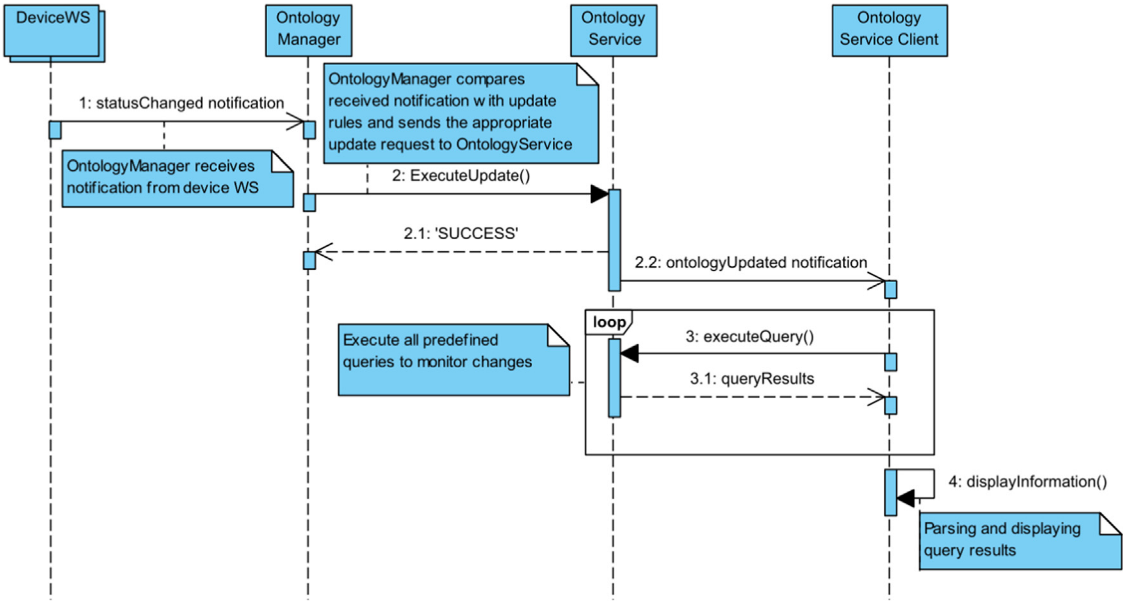

DeviceWS
Ontology Manager
Ontology Service
Ontology Service Client
1: statusChanged notification
OntologyManager receives notification from device WS
OntologyManager compares received notification w ith update rules and sends the appropriate update request to OntologyService
2: ExecuteUpdate()
2.1: 'SUCCESS'
2.2: ontologyUpdated notification
loop
3: executeQuery()
3.1: queryResults
Execute all predefined queries to monitor changes
4: displayInformation()
Parsing and displaying query results

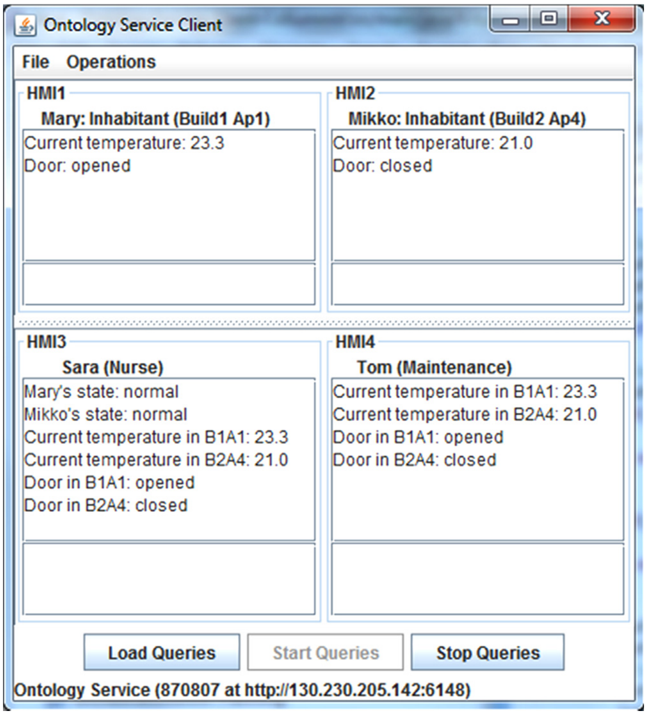
Fig. 5. Message flow in the system on receiving update notifications from device WS.

simple scenarios from the building automation domain demonstrating the reconfiguration and context-aware capabilities of the proposed architecture and exemplifying the knowledge model of the system.

6. Case study

The IMM was developed for data intensive monitoring systems and intended to deliver personalized information to the user,

Fig. 6. General view of the rehabilitation facility.



HMI1

Mary: Inhabitant (Build1 Ap1)

Current temperature: 23.3
Door: opened

HMI2

Mikko: Inhabitant (Build2 Ap4)

Current temperature: 21.0
Door: closed

HMI3

Sara (Nurse)

Mary's state: normal
Mikko's state: normal
Current temperature in B1A1: 23.3
Current temperature in B2A4: 21.0
Door in B1A1: opened
Door in B2A4: closed

HMI4

Tom (Maintenance)

Current temperature in B1A1: 23.3
Current temperature in B2A4: 21.0
Door in B1A1: opened
Door in B2A4: closed

Ontology Service (870807 at http://130.230.205.142:6148)

Fig. 7. Example of text-based HMI of the prototype.

analyze the current context in the environment, and facilitate the reconfiguration process of the system for technical administrators. The examples of those environments are Building and Factory Automation Systems (BAS and FAS correspondingly), which at present produce extensive data flows and are usually difficult to configure with new information such as devices or rules. In order to illustrate the reconfiguration and context-aware capabilities of the system, a sheltered accommodation unit with assisted living services was chosen for purposes of demonstration.

The rehabilitation facility with assisted living services is a group of three buildings having 10 apartments each and equipped with devices for monitoring environmental conditions and the state of its residents; it is situated in Tampere, Finland (Fig. 6). Each apartment has one room and bathroom. The site also provides social services to residents including health care monitoring and accommodation maintenance support. The three main user groups of the system are residents, nurses, and maintenance personnel. The users are interested in different kinds of information and have different access rights; for example, nurses and maintenance personnel monitor the status of the entire premises, while residents are able to monitor the information from the devices installed in their own apartments. The challenge in this system is to provide personalized information delivery to the users and monitor the current situation in the environment producing corresponding notification messages for different users. Moreover, the system is open for further extensions with new devices and the users of the system could also change. Thus it should be easily reconfigurable and open to new information sources and consumers.

The implemented prototype of the IMM has two demoBoxes as a testbed representing the scaled down physical architecture of the system. The demoBoxes are two cases equipped with Inico S1000 controllers of the real system, which enable information from devices as WS. Switches of demoBoxes simulate different sensors in the apartment. The following simulated sensors were taken into account during the testing: temperature sensor (real sensor mounted in the demoBox), IR sensor (it is assumed that two of these are installed in the room and bathroom respectively), door state sensor, nurse alarm button, and heartbeat sensor (simulated via information from database). The knowledge model used in the middleware reflects the site of the real system and is updated with information from demoBox switches simulating the current state in the environment. In the current prototype implementation the middleware displays information in textual form for four users of the system: two residents, one nurse and one maintenance person (Fig. 7). The display serves only for purposes of testing the middleware and visualizes information coming to the users of the system; the approach assumes middleware integration with advanced HMI builders (which are responsible for advanced visualization) and other information consumers. The following subsections present different scenarios of the system illustrating the reconfiguration and context-aware capabilities of IMM.

6.1. Reconfiguration scenarios

The proposed IMM utilizes ontologies as a knowledge model of the system. As mentioned earlier, ontological models are considered to be human–machine readable and there are several free ontology editors, which simplify the editing process even further, providing convenient GUI and thus easier model modifications. This subsection presents two examples demonstrating how ontological modeling could simplify the reconfiguration process of information flows in data intensive environments.

6.1.1. Adding new devices

The present monitoring systems in the buildings are large scaled and could be extended with new devices. Adding new devices requires not only physical installation, but also information flow reconfiguration such as assigning meta-descriptions to devices (location, and device type), access rights (who has rights to monitor the data from the device), context rules (what are the conditions to trigger notifications and who should receive them). The example below is focused on meta-data and access rights configuration (context-awareness and rules examples are presented in Section 6.2).

In Fig. 8 illustrates part of the ontological model used in the prototype scenario. Ontological classes are represented by rectangles, while individuals of ontology are diamonds. The model follows the design principles proposed and has corresponding taxonomies (*Users*, *Roles*, *Places* and *Devices* are presented in Fig. 8). The *Places* taxonomy reflects the general structure of building automation domains with indoor and outdoor places, buildings, apartments, and rooms. The specific location structure is captured via individuals and their relationships (e.g., it is possible to conclude from the picture that building 2 contains apartments 4 and 5, and apartment 5 in turn has room 10 for the accommodation unit presented).

Decoupling of data from the application logic is done with query templates, which keep the same structure and return different results depending on the underlying model. In the example presented, when a new device is added to the system (temperature sensor 3 in room 10), only a few things should be changed in the knowledge model (presented in bold face in Fig. 8): (1) adding new individual of type *TemperatureSensors*; (2) adding *contains* relationship between this individual and the corresponding room. The following query asks the list of devices and their values over the model (the values are literals and are not presented in Fig. 8):

```
SELECT ?device ?value ?apartment
WHERE {
    imm:userName imm:hasRole ?role.
    ?role imm:monitorsPlace ?place.
    ?place imm:contains ?device.
    ?device rdf:type imm:EnvironmentDevices.
    ?device imm:hasTValue ?value.
    ?apartmentInd imm:contains ?device.
    ?apartmentInd rdf:type imm:Apartments.
    ?apartmentInd imm:hasName ?apartment.
}
ORDER BY ?apartment
```

Applied to the user *Tom*, who is the maintenance person on the site responsible for monitoring buildings 1 and 2 (with the following changes in the query: *imm:userName -> imm:Tom*, *imm:hasTValue -> imm:hasDValue*), the query will automatically return a new temperature sensor value for apartment 5, so the parser will be able to display it correctly on the text HMI (Fig. 9). Similarly the query results will be updated for user *Sara*, who is the nurse on the site and is also responsible for monitoring building 2 and its residents. The changes in the knowledge model will not affect the residents' HMIs since their access rights are restricted to their own apartments, which are different from apartment 5 in this example. To summarize, adding new device to the system necessitates changes in the ontological model and its update rules. Such changes are quite easy to hide under convenient user interface facilitating reconfiguration operations.

6.1.2. Access rights management

The distributed nature of the present monitoring systems and the availability of different types of data make their use attractive to various types of users interested in various kinds of information. Access rights management becomes an important topic in the reconfiguration process of data intensive environments. The example below shows a simple approach to access rights management using Semantic Web technologies.

The proposed ontological design principles include *Roles* taxonomy, the purpose of which is to separate users and their responsibilities on the site. With this approach it is possible to create specific roles in the domain and assign these roles to one or more users. For example, the part of the ontology presented in Fig. 10 shows that two maintenance personnel on the site (*Tom* and *John*) have two separate maintenance roles with responsibilities for buildings 1 and 2 respectively.

The distinct responsibilities of different users enable personalized information delivery and a reduction of information load. E.g. in the example presented, maintenance personnel will receive only information from devices of corresponding buildings when a

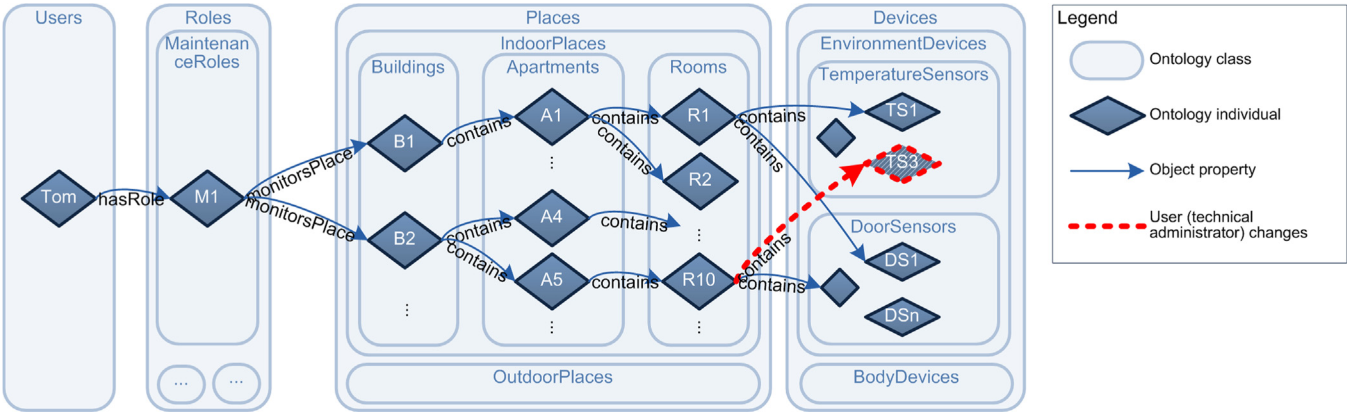
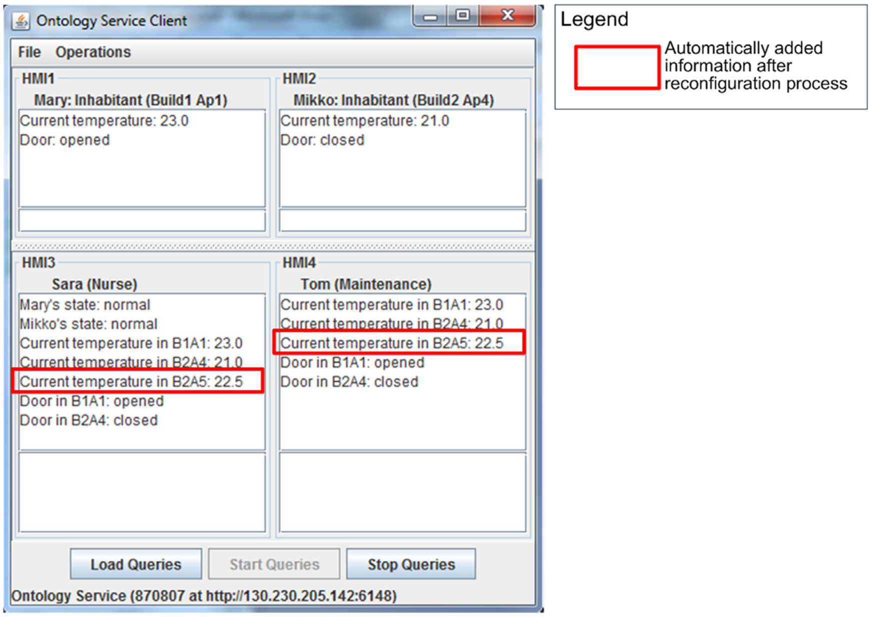


Fig. 8. Ontological reconfiguration for adding new device.



HMI1
Mary: Inhabitant (Build1 Ap1)
Current temperature: 23.0
Door: opened

HMI2
Mikko: Inhabitant (Build2 Ap4)
Current temperature: 21.0
Door: closed

HMI3
Sara (Nurse)
Mary's state: normal
Mikko's state: normal
Current temperature in B1A1: 23.0
Current temperature in B2A4: 21.0
Current temperature in B2A5: 22.5
Door in B1A1: opened
Door in B2A4: closed

HMI4
Tom (Maintenance)
Current temperature in B1A1: 23.0
Current temperature in B2A4: 21.0
Current temperature in B2A5: 22.5
Door in B1A1: opened
Door in B2A4: closed

Ontology Service (870807 at http://130.230.205.142:6148)

Fig. 9. Reconfiguration: adding new device.

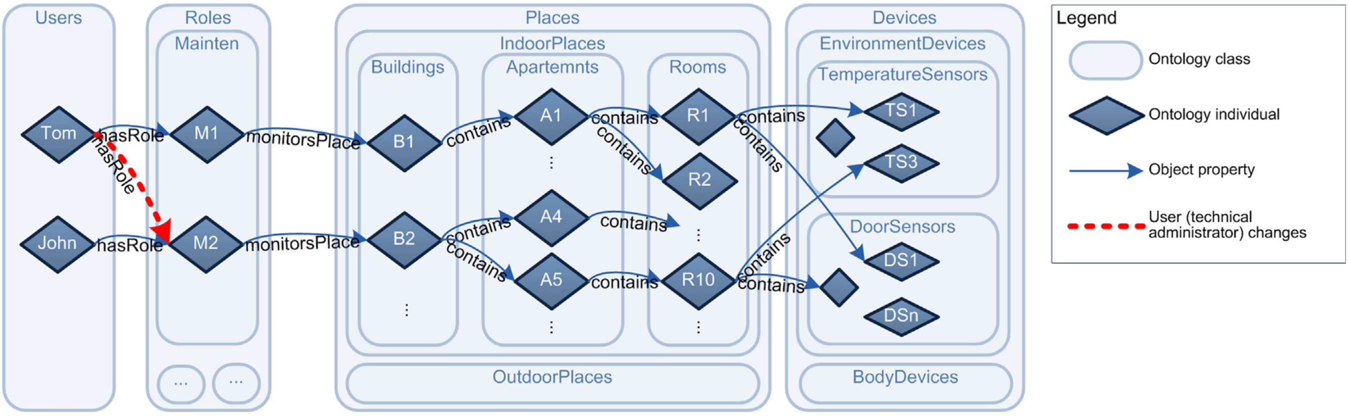


Fig. 10. Ontological reconfiguration for access rights management.

query from Section 6.1.1 is applied to the knowledge model. Distinguishing users and roles enables easier reconfiguration of access rights. For example, in case of John's sudden illness, it is possible simply to add a new *hasRole* relationship between *Tom* and *John's* role (*M2*) to temporarily expand his responsibilities to cover building 2 as well as building 1 (shown in bold face in Fig. 10). Responsibilities of roles could be assigned to places, separate device classes and users (e.g. in case of nurses monitoring residents' condition) creating a flexible way of managing access to information flows in the system.

6.2. *Context-awareness scenarios*

In order to personalize information delivery depending on the current situation in the domain, the system should be context-aware. This section presents examples of context-aware information delivery based on the monitored data from devices. The examples demonstrate the application of SWRL rules and comparison built-ins, which enable simple temporal reasoning over device data.

6.2.1. *Extension with SWRL rules*

Context-awareness is characterized by the ability to react to the current situation in the environment. From this perspective ontologies have an extension in the form of SWRL rules enabling the expression of custom logic and conditions beyond OWL expressivity. The rules are domain and application dependent and should be configured for each system according to its requirements. However, the system logic changes quite seldom, and once the rules are configured they are applicable to a whole class of devices automatically. To illustrate possible SWRL rules usage, scenarios with fire alarm activation based on temperature sensors data, and nurse alarm activation based on wearable heartbeat sensors are considered. The corresponding alarms are triggered if the values of the sensors exceed certain thresholds.

According to proposed ontological design principles different alarm types specific to the system are located under *Alarms* taxonomy. Active alarms are indicated via *activates* relationship, appearing between alarm type and the sensor causing the alarm. Fig. 11 illustrates two active fire and nurse alarms. The appearance of *activates* relationships is dictated via a dedicated SWRL rule in which the data values of the sensors are compared with the threshold values for alarm activation. For example, the rule activating a fire alarm in the system looks like follows:

$$\text{TemperatureSensors(?sensor)} \wedge \text{hasDValue(?sensor ,?value)} \\ \wedge \text{swrlb:greaterThan(?value, 40)} \rightarrow \text{activates(?sensor, FA1)}$$

When a semantic reasoner (in this case Pellet) is applied to the ontology depicted in Fig. 11, which contains the SWRL rule presented above, the *activates* relationship appears between the corresponding temperature sensor and the fire alarm individual *FA1* (bold lines in Fig. 11).

The individuals of the *Roles* taxonomy indicate responsibilities for alarm handling via *handles* relationship. The following query returns the list of devices, values, locations, and handlers on *FA1* alarm type:

```
SELECT ?device ?value ?location ?handler
WHERE {
   ?device imm:activates imm:FA1.
   ?device rdf:type imm:EnvironmentDevices.
   ?device imm:hasDValue ?value.
   ?apartmentInd imm:contains ?device.
   ?apartmentInd rdf:type imm:Apartments.
   ?apartmentInd imm:hasName ?location.
   ?role imm:handles imm:FA1.
   ?handler imm:hasRole?role.
}
```

In this example, the result of the query will be:

device	Value	Location	Handler
TS1	45.0	A1	Tom

Similarly there is an SWRL rule specifying nurse alarm, which is triggered if the heartbeat sensors of users exceed 120

HeartRateSensors(?sensor) ∧ hasDValue(?sensor,?value) ∧ swrlb:greaterThan(?value, 120)→activates(?sensor, NA1)

The reasoner will infer *activates* relationships between Mary's heartbeat sensor and the nurse alarm individual (dashed line in Fig. 11). The query asking about active alarms for *NA1* type of the wearable sensors is#

```
SELECT ?device ?value ?user ?handler
WHERE {
   ?device imm:activates imm:NA1.
   ?device rdf:type imm:BodyDevices.
   ?device imm:hasDValue ?value.
   ?user imm:wears ?device.
   ?role imm:handles imm:NA1.
   ?handler imm:hasRole ?role.
}
```

The returned result is:

Device	Value	User	Handler
HR1	132.0	Mary	Sara

Corresponding parsers split the data received and notification messages will be shown on the screen (Fig. 12).

In case of simultaneous alarms of one type caused by different devices (e.g., several fire alarms from different temperature sensors), the query result set will have as many rows as the number of sensors having values beyond the threshold. The parser will split the result values and deliver corresponding messages for display. If there are no active alarms in the system, then queries

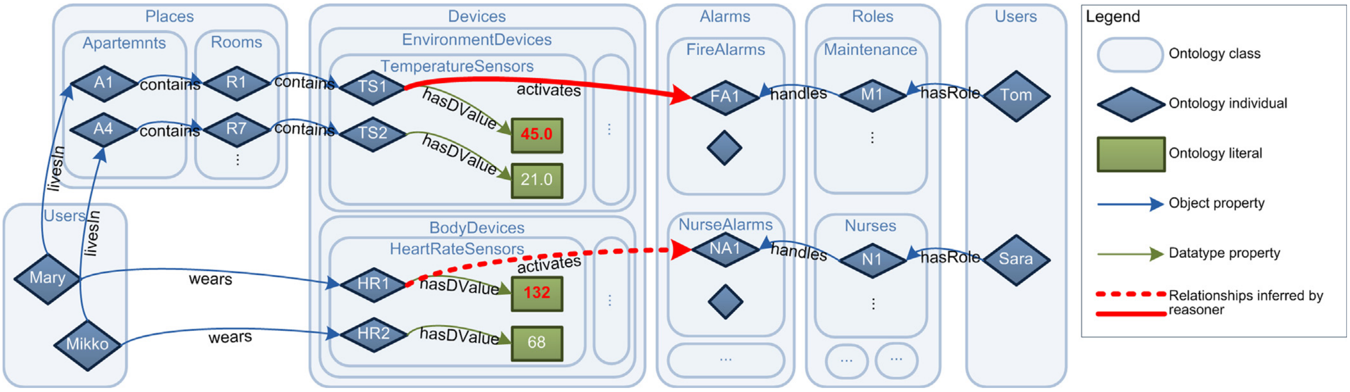
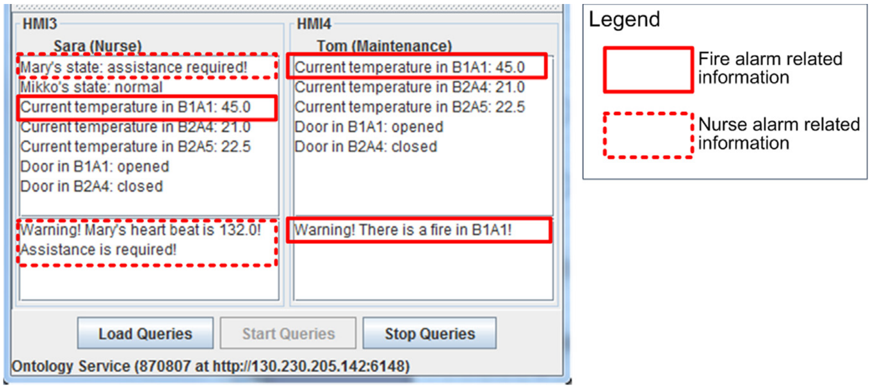


Fig. 11. Changes in the ontology based on current context.



```
Mary's state: assistance required!
Mikko's state: normal
Current temperature in B1A1: 45.0
Current temperature in B2A4: 21.0
Current temperature in B2A5: 22.5
Door in B1A1: opened
Door in B2A4: closed

Warning! Mary's heart beat is 132.0!
Assistance is required!
```

```
Current temperature in B1A1: 45.0
Current temperature in B2A4: 21.0
Current temperature in B2A5: 22.5
Door in B1A1: opened
Door in B2A4: closed

Warning! There is a fire in B1A1!
```

Ontology Service (870807 at http://130.230.205.142:6148)

Fire alarm related information

Nurse alarm related information

Fig. 12. Context-aware behavior of the system.

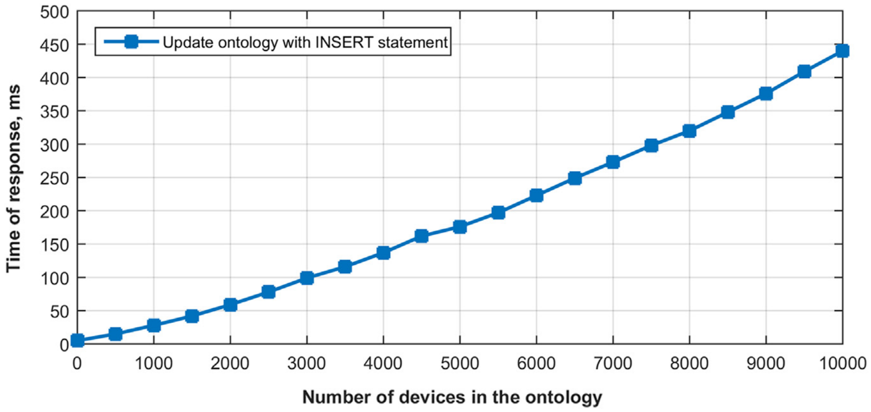


Fig. 13. Updating ontology with SPARQL/Update statements.

return empty results and no additional information will be shown on the screen. Thus, applying SWRL rule extensions to ontological models updated with values from sensors enables reasoning over the current situation in the system and personalized notification messages based on the user roles and responsibilities.

6.2.2. *Temporal relationships reasoning (reasoning with temporal relationships)*

The main challenge of using Semantic Web approaches in context-aware systems is a lack of adequate temporal reasoning support between data sources. In some cases this challenge is addressed by introducing new temporal ontological concepts, which allows description of temporal properties of the data (Hobbs and Pan, 2004), however these approaches still do not provide means for analyzing temporal data during run-time. In this work, the problem of time relationships between source data is addressed by using SWRL built-ins for comparison. These are able to reason over XML schema dateTime datatype.[24] Using built-ins with dateTime datatype properties allows simple comparison between temporal entities, which is reasonable for most cases. One of the examples is reasoning over IR sensor data. The IR sensor is installed in every room of the apartments and signals the presence of residents. In combination with a door sensor it is possible to infer whether the apartment is empty during the fire alarm (if IR activation happened only after door sensor activation) and trigger notification for assistance during the fire alarm (*ADF1*) to the nurse. In the SWRL format the rule looks as follows:

activates(FA1, ?device) ∧ contains(?apartment, ?device) ∧ Apartments(?apartment) ∧ contains(?apartment, ?IRsensor) ∧ IRSensors(?IRsensor) ∧ wasActivatedAt(?IRsensor, ?IRtime) ∧ contains(?apartment, ?doorSensor) ∧ DoorSensors(?doorSensor) ∧ wasActivatedAt(?doorSensor, ?doorTime) ∧ swrlb: greaterThan(?IRtime, ?doorTime)→activates(?IRsensor, ADF1)

The rule above checks active fire alarms in the system and compares the time between the last triggering of IR and door sensors in the corresponding apartment in order to verify whether the apartment is empty or to trigger the assistance alarm.

In summary, this section demonstrated examples of the context-aware and reconfiguration capabilities of the proposed middleware and the principles of decoupling data and application logic by modeling the context of the system with ontologies. This decoupling enables the same tools to be used for different domains and allows logical changes in the system (e.g., new users, devices, system rules, and access rights management) only inside ontological models with no need to recompile supported tools.

7. Evaluation

The crucial block of the proposed middleware is ontological context model with SWRL rules. Thus, the evaluation of middleware is focused on performance check of two key point processes in the proposed solution: (1) updating ontology with new information via SPARQL/Update statements; (2) querying ontology for specific information with SPARQL expressions. Both processes also involve semantic reasoning over ontological model with Pellet reasoner, thus considering SWRL rules incorporated into model.

Experimental ontology models four buildings each having ten apartments with various devices. Total number of devices in the ontology varied from 0 to 10000 during the tests. Time of responses was measured starting from sending request till receiving response from *OntologyService*, thus including following steps: sending request via WS, processing SPARQL/Update or SPARQL request with Pellet reasoner, receiving response via WS. The tests were running on one machine having two CPUs with 1.9 GHz each. *OntologyService* was allocated with 1GB of RAM.

7.1. *Updating ontology*

This experiment shows how fast ontology is updated with new information depending on number of individuals, which already are in the ontology. Each update statement contained insertion of new device, values and device location into ontology (in total six RDF triples in each insert request). The resulting response time is shown in Fig. 13. The growing delay in the ontology update response is explained by application of Pellet reasoner, which requires more time on processing inserted facts into model with large number of existing facts. Although response time is steadily growing, for some of the monitoring systems such as building or home automation, the half-second delay might be not critical.

7.2. *Querying ontology*

This experiment shows response time on querying ontology with SPARQL statements. Each statement was queried for 20 times and average response was put in Fig. 14. The first tested query (blue line in Fig. 14) has asked all devices, their values and locations from Building1. The inserted devices were distributed equally between four buildings, thus the returned number of devices with their information were steadily growing from 0 to 2500 during experiment. The second tested query (red line in

[24] XML schema dataTime datatype: http://www.w3.org/TR/xmlschema-2/#dateTime.

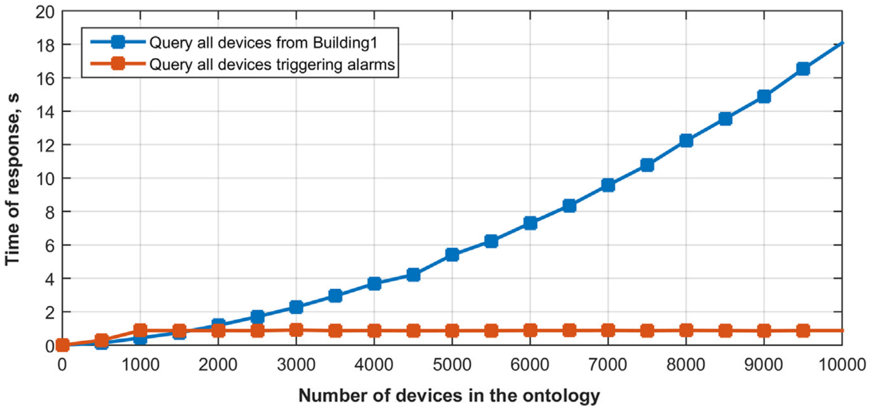


Fig. 14. Querying ontology with SPARQL statements: (a) blue line: number of returned devices was steadily growing from 0 to 2500; (b) orange line: number of returned devices was growing from 0 to 500 (for the first three tests) and then remained 500 till the end (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

Fig. 14) has asked all devices, which currently trigger alarms (i.e. the results of SWRL rule inferencing). The total number of devices with alarms was 500 and these devices were inserted into ontology among first 1000. Thus, starting from 1000 devices in the ontology, the number of devices triggering alarms was the same and each query returned 500 devices.

Results presented in Fig. 14 suggest that response time of SPARQL queries mostly depends on number of returned result sets rather than size of ontology. Thus, the query with constant number of returned devices (red line in Fig. 14) has required around 800 ms of response time irrelevant to ontology size, while query with increasing number of returned devices has considerable increase in response time. In that way, for better performance results it is preferably to estimate number of returned results. For example, instead of querying all type of devices in the ontology for specific user it is better to perform series of queries with different device types in each query.

8. Conclusions

This paper addresses the problem of personalized information delivery with respect to the user's role in the system, his needs and responsibilities, in environments, where user has to follow the information from numerous data sources and situations. The solution is seen in making the system aware of its context (e.g. the needs of a particular user, his role in the system, the layout of the locations with devices and users) with support middleware, which is able to reduce information flow to the user by delivering only information relevant in the current situation. The paper analyzes the main trends in context-aware systems and justifies the chosen approach with Semantic Web technologies.

The solution proposed is information management middleware, which is capable of decoupling data and application logic by (1) modeling the context of the system via ontologies with runtime data updating; (2) using domain-independent tools to support the knowledge model of the system and deliver personalized user information. The paper describes the functional blocks of the middleware, the design principles of ontologies and queries, and the tools implemented. The proposed middleware has been evaluated with performance tests suggesting feasibility of the approach for monitoring systems.

The advantages of the middleware are seen in the strong reconfiguration capabilities (e.g. access rights management, adding new devices and system rules) in such a way that (1) the system could be easily reconfigured with the changes made only to the knowledge model, (2) the changes do not alter the recompilation of software tools comprising middleware architecture. Reconfiguration and context-aware capabilities are demonstrated with examples from building automation domain.

A limitation of the work is seen in the lack of sophisticated temporal reasoning support over available data. At present this problem is partially solved with SWRL comparison built-ins, however further investigation of this issue is required.

Future work will be developed in several directions: (1) while the key modeling principles were proposed in the paper, the deeper investigation of more use cases is needed for identification of missing concepts (e.g. Assets or user Tasks) needed by more general use cases; (2) reconfiguration of the system with human-readable ontologies could still be difficult for technical administrators without prior knowledge of ontologies, thus GUI development of reconfiguration tools is the next natural step; (3) the full potential of context-aware information management might be achieved with integration of the middleware with adaptive HMIs conveniently representing information to users.

Acknowledgments

This work was partially supported by the ARTEMIS First Call 2010 Program under Grant Agreement number 269334, corresponding to the project ASTUTE (Pro-active decision support for data-intensive environments) and the Doctoral Programme of the President of the Tampere University of Technology (TUT).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.engappai.2015.04.008.

References

Abowd, G., Dey, A., 1999. Towards a better understanding of context and context-awareness. Lect. Notes Comput. Sci. 1707, 304–307.
Ailisto, H., Alahuhta, P., Haataja, V., Kyllönen, V., Lindholm, M., 2002. Structuring context aware applications: five-layer model and example case. In: Proceedings of the Workshop on Concepts and Models for Ubiquitous Computing. pp. 1–5.
Atzori, L., Iera, A., Morabito, G., 2010. The internet of things: a survey. Comput. Netw. 54, 2787–2805.
Baldauf, M., Dustdar, S., Rosenberg, F., 2007. A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2, 263–277.
Barros, H., Silva, A., Costa, E., Bittencourt, I.I., Holanda, O., Sales, L., 2011. Steps, techniques, and technologies for the development of intelligent applications based on Semantic Web Services: a case study in e-learning systems. Eng. Appl. Artif. Intell. 24, 1355–1367.

Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., Schwinger, W., 2010. BeAware! – situation awareness, the ontology-driven way. Data Knowl. Eng. 69, 1181–1193.

Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, J., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2004. OWL Web ontology language [WWW Document]. URL ⟨http://www.w3.org/TR/owl-ref/⟩.

Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A., Tanca, L., 2007. A data-oriented survey of context models. ACM SIGMOD Rec. 36, 19–26.

Chen, L., Shadbolt, N., Goble, C., 2007. A Semantic Web-based approach to knowledge management for grid applications. IEEE Trans. Knowl. Data Eng. 19, 283–296.

Chernbumroong, S., Cang, S., Atkins, A., Yu, H., 2013. Elderly activities recognition and classification for applications in assisted living. Expert Syst. Appl. 40, 1662–1674.

Chung, D., Ku, M., Park, Y., Min, D., 2013. Performance evaluation for efficient complex event processing system according to event types and environment. Ubiquitous Inf. Technol. Appl. (LNEE) 214, 529–536.

Dey, A.K., 2001. Understanding and using context. Pers. Ubiquitous Comput. 5, 4–7.

Eckert, M., Bry, F., Brodt, S., Poppe, O., Hausmann, S., 2011. A CEP babelfish: languages for complex event processing and querying surveyed. Reason. Event-Based Distrib. Syst. (SCI) 347, 47–70.

Emmanouilidis, C., Koutsiamanis, R.-A., Tasidou, A., 2013. Mobile guides: taxonomy of architectures, context awareness, technologies and applications. J. Netw. Comput. Appl. 36, 103–125.

Esposito, A., Tarricone, L., Zappatore, M., Catarinucci, L., Colella, R., Dibari, A., 2008. A framework for context-aware home-health monitoring. Ubiquitous Intell. Comput. 3, 75–91.

Evchina, Y., Dvoryanchikova, A., Martinez Lastra, J.L., 2013. Semantic information management for user and context aware smart home with social services. In: Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support. pp. 262–268.

Feng, Y.-H., Teng, T.-H., Tan, A.-H., 2009. Modelling situation awareness for context-aware decision support. Expert Syst. Appl. 36, 455–463.

Fernández-López, M., Gómez-Pérez, A., Suárez-Figueroa, M.C., 2013. Methodological guidelines for reusing general ontologies. Data Knowl. Eng. 86, 242–275.

Fischer, G., 2012. Context-aware systems – the 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person. In: Proceedings of the Conference on Advanced Visual Interfaces. pp. 287–294.

Furno, A., Zimeo, E., 2014. Context-aware composition of Semantic Web Services. Mob. Netw. Appl. 19, 235–248.

Gao, F., Bhiri, S., 2012. User-centric complex event modeling and implementation based on ubiquitous data service. In: Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. pp. 67–70.

Gellrich, A., Lunkwitz, D., Dennert, A., Kabitzsch, K., 2012. Rule-driven manufacturing control based on ontologies. In: Proceedings of the Conference on IEEE Emerging Technologies and Factory Automation. pp. 1–8.

Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O., 2010. Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, London.

Hadj Taieb, M.A., Ben Aouicha, M., Ben Hamadou, A., 2014. Ontology-based approach for measuring semantic similarity. Eng. Appl. Artif. Intell. 36, 238–261.

Hasan, S., Curry, E., Banduk, M., O'Riain, S., 2011. Toward situation awareness for the semantic sensor web: complex event processing with dynamic linked data enrichment. In: Proceedings of the International Workshop on Semantic Sensor Networks. pp. 60–72.

Her, Y., Kim, S.-K., Jin, Y.T., 2010. A context-aware framework using ontology for smart phone platform. Int. J. Digit. Content Technol. Appl. 4, 159–167.

Hoareau, C., Satoh, I., 2009. Modeling and processing information for context-aware computing: a survey. New Gener. Comput. 27, 177–196.

Hobbs, J.R., Pan, F., 2004. An ontology of time for the semantic web. ACM Trans. Asian Lang. Inf. Process. – Spec. Issue Temporal Inf. Process. 3, 66–85.

Holsapple, C.W., Joshi, K.D., 2002. A collaborative approach to ontology design. Commun. ACM 45, 42–47.

Hong, J., Suh, E.-H., Kim, J., Kim, S., 2009. Context-aware system for proactive personalized service based on context history. Expert Syst. Appl. 36, 7448–7457.

Khilwani, N., Harding, J.A., Choudhary, A.K., 2009. Semantic web in manufacturing. Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf. 223, 905–924.

Komazec, S., Cerri, D., 2011. Towards efficient schema-enhanced pattern matching over RDF data streams. In: Proceedings of the International Workshop on Ordering and Reasoning.

Komazec, S., Cerri, D., Fensel, D., 2012. Sparkwave: continuous schema-enhanced pattern matching over RDF data streams. In: Proceddings of the International Conference on Distributed Event-Based Systems. pp. 58–68.

Krieger, H.-U., 2010. A general methodology for equipping ontologies with time. In: Proceedings of the International Conference on Language Resources and Evaluation. pp. 3165–3172.

Le-Phuoc, D., Nguyen-Mau, H.Q., Parreira, J.X., Hauswirth, M., 2012a. A middleware framework for scalable management of linked streams. Web Semant.: Sci. Serv. Agents World Wide Web 16, 42–51.

Le-Phuoc, D., Parreira, J.X., Hauswirth, M., 2012b. Linked stream data processing. Reason. Web Semant. Technol. Adv. Query Answering (LNCS) 7487, 245–289.

Martinez Lastra, J.L., Delamer, I.M., 2006. Semantic Web Services in factory automation: fundamental insights and research roadmap. IEEE Trans. Ind. Inform. 2, 1–11.

Miguelanez, E., Patron, P., Brown, K.E., Perillot, Y.R., Lane, D.M., 2011. Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. IEEE Trans. Knowl. Data Eng. 23, 759–773.

Moreno, A., Valls, A., Isern, D., Marin, L., Borràs, J., 2013. SigTur/E-Destination: ontology-based personalized recommendation of tourism and leisure activities. Eng. Appl. Artif. Intell. 26, 633–651.

Nguyen, T. Van, Lim, W., Nguyen, H.A., Choi, D., Lee, C., 2007. Context ontology implementation for smart home. In: Proceedings of the 2nd International Conference on Ubiquitous Information Technologies and Applications.

Nieto Lee, A.N., Evchina, Y., Dvoryanchikova, A., Martinez Lastra, J.L., 2013. Proactive content managing system for efficient human machine interaction in data intensive environments. In: Proceedings of the IEEE 11th International Conference on Industrial Informatics. pp. 790–796.

O'Connor, M.J., Das, A.K., 2011. A method for representing and querying temporal information in OWL. Biomed. Eng. Syst. Technol. (CCIS) 127, 97–110.

Ottenwälder, B., Koldehofe, B., Rothermel, K., Hong, K., Lillethun, D., Ramachandran, U., 2014. MCEP: a mobility-aware complex event processing system. ACM Trans Internet Technol. – Spec. Issue Event Recognit. 14 (6), 1–6 24.

Ploennigst, J., Hensel, B., Dibowski, H., Kabitzsch, K., 2012. BASont – a modular, adaptive building automation system ontology. In: Proceedings of the IEEE Industrial Electronics Society. pp. 4827–4833.

Puttonen, J., Lobov, A., Soto Cavia, M.A., Martinez Lastra, J.L., 2010. A Semantic Web Services-based approach for production systems control. Adv. Eng. Inform. 24, 285–299.

Rodriguez, J., Bravo, M., Guzman, R., 2012. Multi-dimensional ontology model to support context-aware systems. In: Proceedings of the International Conference on Internet and Web Applications and Services. pp. 80–89.

Sánchez, D., Batet, M., Martínez, S., Domingo-Ferrer, J., 2015. Semantic variance: an intuitive measure for ontology accuracy evaluation. Eng. Appl. Artif. Intell. 39, 89–99.

Santofimia, M.J., Fahlman, S.E., del Toro, X., Moya, F., Lopez, J.C., 2011. A semantic model for actions and events in ambient intelligence. Eng. Appl. Artif. Intell. 24, 1432–1445.

Sauter, T., Soucek, S., Kastner, W., Dietrich, D., 2011. The evolution of factory and building automation. IEEE Ind. Electron. Mag. 5, 35–48.

Schmidt, A., 2014. Context-aware computing: context-awareness, context-aware user interfaces, and implicit interaction. In: Soegaard, M., Dam, R.F. (Eds.), The Encyclopedia of Human-Computer Interaction. The Interaction Design Foundation.

Sequeda, J.F., Corcho, O., 2009. Linked stream data: a position paper. In: Proceedings of the International Workshop on Semantic Sensor Networks.

Simperl, E., 2009. Reusing ontologies on the Semantic Web: a feasibility study. Data Knowl. Eng. 68, 905–925.

Taylor, K., Leidinger, L., 2011. Ontology-driven Complex Event Processing in heterogeneous sensor networks. Semant. Web: Res. Appl. (LNCS) 6644, 285–299.

Terroso-Sáenz, F., Valdés-Vela, M., Campuzano, F., Botia, J.A., Skarmeta-Gómez, A.F., 2015. A complex event processing approach to perceive the vehicular context. Inf. Fusion 21, 187–209.

Teymourian, K., Paschke, A., 2009. Semantic rule-based complex event processing. Rule Interchange Appl. (LNCS) 5858, 82–92.

Uddin, M.K., Puttonen, J., Scholze, S., Dvoryanchikova, A., Lastra, J.L.M., 2012. Ontology-based context-sensitive computing for FMS optimization. Assem. Autom. 32, 163–174.

Viktoratos, I., Tsadiras, A., Bassiliades, N., 2014. Providing a context-aware location based web service through semantics and user-defined rules. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM Press, New York, USA, pp. 9:1–9:11.

Weiss, J., Mandl, P., Schill, A., 2011. Functional testing of complex event processing applications. In: Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing. pp. 407–413.

Yao, W., Chu, C.-H., Li, Z., 2011. Leveraging complex event processing for smart hospitals using RFID. J. Netw. Comput. Appl. 34, 799–810.

Zappia, I., Paganelli, F., Parlanti, D., 2012. A lightweight and extensible complex event processing system for sense and respond applications. Expert Syst. Appl. 39, 10408–10419.

Publication IV

Evchina, Y., & Martinez Lastra, J. L. (2016). Hybrid approach for selective delivery of information streams in data-intensive monitoring systems. *Advanced Engineering Informatics*, *30*(3), 537–552. https://doi.org/10.1016/j.aei.2016.07.006

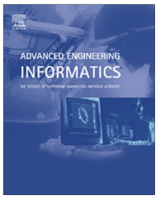
The original print of this publication is available online (DOI: 10.1016/j.aei.2016.07.006)

Advanced Engineering Informatics 30 (2016) 537–552

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Hybrid approach for selective delivery of information streams in data-intensive monitoring systems

Yulia Evchina [*], Jose Luis Martinez Lastra

FAST Laboratory, Tampere University of Technology, P.O. Box 600, FIN-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 4 February 2016
Received in revised form 21 June 2016
Accepted 20 July 2016

Keywords:
Complex Event Processing (CEP)
Information filtering
Monitoring systems
Semantic Web (SW) technologies

ABSTRACT

Background: Present day control and monitoring systems are equipped with a large number of heterogeneous devices and are operated by many users with different roles and responsibilities. The information generated by these devices, although preprocessed and filtered, is usually delivered to users regardless of their actual information needs, thus overwhelming cognitive capacities and potentially affecting safety of the system.
Objectives: This work aims to reduce information load to the users of the data-intensive monitoring systems by delivering selected information to each user based on his/her roles in the system and responsibilities.
Methods: The proposed approach combines Semantic Web (SW) technologies and Complex Event Processing (CEP) for configuration purposes and run-time analyzing. The approach is exemplified with implemented tools and feasibility study based on the performance tests. The paper describes principles of proposed approach, demonstrates illustrative scenarios from building automation domain, gives description of implemented tools, and presents results of the initial performance tests.
Results: The combination of SW and CEP brings two major advantages: (1) the behavior of the system could be easily changed by configuring only underlying ontology and (2) utilization of CEP at runtime makes system event-driven and reactive to frequent changes in the environment. The performance tests demonstrated the response time of implemented tools within one second for 1000 updates per second (which corresponded to 10,000 devices in the performed experiments).
Conclusions: It is expected that the proposed approach is able to make monitoring systems personal oriented and thus safer during the operation. The results of the performance tests suggest feasibility of the approach for such systems as building and home automation, and non-critical industrial automation.

1. Introduction

Present day control and monitoring systems are equipped with multifarious devices, providing information about different aspects of the environment. The problem arises, when this information, although preprocessed and filtered, is fully delivered to users of the system, overwhelming cognitive capacities and potentially affecting safety of the system. The information overload is relevant in several domains of the monitoring systems, including process automation, discrete manufacturing and building automation (BA) systems. In the process automation industry around 30–90% of system failures are attributed to operator errors, most of which appear due to information overload [1–3]; while discrete manufacturing and BA systems operators suffer from the need of monitoring tens of thousands devices and addressing arising problems in the timely manner [4,5]. This paper addresses the problem of selective information delivery based on user role in the system and their information needs and proposes hybrid approach combining Semantic Web (SW) technologies and Complex Event Processing (CEP) for solving it.

The evolution of ICT has opened new possibilities for control and monitoring systems. Ethernet and IP-based networks have enabled easier integration of various devices, thus, systems are capable to deliver almost any information about environment, process of interest or even operators and other actors of the system [6]. The availability and diversity of information have made systems attractive to more users and user groups, who now have the possibility to find specific information assisting in their daily tasks. However, most of the monitoring systems are not capable of tailoring information flows to user needs and requirements.

[*] Corresponding author.
E-mail addresses: yulia.evchina@tut.fi (Y. Evchina), jose.lastra@tut.fi (J.L. Martinez Lastra).

http://dx.doi.org/10.1016/j.aei.2016.07.006
1474-0346/© 2016 Elsevier Ltd. All rights reserved.

Instead, almost all available information, although preprocessed and filtered on a field level, is delivered to each and every user regardless of their roles and responsibilities. In the best case scenario, monitoring systems have configuration abilities to control information flow for different user groups, but they do not consider individual information needs and are not able to address some short term requirements in information delivery, such as temporal reallocation of duties (e.g. due to someone's sick leave) or delivering of specific information to visiting workers (e.g. maintenance provided by outside companies). As a result, such systems create high information loads to the users, overwhelming cognitive capacities and complicating daily routine [6,7]. Therefore, monitoring systems require new ways of handling information streams, which would enable selective information delivery with (re)configuration capabilities.

This paper deals with selective information delivery in data-intensive monitoring systems. In this paper, the term "data-intensive" refers to the systems with large number of heterogeneous devices each producing at least one information stream, rather than devices producing highly intensive information streams. Due to large number of devices and their variety, such systems have additional properties related to information streams: (1) the variety of information about various aspects of the environment makes these systems potentially useful for different users and user groups, therefore the systems have multiple users with various information needs and (2) the large numbers of devices and users create more potential situations, when something has to be changed in the system (i.e. new requirements from existing or new users, new devices, etc.), therefore the requirements in delivery of information streams might change over time reflecting needs of the users or changed properties of devices. Thus, the problem of selective information delivery in data-intensive monitoring system could be formulated as delivery of multiple information streams to multiple users with respect to individual information needs and possible changes within these environments and/or user information needs.

This work presents a middleware capable to deliver information tailored to user needs based on his/her role and responsibilities. The approach combines SW technologies and CEP. The combination of mentioned technologies brings two major advantages: (1) the behavior of the middleware could be easily changed by configuring only underlying ontology and (2) utilization of CEP at runtime makes middleware event-driven and reactive to changes in the environment. The original contributions of the paper could be summarized as follows: (1) the proposed methodology for selective information delivery, featuring easier system reconfiguration, is described in detail with the guidelines for adopting it for related domains and applications; (2) the architecture and corresponding software implementation supporting proposed methodology are presented; and (3) the implemented software is evaluated with performance tests, demonstrating feasibility of the proposed methodology for the environments with numerous devices.

The rest of the paper is organized as follows. Section 2 presents a brief overview of related work and justifies taken approach. Section 3 presents detailed principles of combination of SW technologies and CEP. Section 4 describes functional architecture of the middleware, which is followed by Section 5 describing implementation. Section 6 presents examples of reconfiguration scenarios from a building automation (BA) domain. Section 7 presents the results of empirical evaluation of suggested approach. Finally, Section 8 concludes the paper and highlights directions for future work.

2. Related work

The problem of selective information delivery, when selection should be done out of many options, is relevant in many application domains. These examples include searching information in the Internet, medical diagnostic applications, delivery of information and/or services depending on context, etc. Thus, the approaches to solve the problem are numerous, and the right choice depends on application requirements and context of use. The specific features of data-intensive monitoring systems are (1) the large number and variety of devices in the system and its operators, (2) the run-time flow of information from the environment devices, which should be delivered in timely manner and (3) the possible alterations in the system due to changing rights or duties of users, adding/configuring new devices, etc. Based on these criteria we have highlighted five main approaches, which might be used in selective information delivery: SW technologies, publish/-subscribe middleware, CEP, Linked Stream Data, and hybrid solutions.

2.1. Semantic Web technologies

Semantic Web technologies are the family of W3C recommendation standards for describing and relating data on the Web and inside enterprises. Among these standards, the most popular are OWL[1] ontologies and SPARQL[2] queries. Although SW technologies were originally developed for describing and relating static information on the web, it is possible to use it within dynamic systems with additional technologies such as SPARQL/Update[3] allowing ontologies to reflect present state of the environment, and SWRL[4] rules for expressing conditional rules within OWL languages. The advantages of using Semantic Web technologies are in expressivity of knowledge models, which are human and machine readable. This prompts easier reconfiguration process, when changes made by human could be instantly interpreted by machine. Usually, systems based on Semantic Web technologies model situations of interest into an ontology, which describes and links various concepts together, and query this model for specific information, which is used for changing behavior of the system. The examples of such applications are numerous: home health monitoring [8], industrial applications [9–12], road traffic management [13], and delivering personalized services [14–17] to name a few.

SW technologies is a powerful tool for expressing concepts and their relationships, however there are limitations in temporal reasoning support, especially within context of dynamic information flows. It is a challenge to express temporal relationships among events such as "*before*", "*after*", and "*overlaps*". There are attempts to overcome these challenges by using extensions of SWRL language as in [18] or by adding additional models or reasoners on top of the ontological model as in [13,19,20]; however, the application is still limited and there is no standardized approach to reason with time constraints.

2.2. Publish/subscribe middleware

The publish/subscribe middleware are solutions providing asynchronous communication between multiple entities of the system. The approach is especially popular in Wireless Sensor Networks, which consist of multiple communicating devices and require efficient ways of communication [21–23]. The publishers produce information to the consumers (subscribers). Usually, publishers and subscribers are unaware of each other, i.e. publisher does not know to whom it produces information, while consumer is unaware from where the piece of information came. The

[1] Web Ontology Language (OWL): http://www.w3.org/2004/OWL/.

[2] SPARQL Protocol and RDF Query Language: http://www.w3.org/wiki/SPARQL.

[3] SPARQL 1.1 Update, update language for RDF graphs: http://www.w3.org/TR/sparql11-update/.

[4] Semantic Web Rule Language: http://www.w3.org/Submission/SWRL/.

communication is usually done through the set of message brokers, which handle subscriptions and redirect the messages. There are several possible ways of how subscriber declares interests in the certain type of messages. The most common subscription mechanisms are topic and content based. In the topic based mechanism consumer subscribes to certain types of topics (or channels), while in the content based mechanisms the delivered messages are filtered based on the actual content of the message. Thus, the selective delivery of information could be achieved by client subscriptions to the certain types of published messages.

The publish/subscribe paradigm is a widespread approach in distributed, event-based applications, thus brining research interests into various aspects of publish/subscribe communication. For example, some of the works are concerned with providing scalability of solution in terms of number of subscribers and message rates [24] or modeling communication networks for more efficient message routing [23]; while the others propose approaches for generalized performance evaluation of publish/subscribe systems [25]. Also, quite intensive topic of research is in increasing flexibility of subscription mechanisms. For example, in [26] authors propose the graph-bases subscription mechanism, where the graphs model part of the system entities and simplify subscriptions for users interested in certain type of the information; the work described in [24] deals with semantically annotated data streams, and also proposes filtering mechanisms based on semantics of published data.

While publish/subscribe is a powerful approach in redirecting and delivering streams of information in distributed systems, it has some limitations in application to selective delivery in monitoring systems. The limitation comes from the fact that publish/subscribe paradigm is originally developed as a way of communication, not the analysis of data streams; thus, the capabilities to perform ongoing analysis of data streams are pretty much limited to content based filtering of information, and are not capable to detect complex interrelated patterns of events between several streams.

2.3. Complex event processing

Complex Event Processing (CEP) is a defined set of tools and techniques for analyzing of complex series of interrelated events and matching them with predefined patterns or rules [27]. This approach overcomes difficulty of analyzing temporal relationships and makes systems reactive to events as opposed to SW technologies, where knowledge model needs to be polled repeatedly. Usually CEP engines are standalone tools, which might be easily adopted by application via provided API. They provide scalable infrastructure for analyzing streams of different events. CEP applications react according to detected patterns of interest, and in this way could be used for selective information delivery. CEP approaches are especially useful, when separate readings are not self-descriptive, e.g. when systems utilize numerous RFID sensors for detecting situations as in [28–30], or detection of a situation includes large number of heterogeneous sensors as in [31].

The difficulty of using CEP comes from engine specific rule languages. Usually, each CEP engine has its own rule language and features in configuration, thus challenging creation of rules for environments with heterogeneous devices and multifarious situations [32,33]. Although, there are attempts to unify rule description by proposing higher level models, such as in [34], or standards as in RIF[5] Working Group, the lack of interoperability and simplicity of rule languages is still an issue.

2.4. Linked Stream Data

Linked Stream Data is emerging area of research that combines stream reasoning over semantic models [35], thus bringing together advantages of CEP and Semantic Web technologies. The idea is to format data streams in common way and publish them as streams of URI-based tuples, enabling linking of these streams to knowledge models [36]. Applications using Linked Stream Data attempt to reason over newly available on the Web streams of information from devices and catch events of interest in respect to temporal constraints and underlying knowledge models [37–39].

Potentially, Linked Stream Data technologies are able to bring together stream processing and semantic reasoning, however the reasoning process over knowledge models is computationally expensive, and existing solutions implement only reduced number of reasoning capabilities or none at all [40]. Thus, the technology is still immature and requires further investigation.

2.5. Hybrid approaches

Hybrid approaches use existing technologies and tools and combine them together for achieving better results. For example, combination of ontologies and fuzzy logic enables representation of approximated information in the ontologies [41–43]. In [44] the ontologies and machine learning techniques are used to represent user social data and to provide personalized recommendations on friend's social activities. The combination of ontologies and CEP was used in [45], where information from ontologies is translated into native CEP language used in the work with the purpose to facilitate configuration of CEP engine. All of the mentioned works blend several technologies together to overcome certain limitations of existing technologies with respect to particular applications and its requirements; however, they lack the generalized approach and are quite specialized to application specific requirements, which makes them hard to adopt for other domains and/or applications.

2.6. Summary

The solution for selective information delivery in data-intensive monitoring systems should comply with the following requirements: (1) it should provide ways to deal with multiple devices and multiple users of the system; (2) it should be reconfigurable to reflect changes in the environment and/or user information needs; and (3) the device updates should be delivered to users within reasonable amount of time (e.g. within few seconds for the building automation domain). It is further assumed, that information streams are already preprocessed (i.e. no noise reduction or data transformation are needed).

The SW technologies are able to provide ways for representation of data about multiple devices and multiple users on the site; moreover, human-machine readable models could be modified quite easily to reflect changes in the environment or user requirements. However, for large scale systems with multiple information streams, continuous polling of knowledge model for reading device updates could slow down the response time of the whole system, thus challenging the pure SW implementation of selective information delivery. The public/subscribe approaches is a powerful way of redirecting information streams in distributed systems; however, it lacks analyzing capabilities of event patterns in cases, where situation should be tracked based on values from multiple sensors. The CEP technology is able to process numerous streams of data in a timely manner; however, configuration of CEP rules to changing requirements in large-scale systems is error prone and requires expertise in chosen type of CEP engine. The Linked

[5] Rule Interchange Format (RIF): http://www.w3.org/TR/rif-overview/.

Stream Data approach potentially bridges advantages of SW technologies and CEP; however, it is still immature and is not able to compete neither with expressiveness of SW approaches nor with performance of CEP engines.

Considering statements above, this paper proposes combination of SW technologies and CEP for selective delivery of information streams in data-intensive monitoring systems based on user roles and responsibilities. It is expected that the combination of SW technologies and CEP will potentially bring two major advantages: (1) the easier reconfiguration of the system, when only knowledge model must be changed and (2) run-time event processing of large numbers of information streams. The combination is done through abstraction of general concepts of the monitoring systems such as devices, users, and information streams. These abstractions are used as a link between knowledge model of the middleware and run-time event processing. Unlike the direct reconfiguration of CEP rules proposed in [45], ontologies are used to store information about the entities of the system, which is later used to fill in the properties of data abstractions objects. This separation makes knowledge model more intuitive to the users, simplifying reconfiguration process and adding more flexibility to it; while CEP provides runtime analyzing capabilities with strong temporal reasoning support. The next section of the paper presents detailed description of combination principles of SW technologies and CEP used in this work.

3. Principles of hybrid approach

The approach is based on combination of key advantages of SW technologies and CEP: expressivity of knowledge models and analyzing capabilities of CEP at run-time. The technologies complement each other through data abstractions concepts, which provide the link between SW and CEP.

3.1. Data abstractions

Data abstractions are general concepts of the monitoring system with properties. At present, the approach abstracts four main concepts: devices, users, information stream data, and places. These concepts capture key properties necessary for selective delivery of information in monitoring systems. For example, *User* abstraction reflects roles and responsibilities of each user in the system. *Device* abstraction has information about location of the device, type of the device, the IDs of the information streams, which this device has. The *StreamData* contains information of new data reading. Since data-intensive monitoring systems have large number of devices, the *StreamData* has only minimum set of properties, which are stream ID, value of the reading and its timestamp. *Place* abstraction describes locations of the environment, e.g. buildings, apartments, rooms, etc.

From implementation point of view, data abstractions correspond to classes of OOP paradigm, and thus could be easily implemented in any required environment. The detailed information on adopted abstractions and their properties could be found from Table 1.

It is worth to notice, that some of the properties are actually list of other concepts. For example, each *Device* has list of places (*placeList* property), which reflects that the same device is located in several places (i.e. the same device might have location in room, apartment, floor, building, etc.). Thus, properties in the form of list of concepts allow describing of multiple relationships of the same type as in semantic models.

The objects of data abstraction classes do not directly communicate with each other, instead they serve as a bridge between knowledge model of the system (which is used as a source of information for properties of data abstraction objects) and CEP engine (the rules of which operate with data abstractions classes).

3.2. SW technologies usage

Semantic Web technologies enable description of concepts and their relationships between each other, and accessing this data in standardized manner. Linking information through concepts and relationships provides flexible ways for data modeling. In the proposed approach, SW technologies are used for describing information about the system, e.g. its users, devices, places and information streams. This information is later extracted through series of queries and is used to initialize objects of data abstraction classes described above. The next subsections describe knowledge model of the middleware and queries used to obtain data from this knowledge model.

3.2.1. Application ontology

Ontology is a knowledge representation technique, which enables expression of concepts and their relationships in various domains. To answer the question of selective information delivery based on user role and responsibilities, the solution should be able to operate with specific terminology and relationships, which corresponds to application ontology type according to classification proposed in [46].

The application ontology of the middleware contains five concepts for describing information in the system: *Users*, *UserRoles*, *Places*, *Devices* and *DataStreams* (Table 2). The *Users* concept is used for representation of users of the system; it is separated from *UserRoles* concept assuming that each user might have several roles within the system. The *Places* concept is used for semantic description of physical layout of monitoring system (e.g. buildings, rooms, apartments, common areas, etc.). The *Devices* concept represents devices of the system, which have information streams described via *DataStreams* concept. It is assumed, that each concept has underlying taxonomy classifying related concept in the domain. For example, the *Devices* might be extended with Semantic Sensor Network (SSN[6]) ontology classifying sensors in the system [47] or IoT-Light[7] ontology describing Internet of Things (IoT) devices and resources, while *DataStreams* concept might be extended, e.g. with Stream Annotation Ontology proposed in [48]. In this way, the middleware is able to operate over various semantic knowledge models of different systems and still keep core application ontology quite small and manageable.

The application ontology of the middleware has also relationships for describing possible connections among individuals and concepts. OWL ontologies allow three types of relationships: object property, data property and annotation property relationships. The object property relationships are assigned between individuals of the ontology. In the proposed middleware, this type of relationship is used to connect certain places, devices, and user responsibilities between each other (Table 3). The data property relationship connects individuals to data values of the specified type; in the middleware, this type of relationships is used for information, such as IDs and user friendly descriptions of ontology individuals. Finally, annotation property relationships might link any type of ontological concepts including individuals, data types and classes of ontology. This type of relationship is used to express responsibilities of users for classes of devices and/or places (e.g. to monitor data from all devices of specific type).

[6] Semantic Sensor Network (SSN) ontology: http://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

[7] IoT-Light Ontology: https://www.w3.org/Submission/2015/SUBM-iot-lite-20151126/.

Table 1
Data abstractions with properties.

Data abstraction	Description of data abstraction	Property	Description of property
User	The user abstraction is used to describe roles and responsibilities of each user in the system	userId roleList monitoredPlaceIdList monitoredDeviceIdList	ID of the user in the system List of roles, which user has IDs of all places, the user is supposed to monitor (in other words, user monitors all devices of these places) IDs of all devices, the user is supposed to monitor (in other words, the user monitors specific devices regardless of places, where device is located)
StreamData	StreamData contains new readings from devices	streamId value timestamp	ID of the data stream Value of new reading Timestamp of new reading
Device	Device abstraction has information for describing device class and location. It also has description of information streams, which this device has, serving as a link between new StreamData and device info	deviceId deviceType placeList streamDescriptorList	ID of the device in the system Type of the device (e.g. temperature sensor) List of places, where device is located (consists of the Place objects) List of stream descriptors, which device has (consists of the streamDescriptor objects)
Place	Place has information describing each particular location in the system	placeId placeDisplayName placeClass	ID of the place in the system The descriptive name of the place The category of the place, e.g. “Buildings”, “Apartments” etc.
StreamDescriptor	The helper concept for linking of new StreamDatas with particular Device (usually via stream ID)	streamId streamClass unit	ID of the information stream from device Class of the stream (e.g. temperature sensor stream) Unit of the data in the stream

Table 2
Concepts of application ontology.

Ontology class	Description
Users	Represent users of the system
UserRoles	Represent possible roles in the system. Two individuals of the same class (e.g. “maintenanceRole1″, “maintenanceRole2″) reflect different responsibilities of the same role type
Places	Represent places of the system (buildings, apartments, rooms, etc.)
Devices	Represent devices of the system
DataStreams	Represent data streams from devices of the system

The proposed ontology defines general concepts and relationships used for selective information delivery. The combination of ontology, underlying taxonomies, their individuals, and relationships among those individuals form knowledge model of particular monitoring system, which serves as a source of information for data abstractions objects. Next subsection presents how this information is extracted from knowledge model.

3.2.2. Querying ontology

The application ontology has defined set of classes and relationships. These classes and relationships are used in query templates for retrieving specific information from ontology. Query templates are small queries with variables, which are substituted with values for completing the query and retrieving information about specific objects. The possible examples of query templates are given in Table 4.

The query results might be also controlled with semantic reasoner. For example, query #1 for *Devices* as specified type and with

Table 3
Relationships of the application ontology.

Ontology relationship	Type	Domains	Ranges	Description
containsDevice	Object property	Places	Devices	Describes location of devices
containsPlace	Object property	Places	Places	Describes relationships between locations
hasDataStream	Object property	Devices	DataStreams	Relates devices to data streams
hasRole	Object property	Users	Roles	Relates users and user roles
monitorsDevice	Object property	Roles	Devices	Describes role’s responsibility to monitor particular device
monitorsPlace	Object property	Roles	Places	Describes role’s responsibility to monitor place (and hence, all underlying places and devices located in those places)
hasId	Data property	DataStreams, Devices	string	Relates data streams and devices to IDs in the system
hasDisplayName	Data property	Any	string	Relates objects to friendly names (e.g. instead of IDs)
hasUnit	Data property	DataStreams	string	Contains units of specific data stream
monitorsDeviceClass	Annotation property	Roles	Devices	Connects specific roles to classes of devices for describing responsibilities to monitor devices of specific type regardless of location
monitorsPlaceClass	Annotation property	Roles	Places	Connects specific roles to classes of places for describing responsibilities to monitor places of specific type (e.g. all apartments, all common rooms, etc.)

active semantic reasoner will return all the objects inferred as *Devices*, i.e. all the objects of underlying *Devices* taxonomy; while the same query without semantic reasoner would return only objects, which were explicitly assigned to *Devices* type, i.e. no objects of underlying taxonomies with other type would be returned. Thus, part of the data logic and relationships could be inferred with semantic reasoning, further adding expressivity to ontology models.

The series of queries has to be performed, in order to retrieve data from application ontology and initialize data abstractions objects. For example, to initialize User objects the following queries should be invoked:

(1) Retrieve all user IDs represented in the ontology (query #1 from Table 4 with "*Users*" as type with active semantic reasoner).
(2) For each retrieved user ID initialize *roleList*: retrieve all specified user roles (query #2 with user ID and "*hasRole*" as object and relationship respectively).
(3) For each user role retrieve *monitoredPlaceIdList* and *monitoredDeviceIdList*:
    (a) Use series of query #2 with role ID and such relationships as "*monitorsPlace*", "*monitorsPlaceClass*", "*monitorsDevice*", and "*monitorsDeviceClass*"; and fill in corresponding properties of object.

The result of the steps above is the list of objects representing all users in the system. The similar set of queries is used, for example, to initialize *Device* objects with data from ontology.

Keeping information about the system in ontology and retrieving it through query templates decouples system information from application logic required in selective information delivery. This decoupling enables easier reconfiguration of the system, when simple changes in individuals and relationships in the ontology are reflected in properties of data abstractions without the need to change application logic or recompile underlying tools.

3.2.3. Example of knowledge model

The knowledge model of particular monitoring system with taxonomies, individuals and relationships builds intuitive semantic structure of this system. The model could be changed reflecting changes in user and/or system requirements. This section gives example of knowledge model of a monitoring system and how changes in the knowledge model are reflected in the properties of data abstractions objects.

Considering example from BA domain, it is assumed that BA system has buildings with apartments and various installed devices. The buildings are monitored by maintenance personnel with different responsibilities. The part of example knowledge model with simple underlying taxonomies is presented in Fig. 1. As it was mentioned in Section 3.2.1, the small application ontology could be extended with underlying taxonomies of such ontologies as SSN or IoT-Light. However, in order to provide simple illustrative example of knowledge model from BA domain, the application ontology was extended with small custom taxonomies, part of which is represented in rounded rectangles of Fig. 1.

The model reflects layout of places and coarse location of devices with respect to those places. For example, from Fig. 1 it is seen that building 1 (*B1*) has two apartments in it (*A1* and *A2*), and each of them has set of installed devices with data streams. It is possible to improve granularity of location by extending *Places* taxonomy and using more precise definitions of places inside apartments, rooms, etc. depending on domain of monitoring system.

The model describes three users of the system, two of which have maintenance role (*M1* and *M2*), and one has visiting maintenance role (*VM1*). The responsibilities of each specific role might be freely defined by relationships in the application ontology. For example, the maintenance role 1 (*M1*) has responsibility to monitor building 1, thus the user having role *M1* will receive data from all devices located in *B1* (and, consequently, in *A1* and *A2*). In similar way, *monitorsDeviceClass* annotation property indicates that user with visiting maintenance role *VM1* should receive data from all temperature sensors in the location. Roles might combine various responsibilities, for instance, maintenance role might have monitoring of place and certain type of devices (or specific device from other place). Also user might have several roles in the system, which further adds flexibility to possible reconfiguration scenarios.

Table 4
Examples of SPARQL query templates for application ontology.

#	Query template[a]	Variables description	Query description

1.	SELECT ?object
	WHERE {
	?object rdf:type ao:**TYPE** .}
		TYPE is replaced with needed type of objects (e.g. *Devices*, *Users*)	The query returns all objects of the specified type

2.	SELECT ?objects
	WHERE {
	ao:**OBJECT** ao:**RELATIONSHIP** ?objects .}
		OBJECT and RELATIONSHIP are specified values	The query returns all objects, which connect specified object via specified relationship

3.	SELECT ?class
	WHERE {
	ao:**OBJECT** rdf:type ?class .
	filter(! strstarts(str(?class),str(owl:))) }
		OBJECT is replaced with object ID	The query returns classes of specified object. The filter option is used to exclude owl prefix results, such as owl: Thing and owl: NamedIndividual

4.	SELECT ?places
	WHERE {
	ao:**PLACE** ao:containsPlace* ?places .}
	ORDER BY ?place
		PLACE is replaced with ID of place	The query returns set of places, which specified place contains. The star property path (*) indicates, that the query will return the IDs of objects in the whole path connected via *containsPlace* relationship

5.	SELECT ?location ?displayName ?class
	WHERE {
	?location3 ao:containsDevice ao:**DEVICE** .
	?location2 ao:containsPlace* ?location3 .
	?location ao:containsPlace* ?location2 .
	?location rdf:type ?class .
	?location ao:hasDisplayName ?displayName .
	filter(! strstarts(str(?class),str(owl:)))}
	GROUP BY ?location ?displayName ?class
	ORDER BY DESC(count(?location2))
		DEVICE is replaced with device ID	The query returns ordered set of location IDs, description names and location classes for specified device. It orders returned result according to path length from device in descended order (e.g. if device is located in Room1 of Apartment1 of Building1, the returned set will contain information of places starting from Building1 to Room1)

[a] For simplicity, the prefixes of queries, such as rdf and ao (states for application ontology) are omitted from the query; however it is assumed that they are presented in each query.

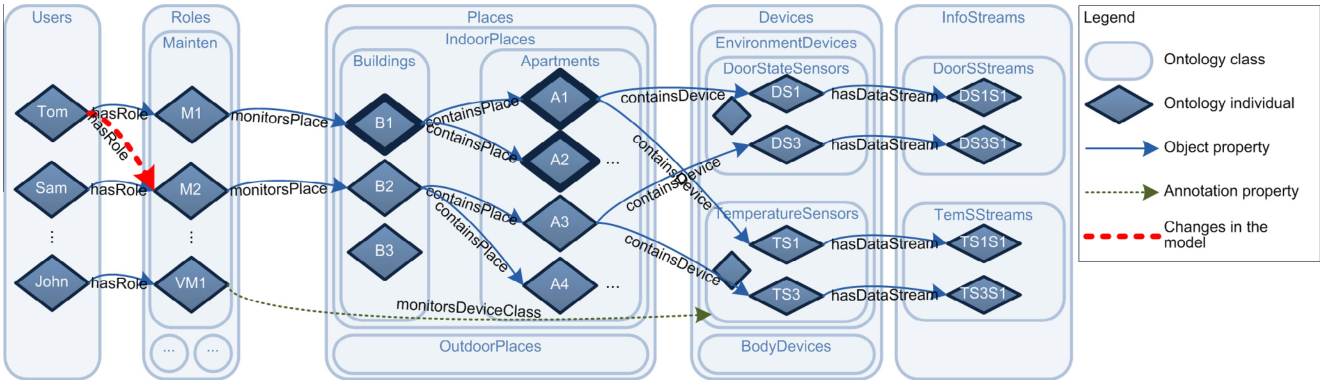


Fig. 1. Part of example knowledge model.

In the given example, when ontology is queried for user specific information, three User objects are formed:

```
User(userId = Tom, roleList = [M1], monitoredPlaceIdList = [B1, A1, A2], monitoredDeviceIdList = [])
User(userId = Sam, roleList = [M2], monitoredPlaceIdList = [B2, A3, A4], monitoredDeviceIdList = [])
User(userId = John, roleList = [VM1], monitoredPlaceIdList = [], monitoredDeviceIdList = [TS1, TS2])
```

If some changes has to be made in information delivery, e.g. one of the users has to take additional role in the system as shown in bold dashed arrow in Fig. 1, the changes in knowledge model will be reflected in properties of data abstractions. In the given example, when user *Tom* takes also role *M2*, the following User object will be constructed:

```
User(userId = Tom, roleList = [M1, M2], monitoredPlaceIdList = [B1, B2, A1, A2, A3, A4], monitoredDeviceIdList = [])
```

The same approach applies to other data abstractions objects, such as devices and places. For example, when new device must be added in the system, only individual of that device and data stream should be added with proper relationships between device, data stream and location place; the queries will automatically build device objects with correct list of all places for device.

Building knowledge model with layout of places and description of all devices might be a time consuming task, however it could be automated with software using SPARQL/Update as language for building ontology. After building the initial knowledge model, the further reconfiguration changes are quite small and manageable, making this approach especially useful in large-scale systems.

3.3. CEP usage

CEP is the method for analyzing run-time events and matching them with predefined patterns or rules. In the proposed approach, CEP provides run-time analysis of device updates in the system. CEP rules operate on properties of data abstractions. Various combinations of properties and rule conditions enable selective delivery of information based on roles and responsibilities of users. Thus, the CEP engine analyzes objects of data abstraction classes, which are initialized with information from ontology and run-time updates from devices. The examples of rules and their description are given in Table 5 (the CEP engine used in this work is Drools, thus Table 5 contains rules with Drools syntax).

The rules in Table 5 give an example of redirection of device updates and notifications based on user responsibilities in the system and roles. The first two rules of Table 5 give an example of redirecting information streams based on user roles and responsibilities, while the third rule gives an example of monitoring of abnormal situations in the system and delivery of corresponding notifications based on user roles. It is worth to notice, that abnormal situations could be detected by monitoring values of several streams simply by combining them in one rule. While rules might appear lengthy, they are created with the uniform usage of classes representing system entities, which greatly facilitates their construction.

The rules of the middleware should be predefined during development phase, but it is not assumed that they should be changed during the reconfiguration of the middleware. i.e. If new requirements should be presented in the delivery of information streams (e.g. new user role, responsibility or new device in the system of registered device type), it is assumed that only knowledge model of the system changes. The changes in the knowledge model are reflected in properties of data abstractions objects, and thus are automatically considered in CEP engine, without need to rewrite any rules.

3.4. Adapting approach to other domains of monitoring systems

Given examples of data abstractions, SW models and CEP rules are quite general and could be applied in different domains of the monitoring systems, such as building automation and some of the discrete manufacturing systems. However, most of the cases will require adapting solution to specific requirements of the monitoring system, e.g. some of the health care related systems would require functionality to monitor patient's health status and/or filter information for user based on his/her location and responsibilities. This section presents a set of steps for adapting approach to particular monitoring system. The steps are presented in unnumbered order, meaning that they could be rearranged and repeated depending on specific system complexity and requirements.

- Analyze requirements of the information delivery and derive, which factors should be considered in selective delivery (e.g. roles, location of user, etc.);
- Check data abstractions concepts and augment, if needed, with new properties (e.g. user location, stress level);
- Check application ontology and add missing concepts and/or relationships (e.g. in case of health care monitoring it could be *monitorsUser* relationship);

Table 5
Examples of CEP rules for redirecting of device updates.

<table>
<tr><th>#</th><th>Rule</th><th>Description</th><th>Drools implementation</th></tr>
<tr><td>1.</td><td>Update device data for user, monitoring place, where device is located</td><td>The update notification is sent, when there is data from stream (<i>StreamData</i> class), the device has this stream (<i>streamId</i> variable), and there is user, who monitors place, where device is located</td><td>rule "When stream data is updated, trigger device status
notification for users, monitoring device's PLACE"
when
   $sData : StreamData()
   $device : Device()
   //device has the same streamId as given stream
   $streamDescriptor : StreamDescriptor(streamId ==
sData.getStreamId()) from $device.streamDescriptorList
   //there is user, who monitors place, where device is
located
   $user : User()
   $place : Place($user.getMonitoredPlaceIdList contains
placeId) from $device.getPlaceList()
then
   notificationsUtility.updateDeviceStatusForUser($device,
$sData, $user);
end</td></tr>
<tr><td>2.</td><td>Update device data for user, monitoring this specific device</td><td>The update notification is sent, when there is data from stream (<i>StreamData</i> class), the device has this stream (<i>streamId</i>), and there is user, who should monitor this device</td><td>rule "When stream data is updated, trigger device status
notification for users, monitoring this DEVICE"
when
   $sData : StreamData()
   $device : Device()
   //device has the same streamId as given stream
   $streamDescriptor : StreamDescriptor(streamId ==
sData.getStreamId()) from $device.streamDescriptorList
   //there is user, who should monitor device
   $user : User($user.getMonitoredDeviceIdList contains
$device.getDeviceId())
then
   notificationsUtility.updateDeviceStatusForUser($device,
$sData, $user);
end</td></tr>
<tr><td>3</td><td>Trigger fire alarm if indoor temperature is higher than 40°</td><td>The fire alarm notification to specific user is created, when all of the following conditions are met: (1) the indoor temperature device has value greater or equal to 40°, (2) this device is located in the apartments, or kitchen, or laundry, or dining rooms, and (3) one of the user roles is either maintenance or resident. The notification is active, while conditions for alarm exist. When conditions disappear, the notification is automatically retracted from CEP memory</td><td>rule "When temperature is more than 40 degrees, trigger fire
alarm of priority 3 to all users with MaintenanceRoles and
ResidentRoles"
when
   $device : Device(deviceClass == "IndoorTemperatureSensors")
   $streamDescriptor : StreamDescriptor() from
$device.streamDescriptors
   $sData : StreamData((streamId ==
$streamDescriptor.getStreamId()) && (value.getDoubleValue() >=
40))
   $location : Place(placeOntoClass == "Apartments" ||
placeClass == "KitchenRooms" || placeClass == "LaundryRooms"
|| placeClass == "DiningRooms") from $device.places
   $user : User(roleList contains "MaintenanceRoles" ||
roleList contains "ResidentRoles")
then
   List<ContextDevice> deviceList = new
ArrayList<ContextDevice>();
   deviceList.add($device);
   List<StreamData> streamDataList = new
ArrayList<StreamData>();
   streamDataList.add($sData);
   SimpleDroolsNotification newNotification = new
SimpleDroolsNotification("FireAlarm", $location.getPlaceId(),
$user.getUserId(), 3, deviceList, streamDataList);
   insertLogical(newNotification);
end</td></tr>
</table>

- Add/modify query templates and/or initialization steps of data abstractions objects (e.g. the User object might have new property *monitorsUsersList*, which have to be initialized);
- Select domain ontologies as underlying classes of application ontology (e.g. in case of manufacturing domain, the *Places* application ontology concept could also incorporate some factory floor specific places);
- Instantiate knowledge model of the system with individuals and relationships describing specific monitoring system;
- Design CEP rules using properties of data abstractions and requirements in information delivery.

The required number of steps depends on the type of the considered monitoring system. If the system is from the same domain (e.g. other BA system), some of the adaptation steps might be skipped (e.g. modification of the application ontology and queries). However, if the system is from other domain (e.g. healthcare or discrete manufacturing), most probably, application ontology, queries, and CEP rules should be modified accordingly. The steps above describe initial middleware adaptation to a new monitoring system; they should be performed once. Afterwards, it is assumed that reconfiguration process is only about changing of a few relationships in the knowledge model, without the need to rewrite CEP rules or recompile underlying tools. The examples of reconfiguration process could be found in Section 6.

4. Functional blocks of the middleware

The middleware operates over data produced by devices of the monitoring systems and delivers personalized information streams. The approach combines SW technologies and CEP. The proposed combination principles could be grouped in functional blocks, which form a layer in the architecture of the middleware for selective information delivery. The functional blocks of this middleware are presented in Fig. 2 and are described in the following subsections.

4.1. Information Source Adapters

The main functionality of this layer is to transform data coming from various data sources to unified format adopted in the application. It consists of adapter blocks capable to read data in different formats and publish them to upper blocks of the architecture.

4.2. Ontology provider

This block holds ontological model of the monitoring system with semantic reasoner. It provides access to this knowledge model to external applications. Although ontological models are considered to be human-machine readable, they are still might be difficult to manage by technical system administrators; for these purposes it is assumed, that ontology is configured via *Configuration UI*, providing convenient access to knowledge model for reconfiguration purposes.

4.3. Run-time event processor

Run-time event processor collects updates from devices and queries *Ontology provider* for information from knowledge model. Based on this information, it initializes objects of data abstractions (with *Data abstractions initializer* block) and submits those objects for further analysis to *CEP engine* with predefined rules. The *CEP engine* analyzes all information and produces update and notification messages to specific users of the system.

4.4. Information Dispatcher

It is assumed that middleware is used by external applications, which display information properly to the user. Thus, the main functionality of this layer is to transform data from internal format to proper external messages and send them via agreed protocols.

5. Implementation

Each layer of the architecture is implemented as separate Java application running on Tomcat server, and Ontology Service running on self-launched Jetty web server (Fig. 3). Applications communicate using ZeroMQ[8] socket messaging library and web service implementation libraries.

5.1. Information Source Adapters

The application consists of series of adapters; in the present implementation there are two adapters: database and Web Services (for devices exposing data as a web services). Each adapter runs in separate thread and acquires data in specific to data source way, e.g. DB thread polls databases and publishes information to upper layers of the architecture only if there are changes, WS thread subscribes to Web Services and listens to event notifications. Each thread publishes updates of data sources via ZeroMQ publish socket. The WS adapter thread uses JMEDS[9] library to work with device web services, while DB adapter thread utilizes Spring JDBC Framework for database access.

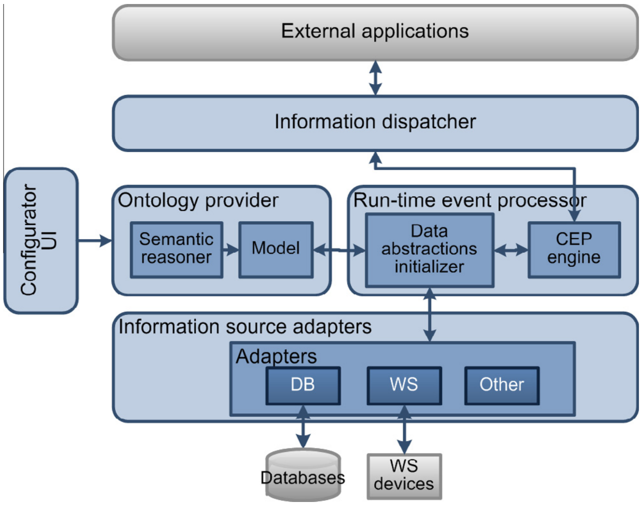


Fig. 2. Functional blocks of the middleware.

5.2. Ontology Service

Ontology Service holds ontological model of the system with semantic reasoner, and provides access to it via DPWS4J[10] library compliant with Devices Profile for Web Services (DPWS) standard. All manipulations with knowledge model are done with Jena[11] library. Ontology Service has various Web Services provided to external world, which wrap manipulations with knowledge model. Those operations include loading ontology from specified URI, updating ontology with SPARQL/Update expressions, querying ontology with SPARQL queries, setting Pellet semantic reasoner on and off, etc. These services enable flexible communications with ontology and allow various scenarios of Ontology Service usage. Present implementation does not have configuration UI yet, thus, modifications of knowledge model are done with external ontological tools, such as Protégé[12], and with SPARQL/Update queries via Ontology Service services.

5.3. Event Processor

Event Processor performs run-time analysis of available data. It queries ontology model for the information and receives updates from devices; based on this data, it initializes data abstractions objects and submits them to CEP engine (Drools in this case) for further analysis. The CEP engine processes available data and sends update and notification messages to specific users of the system.

5.4. Information Dispatcher

It is assumed that middleware is used by external applications, which display information properly to the user. The Information Dispatcher subscribes to notifications from Event Processor, transforms them from internal format to proper external messages and sends them via agreed protocols. It has internal Message Board for keeping latest messages, which are sent to the user after successful login. In the current implementation, the focus is put on delivery of

[8] ZeroMQ messaging library: http://zeromq.org/.

[9] JMEDS is Java implementation for DPWS: http://sourceforge.net/projects/ws4d-javame/.

[10] DPWS4J is a Java Web Services stack compliant with DPWS specification: https://forge.soa4d.org/projects/dpws4j/.

[11] Apache Jena (framework for building Semantic Web and Linked Data applications): https://jena.apache.org/.

[12] Protégé ontology editor: http://protege.stanford.edu/.

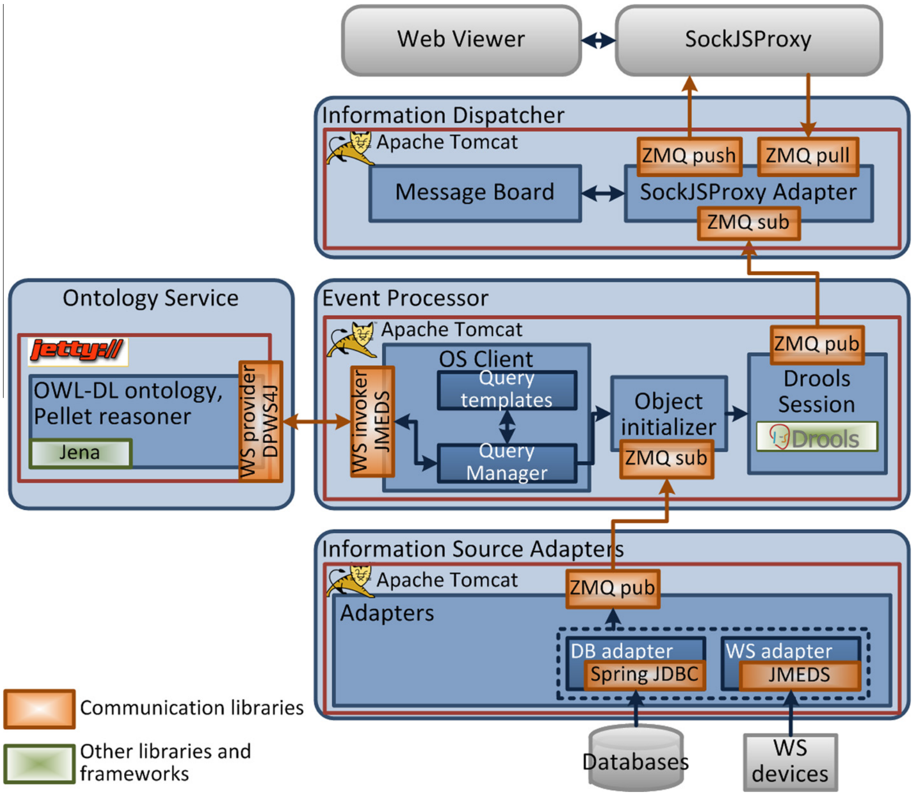


Fig. 3. Implementation architecture of the middleware.

personalized information streams. Although visualization is also an important topic of personalized information delivery, it is out of scope of this paper. Thus, the present external application displays information streams in simple tables of data. Information Dispatcher transforms data from internal format to SockJSProxy messaging protocol. SockJSProxy[13] is an external application, which enables ZeroMQ-like communication between server and browser. In the Web Viewer application, information is updated in tables with usage of JavaScript and DataTables[14] plug-in.

5.5. Message flow within the middleware

Each application in the architecture works autonomously of the others. Fig. 4 demonstrates example of message flow in the system with the focus on Event Processor and its configuration. To explain configuration process Event Processor is unfolded into two threads: Drools Session and Ontology Service Client; other applications are also multi-threaded, although it's out of scope for Fig. 4.

At start-up, each application configures ZeroMQ communication sockets and starts its internal processes. Event Processor launches Ontology Service Client and Drools Session threads. Ontology Service Client thread opens internal publish socket for Drools Session. It then discovers Ontology Service application according to WS-Discovery implementation with JMEDS, loads model of the system into Ontology Service (with *SetBaseOntology* () operation), and starts querying ontology for needed information using Query templates class, which contains templates of needed queries. The results of the queries are then transformed to Java objects and are sent to Drools Session thread for analyzing. Drools Session thread opens subscribe socket for Data Source Adapters application and Ontology Service Client thread and publish socket for Information Dispatcher. It loads predefined rules for streams analyzing and then combines updates coming from data sources and Ontology Service Client into Drools session opened in the thread. As soon as updates coming from various sources match to the rules in the session, the update notification is sent to the Information Dispatcher, which keeps latest messages for unlogged users and sends update notifications to HMI for logged in users.

6. Middleware reconfiguration examples

The proposed approach delivers personalized information streams based on user roles and responsibilities and enables easier reconfiguration. The reconfiguration of the middleware is performed by changing knowledge model of the system. This section gives some examples of changing knowledge model and describes effects of this change. The demonstrated scenarios are deliberately simple, but they illustrate key advantages of the middleware, which become crucial for real systems, when devices and users are numerous and are of many different types.

6.1. Changing roles and responsibilities of users

The present version of application ontology described in Section 3.2.1 gives several options for configuring of user roles and responsibilities. First of all, there is a possibility to assign several roles to one user of the system via *hasRole* relationship. Secondly, each role could have various responsibilities described via *monitorsDevice*, *monitorsPlace*, *monitorsDeviceClass* and *monitorsPlaceClass* relationships. The *monitorsDevice* relationship is used, when specific device has to be monitored by user. If devices are in the same location (i.e. the monitoring is based on location), then *monitorsPlace* relationship could be used, indicating, that all devices of this place are monitored. The *monitorsDeviceClass* and *monitorsPlaceClass* are used, when all devices of certain type or

[13] SockJSProxy proxy server: https://bitbucket.org/vladev/sockjsproxy.
[14] DataTables table plug-in for jQuery: http://www.datatables.net/.

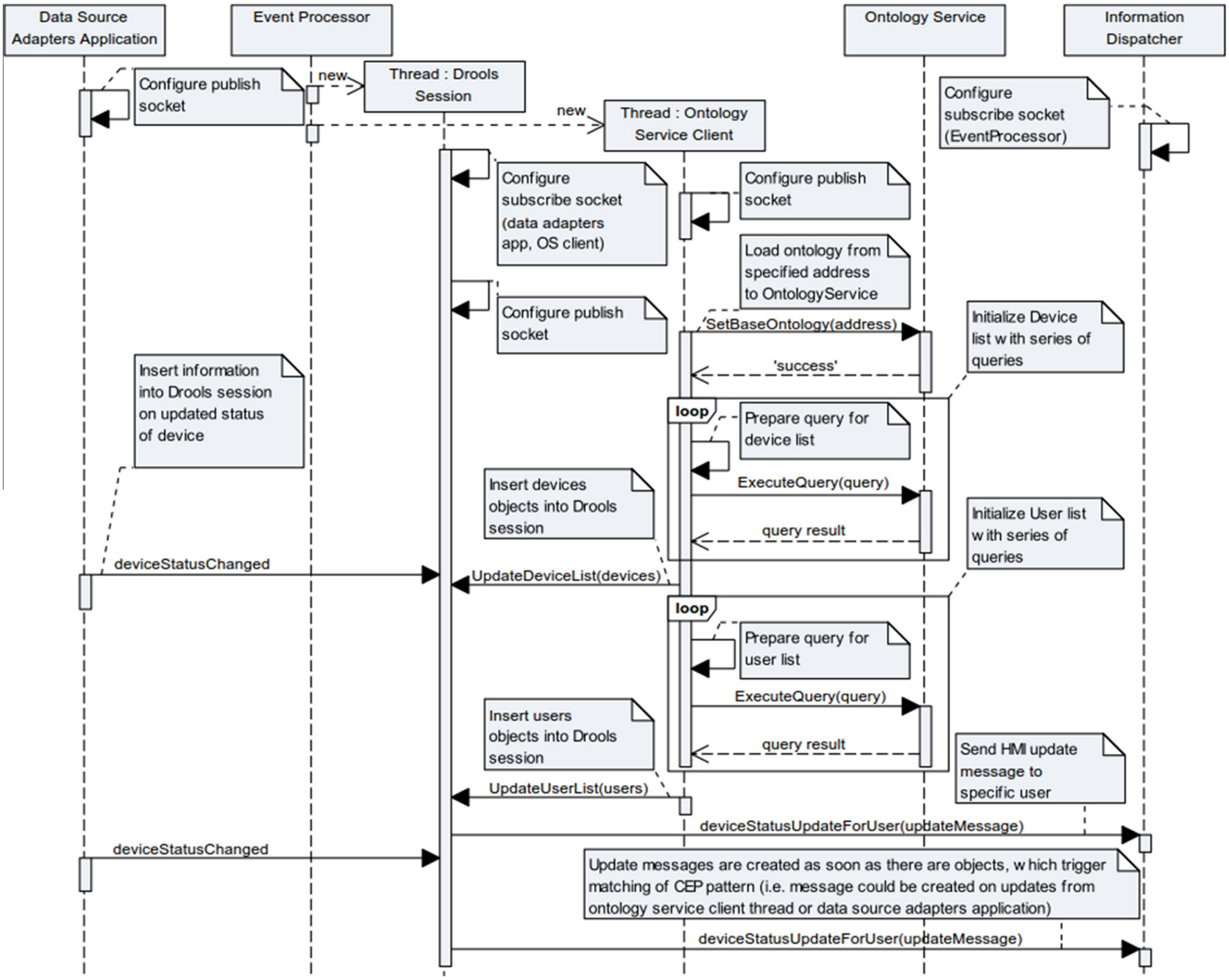

Data Source Adapters Application
Event Processor
Ontology Service
Information Dispatcher
new
Thread : Drools Session
new
Thread : Ontology Service Client
Configure publish socket
Configure subscribe socket (EventProcessor)
Configure subscribe socket (data adapters app, OS client)
Configure publish socket
Load ontology from specified address to OntologyService
Configure publish socket
SetBaseOntology(address)
'success'
Initialize Device list w ith series of queries
Insert information into Drools session on updated status of device
loop
Prepare query for device list
ExecuteQuery(query)
Insert devices objects into Drools session
query result
Initialize User list w ith series of queries
deviceStatusChanged
UpdateDeviceList(devices)
loop
Prepare query for user list
ExecuteQuery(query)
Insert users objects into Drools session
query result
Send HMI update message to specific user
UpdateUserList(users)
deviceStatusUpdateForUser(updateMessage)
deviceStatusChanged
Update messages are created as soon as there are objects, w hich trigger matching of CEP pattern (i.e. message could be created on updates from ontology service client thread or data source adapters application)
deviceStatusUpdateForUser(updateMessage)

Fig. 4. Example of message flow between implemented tools.

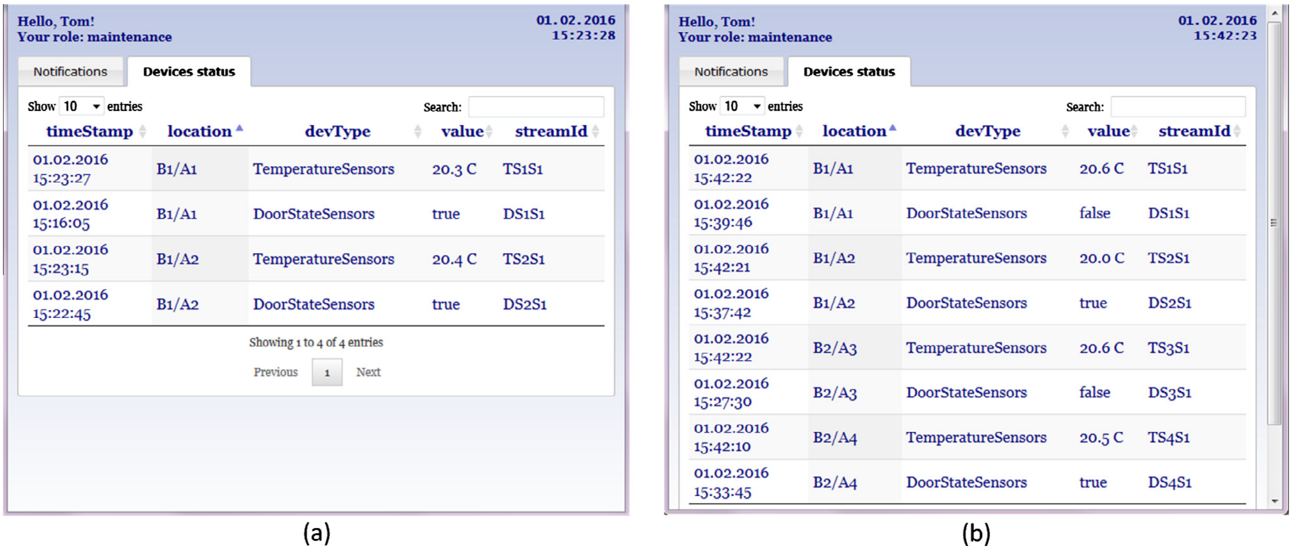
place have to be monitored rather than specific devices or places. The mentioned relationships could be also used in combination to achieve needed behavior of the system. If some of the responsibilities are not possible to express with given relationships, then middleware should be adopted to specific domain as described in Section 3.4.

Changing roles and responsibilities requires only changing of few relationships in the ontology for users, who already have been registered in the system (i.e. have their ontological representations in the knowledge model). e.g. For ontological model presented in Section 3.2.3, in order to assign additional role *M2* to user *Tom*, only one *hasRole* relationship has to be added into knowledge model between this user and assigned role (Fig. 1). This change would result in delivering of information from all devices from *B1* and *B2* to this user (Fig. 5). Thus, only by adding one relationship in the model and changing neither any underlying tools nor the CEP rules, the behavior of the system could be flexibly controlled. When number of devices, locations and users are growing dramatically (which is the case for large-scale data-intensive systems), the approach becomes powerful and flexible tool for reconfiguration of the monitoring systems. Moreover, the changes also affect delivery of notifications of detected abnormal situations (Notifications tab of Fig. 5) also without the need to additionally reconfigure any parts of the middleware.

6.2. Adding new device

It is also possible to configure new devices and/or places in the knowledge model. When new device is installed in the monitoring system, a few changes have to be made in the knowledge model: (1) Adding corresponding individuals representing this device and information stream in the ontology, (2) adding needed data properties, such as ID and description, and (3) connecting those individuals with needed place relationship. For example, for the knowledge model presented in Fig. 1, when new temperature sensor has to be added to apartment 5 (*A5*), two individuals representing sensor and information stream has to be added (*TS5* belonging to *TemperatureSensors* class, and *TS5S1* belonging to *TemSStreams* class); the needed object property relationships between individuals are *hasDataStream* between *TS5* and *TS5S1*, and *containsDevice* between *A5* and *TS5*.

When the model is queried for the information, the new device object will be automatically constructed with list of all places, where device is located; thus, users monitoring those places will automatically receive all update notifications from new device. Furthermore, if CEP engine already has predefined rule for monitoring values from temperature sensors, such as tracking fire alarms, the values of this sensor will be automatically checked for fire alarm situation for the location specified in knowledge



(a)

Hello, Tom!
Your role: maintenance

01.02.2016
15:23:28

timeStamp	location	devType	value	streamId
01.02.2016 15:23:27	B1/A1	TemperatureSensors	20.3 C	TS1S1
01.02.2016 15:16:05	B1/A1	DoorStateSensors	true	DS1S1
01.02.2016 15:23:15	B1/A2	TemperatureSensors	20.4 C	TS2S1
01.02.2016 15:22:45	B1/A2	DoorStateSensors	true	DS2S1

Showing 1 to 4 of 4 entries

(b)

Hello, Tom!
Your role: maintenance

01.02.2016
15:42:23

timeStamp	location	devType	value	streamId
01.02.2016 15:42:22	B1/A1	TemperatureSensors	20.6 C	TS1S1
01.02.2016 15:39:46	B1/A1	DoorStateSensors	false	DS1S1
01.02.2016 15:42:21	B1/A2	TemperatureSensors	20.0 C	TS2S1
01.02.2016 15:37:42	B1/A2	DoorStateSensors	true	DS2S1
01.02.2016 15:42:22	B2/A3	TemperatureSensors	20.6 C	TS3S1
01.02.2016 15:27:30	B2/A3	DoorStateSensors	false	DS3S1
01.02.2016 15:42:10	B2/A4	TemperatureSensors	20.5 C	TS4S1
01.02.2016 15:33:45	B2/A4	DoorStateSensors	true	DS4S1

Fig. 5. Changes in the delivery of information streams after reconfiguration of knowledge model: (a) user *Tom* has role *M1* with responsibility to monitor devices of building 1 (*B1*) and (b) user *Tom* has two roles (*M1* and *M2*) with responsibility to monitor devices from two buildings.

model, without need to rewrite any rules and/or recompile middleware components.

The same way as in the previous case, neither tools nor CEP rules have to be changed for the reconfiguration to take an effect. Fig. 6 gives an example of the effect of adding new *TS5* device into the system with knowledge model represented in Fig. 1. The user *Jhon* is assigned with role of visiting maintenance (*VM1*) with responsibility to monitor temperature sensors. The new added device will appear automatically in the Jhon's HMI after the reconfiguration of knowledge model. At the same time, this change will not affect HMI of the other user *Tom*, who does not have responsibility to monitor *B3* nor *TemperatureSensors*. The presented scenario is simple, but it demonstrates reconfiguration principles of adding new information into the knowledge model. When the knowledge model represents large-scale monitoring system with thousands of devices, the reconfiguration steps to change parts of knowledge model (e.g. add new devices or edit user responsibilities) are still the same, greatly simplifying system reconfiguration regardless of its size.

7. Performance evaluation

The proposed approach for selective delivery of information streams combines SW technologies for configuration purposes and CEP for run-time processing. Thus, performance evaluation was focused on configuration and run-time performance of implemented middleware.

Data-intensive systems imply high number of devices and information streams, therefore the following scenario was considered. It was assumed that facility has ten buildings with number of apartments varying from 1 to 1000 in total. Each apartment has 10 devices, thus, the number of devices varies from 10 to 10,000. The usual configuration of devices of monitoring systems is to provide updates once per several seconds; thus all devices and their values are software simulated, and have update rate once per ten seconds in average. It was further assumed that buildings are operated by four users with responsibilities varying by locations; e.g. user1 monitors 100% of locations (or all buildings), user2 monitors 50% of locations (or 5 buildings), user3 monitors 20% of locations (or 2 buildings), and, finally, user4 monitors 10% of locations (or 1 building). The two crucial performance measurements are (1) the estimation of configuration and launch time of the middleware with SW technologies and (2) the time, which is required by the middleware to deliver update from devices to the users. The tests were run locally on one machine having two CPUs with 1.9 GHz each.

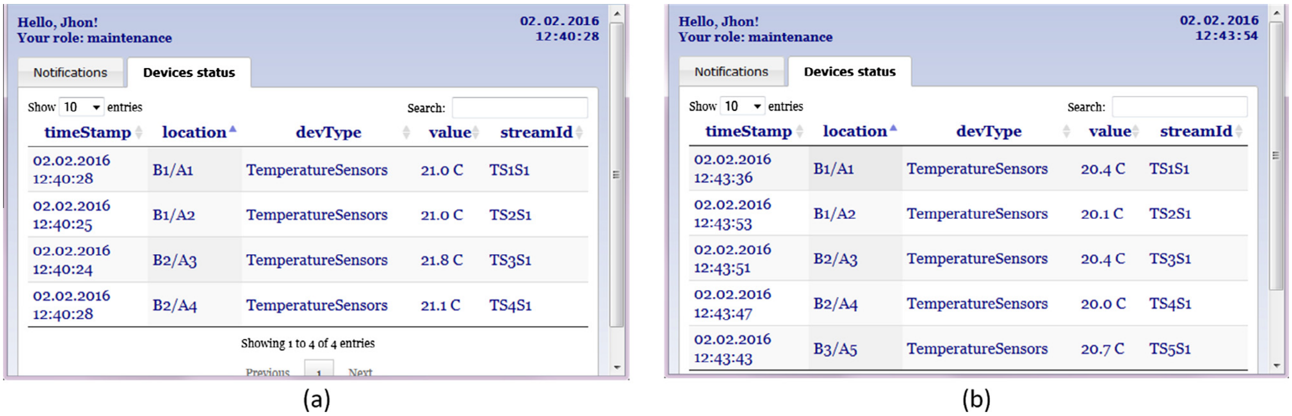
7.1. Configuration performance

The configuration performance is the time required to send query request, process the query by model holder, and receive query reply. In the presented system Ontology Service holds the model and is used during configuration time, thus the performance measurements were done on the Event Processor side with time marks of sending request to and receiving replies from Ontology Service. Although the query requests are made locally, we believe that this fact does not compromise system performance significantly, since the querying of the knowledge model is done only at the configuration stage of the middleware, while runtime sending data from devices to CEP engine is done via high throughput ZeroMQ[15] socket library.

In general, the configuration process consists of multiple steps including reasoning over knowledge model and querying of the ontology. One of the strongest factors affecting reasoning performance is the size of the knowledge model (i.e. the number of individuals) [49], while the factors affecting query performance include the number of returned by query results and query complexity [50]. During our tests, the query templates presented in Table 4 did not vary significantly in the performance, thus leaving only the number of returned results as significant factor affecting the query time. The dependency of reasoning performance from the size of knowledge model and query performance from the number of returned results could be found in Figs. 7 and 8 correspondingly.

According to Fig. 7, the reasoning time increases significantly with the number of individuals in the ontology; however, the reasoning process is done only several times during configuration, thus the reasoning time does not affect dramatically the total configuration time of the system. Moreover, as it is seen from Fig. 8, the query time does not depend on whether the querying is done over inferenced model or over the model without inferencing (black and beige bars of Fig. 8 give similar results), which even

[15] ZeroMQ performance tests: http://zeromq.org/area:results.



(a)

Hello, Jhon!
Your role: maintenance

02.02.2016
12:40:28

timeStamp	location	devType	value	streamId
02.02.2016 12:40:28	B1/A1	TemperatureSensors	21.0 C	TS1S1
02.02.2016 12:40:25	B1/A2	TemperatureSensors	21.0 C	TS2S1
02.02.2016 12:40:24	B2/A3	TemperatureSensors	21.8 C	TS3S1
02.02.2016 12:40:28	B2/A4	TemperatureSensors	21.1 C	TS4S1

Showing 1 to 4 of 4 entries

(b)

Hello, Jhon!
Your role: maintenance

02.02.2016
12:43:54

timeStamp	location	devType	value	streamId
02.02.2016 12:43:36	B1/A1	TemperatureSensors	20.4 C	TS1S1
02.02.2016 12:43:53	B1/A2	TemperatureSensors	20.1 C	TS2S1
02.02.2016 12:43:51	B2/A3	TemperatureSensors	20.4 C	TS3S1
02.02.2016 12:43:47	B2/A4	TemperatureSensors	20.0 C	TS4S1
02.02.2016 12:43:43	B3/A5	TemperatureSensors	20.7 C	TS5S1

Fig. 6. Changes in the delivery of information streams after reconfiguration of knowledge model: (a) user *Jhon* has role *VM1* with responsibility to monitor class of *TemperatureSensors* and (b) new *TemperatureSensors TS5* was added and automatically updated for user *Jhon*.

more reduces the significance of time of the reasoning for the proposed middleware. Regarding the querying time of Fig. 8, the system configuration process is designed in a way, that it does not require repeated processing of large query results. Usually, the size of result of configuration queries is no more than 100 rows with query times within 10 ms.

As it was mentioned earlier, the resulting objects of the system are configured by combination of several queries. For described scenario, each user role is configured to monitor various numbers of locations, which vary during experiments. Thus, the configuration time of each user depends on the number of locations, which he/she is supposed to monitor and is increasing with number of locations and devices in the system (Fig. 9). The device configuration includes queries of device type and device locations and does not depend on the total number of devices in the system. Thus, configuration time of one device remains on the same level, around 15 ms, regardless of total number of devices (Fig. 9). The total configuration time of the system mostly depends on total number of devices, and takes up to few minutes in case of 10,000 devices (i.e. around 15 ms multiplied by total number of devices). However, for real case scenarios it is expected that only limited number of devices would require reconfiguration at once, thus this approach remains feasible for configuration of large scale systems.

7.2. Runtime data processing

The considered *data processing time* is the time required by the system to read update data from device and to send update notifications to selected users after CEP processing. It consists of two measured intervals: (1) the time from reading of update status of device in the Information Source Adapters application to receiving it on Event Processor side (or *update reading time*) and (2) the time required by CEP to process new information (or *CEP processing time*). During our experiments, the triggering of update notifications to all users of the system occurred within 1–2 ms; thus, the performance measurements are reported for one user.

There are two possible ways of implementing the procedure of rule check with Drools CEP engine in the Event Processor application. One way is to check rules conditions on each incoming update from devices. This option guarantees that each update message will be evaluated with CEP rules before any other updates are permitted for CEP evaluation. This approach works fast, when number of devices and updates is quite low; it has around 50 ms of *data processing time*, when update rate from all devices is about 300 updates per second or 3000 devices in the presented scenario (black bar in Fig. 10). In order to process update from each device, CEP engine requires 6 ms for each update (*CEP processing time*). When number of devices and update rate grow, this delay becomes significant, resulting in accumulation of update messages on Event Processor side and increasing dramatically *update reading time*. Thus, the time of update for 10,000 devices with the total update rate 1000 per second increases to 3.5 s (black bar in Fig. 10).

The other way of CEP implementation is to periodically check rules in the separate thread. This approach improves *update reading time*, because receiving site of Event Processor only updates facts for the CEP session without triggering rules check, and thus without having additional 6 ms delay. For given scenario, *update reading time* was within 1–2 ms regardless of number of devices and update rates. The processing time of second approach consists of *CEP processing time*, when more facts have to be evaluated at once. In general, this approach scales better and has *data processing time* of around 1 s at a maximum number of considered update rate and number of devices (beige bar in Fig. 10). However, for device updates of higher frequency, there is a chance of missing some of the readings, when device updates happen more frequently than rules evaluation. For example, according to Fig. 10, if system produces 900 updates from devices per second, all device updates will be considered only if update rate from each device is not more frequent than once per second. If it is crucial for application to evaluate each device update, it should be taken into account. In the lower update frequencies, the performance depends on the pause interval between rules check. The beige bar of Fig. 10 demonstrates this performance with the pause of 100 ms (i.e. the *data processing time* consist of *update reading time*, *CEP processing time*, and 100 ms sleep), which might be improved by reducing sleep interval between rules evaluation.

For considered scenario, when each device has average update rate once per ten second (or in total 1000 updates per second from all devices, which come at different times), the *data processing time* of the second CEP implementation is around one second at maximum update rates, which means that each device update will be considered in CEP and all updates will be delivered to users within one second.

When transferring from simulated environment to a real case system, it is expected that additional delays in the communication could be introduced in the system. The delays might be most prominent in communication between Event Processor and Ontology Service (Fig. 3) due to their usage of TCP/IP protocols for web services communication. However, these applications communicate only at initial configuration phase making the configuration delays not significant for general performance. The updates of the devices arrive through ZeroMQ socket library, which is

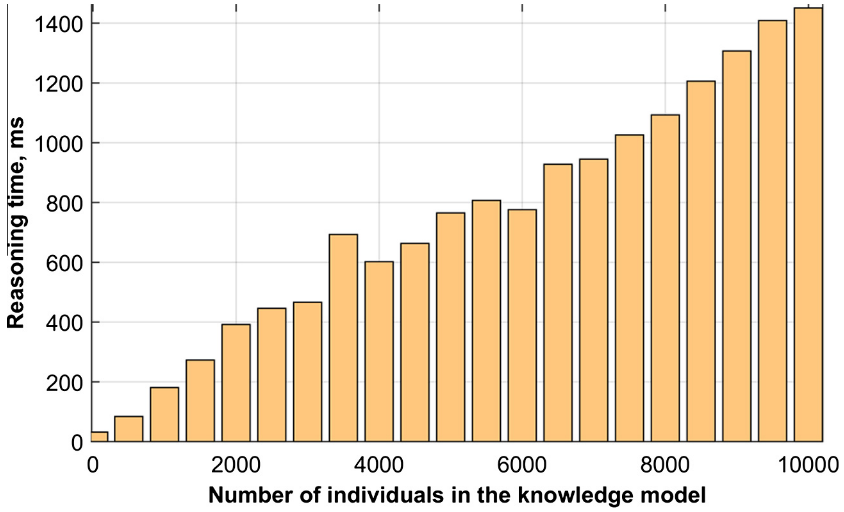


Fig. 7. Reasoning time of Ontology Service depending on size of knowledge model.

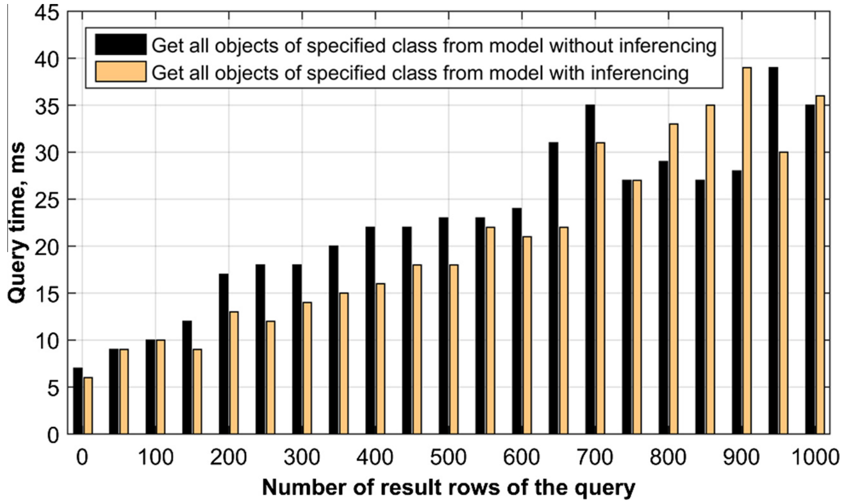


Fig. 8. Query time of Ontology Service depending on number of result rows.

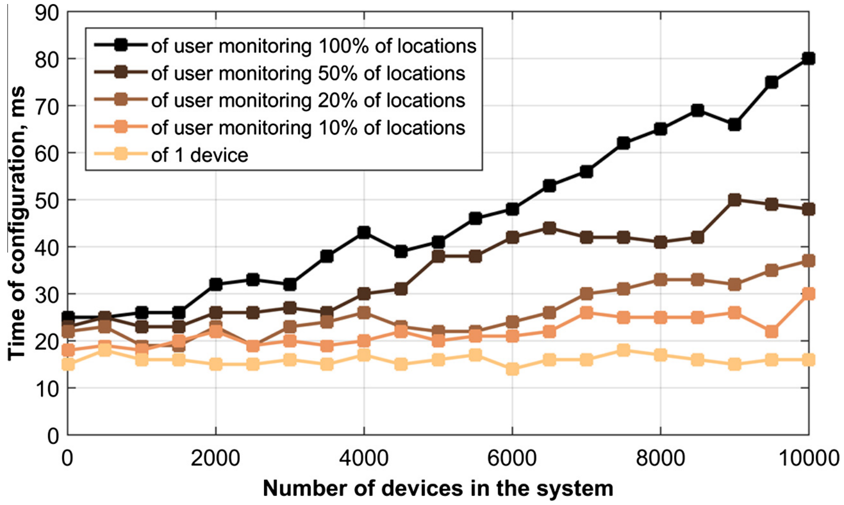


Fig. 9. Configuration times of various objects in the system depending on total number of devices.

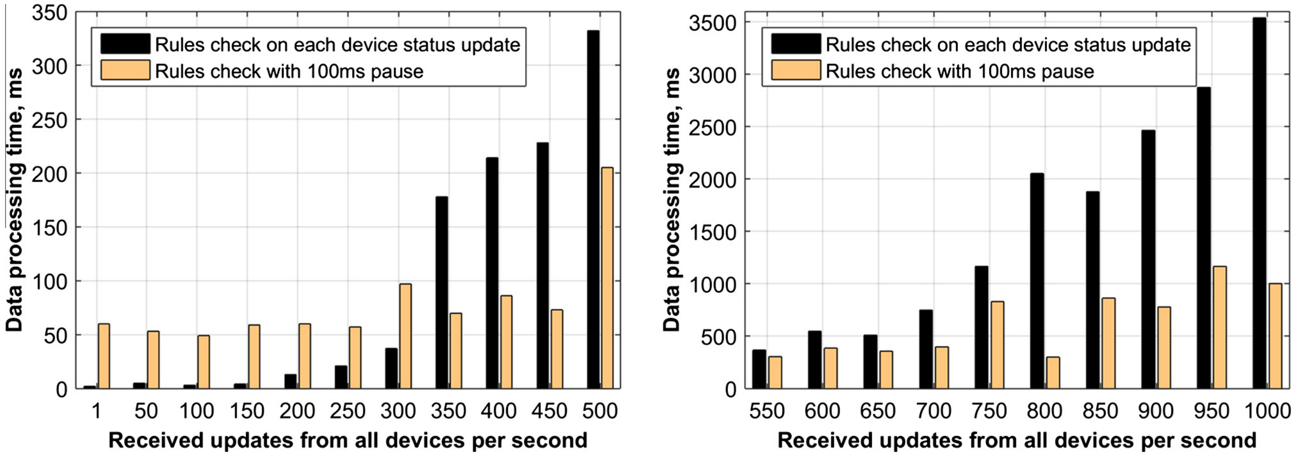


Fig. 10. Runtime data processing depending on number of updates received from all devices per second (for better visualization the figure is split in two with different scale for y-axis).

designed as high throughput communication library with small communication delays. Thus, the demonstrated performance tests suggest feasibility of the approach for monitoring systems, where the response time within a few seconds is tolerated.

8. Conclusion and future work

This paper deals with selective delivery of information streams in data-intensive monitoring systems based on user role and responsibilities. The proposed approach combines ontological model for reconfiguration purposes with CEP engine for run-time analyzing. The paper has presented in detail principles of hybrid approach with illustrative scenarios from building automation domain, and has demonstrated feasibility of used technologies and implemented tools through set of performance tests.

The main advantages of the system are twofold: (1) utilizing CEP engine at system run-time enables event-driven monitoring and update notifications in comparison to pure SW approaches and (2) modeling system with human-machine understandable ontologies ensures easier reconfiguration of the system, when only knowledge model should be changed for altering of system behavior instead of reconfiguring native CEP rules each time new configuration has to be introduced in the system.

The future work will be directed to an extension of approach by: (1) analyzing of more factors, which affect information relevancy in monitoring systems, such as activities and tasks of the user, and state of the user and (2) situation tracking and selective delivery of notifications and alerts based on information from environment and information relevancy for particular user.

Acknowledgements

This work was supported by the Doctoral Programme of the President of the Tampere University of Technology.

References

[1] P.H. Seong (Ed.), Reliability and Risk Issues in Large Scale Safety-Critical Digital Control Systems, Springer, 2009.
[2] Improve Operations Performance, Emerson Process Manag. (2016) 1–16.
[3] K. Larson, Operator effectiveness: the next frontier of process automation, ABB (2014) 1–4.
[4] BAS delivers a new kind of smart, Johnson Control. (2015) 1–6.
[5] J. Sinpoli, Don't be panicked by alarm management of building systems, AutomatedBuildings.com (2011).
[6] T. Sauter, S. Soucek, W. Kastner, D. Dietrich, The evolution of factory and building automation, IEEE Ind. Electron. Mag. 5 (2011) 35–48.
[7] G. Fischer, Context-aware systems, in: Proc. Int. Work. Conf. Adv. Vis. Interfaces – AVI '12, ACM Press, New York, USA, 2012, pp. 287–294.
[8] A. Esposito, L. Tarricone, M. Zappatore, L. Catarinucci, R. Colella, A. Dibari, A framework for context-aware home-health monitoring, Ubiquitous Intell. Comput. 3 (2008) 75–91.
[9] J. Puttonen, A. Lobov, M.A. Cavia Soto, J.L. Martinez Lastra, Planning-based Semantic Web service composition in factory automation, Adv. Eng. Informatics 29 (2015) 1041–1054.
[10] J. Puttonen, A. Lobov, M.A. Soto Cavia, J.L. Martinez Lastra, A Semantic Web services-based approach for production systems control, Adv. Eng. Informatics 24 (2010) 285–299.
[11] X. Xue, J. Chang, Z. Liu, Context-aware intelligent service system for coal mine industry, Comput. Ind. 65 (2014) 291–305.
[12] M.K. Uddin, J. Puttonen, S. Scholze, A. Dvoryanchikova, J.L.M. Lastra, Ontology-based context-sensitive computing for FMS optimization, Assem. Autom. 32 (2012) 163–174.
[13] N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, W. Schwinger, BeAware! – situation awareness, the ontology-driven way, Data Knowl. Eng. 69 (2010) 1181–1193.
[14] M. Ruta, F. Scioscia, G. Loseto, E. Di Sciascio, Semantic-based resource discovery and orchestration in Home and Building Automation: a multi-agent approach, IEEE Trans. Ind. Informatics 10 (2014) 730–741.
[15] J. Rodriguez, M. Bravo, R. Guzman, Multi-dimensional ontology model to support context-aware systems, in: Int. Conf. Internet Web Appl. Serv. (2012) 80–89.
[16] A. Furno, E. Zimeo, Context-aware composition of Semantic Web Services, Mob. Networks Appl. 19 (2014) 235–248.
[17] F. Smith, D. Bianchini, Selection, ranking and composition of semantically enriched business processes, Comput. Ind. 65 (2014) 1253–1263.
[18] Y. Evchina, J. Puttonen, A. Dvoryanchikova, J.L. Martinez Lastra, Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems, Eng. Appl. Artif. Intell. 43 (2015) 111–126.
[19] M.J. O'Connor, A.K. Das, A method for representing and querying temporal information in OWL, Biomed. Eng. Syst. Technol. (2011) 97–110.
[20] G. Okeyo, L. Chen, H. Wang, Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes, Futur. Gener. Comput. Syst. 39 (2014) 29–43.
[21] C. Seeger, K. Van Laerhoven, J. Sauer, A. Buchmann, A. Publish, A Publish/Subscribe middleware for body and ambient sensor networks that mediates between sensors and applications, 2013 IEEE Int. Conf. Healthc. Informatics, IEEE (2013) 199–208.
[22] G. Russello, L. Mostarda, N. Dulay, A policy-based publish/subscribe middleware for sense-and-react applications, J. Syst. Softw. 84 (2011) 638–654.
[23] K. Shi, Z. Deng, X. Qin, TinyMQ: a content-based publish/subscribe middleware for Wireless Sensor Networks, Sens. Technol. Appl. (2011) 12–17.
[24] J. Arias Fisteus, N. Fernández García, L. Sánchez Fernández, D. Fuentes-Lorenzo, Ztreamy: a middleware for publishing semantic streams on the Web, Web Semant. Sci. Serv. Agents World Wide Web. 25 (2014) 16–23.
[25] G. Mühl, H. Parzyjegla, M. Prellwitz, Analyzing content-based publish/subscribe systems, in: 9th ACM Int. Conf. Distrib. Event-Based Syst., ACM Press, New York, New York, USA, 2015, pp. 128–139.
[26] C. Cañas, E. Pacheco, B. Kemme, J. Kienzle, H.-A. Jacobsen, GraPS: a graph publish/sybscribe middleware, in: 16th Annu. Middlew. Conf., ACM Press, New York, New York, USA, 2015, pp. 1–12.
[27] D. Luckham, The power of events: an introduction to Complex Event Processing in distributed enterprise systems, in: Rule Represent. Interchang. Reason. Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, p. 3.
[28] I. Zappia, F. Paganelli, D. Parlanti, A lightweight and extensible Complex Event Processing system for sense and respond applications, Expert Syst. Appl. 39 (2012) 10408–10419.

[29] X. Zhao, C. Liu, T. Lin, Incorporating business logics into RFID-enabled applications, Inf. Process. Manage. 48 (2012) 47–62.

[30] W. Yao, C.-H. Chu, Z. Li, Leveraging complex event processing for smart hospitals using RFID, J. Netw. Comput. Appl. 34 (2011) 799–810.

[31] F. Terroso-Sáenz, M. Valdés-Vela, F. Campuzano, J.A. Botia, A.F. Skarmeta-Gómez, A complex event processing approach to perceive the vehicular context, Inf. Fusion. 21 (2015) 187–209.

[32] J. Weiss, P. Mandl, A. Schill, Functional testing of Complex Event Processing applications, 2011 IEEE 7th Int. Conf. Intell. Comput. Commun. Process., IEEE (2011) 407–413.

[33] M. Eckert, F. Bry, S. Brodt, O. Poppe, S. Hausmann, A CEP babelfish: languages for complex event processing and querying surveyed, Reason. Event-Based Distrib. Syst. (2011) 47–70.

[34] J. Boubeta-Puig, G. Ortiz, I. Medina-Bulo, A model-driven approach for facilitating user-friendly design of complex event patterns, Expert Syst. Appl. 41 (2014) 445–456.

[35] D. Anicic, P. Fodor, S. Rudolph, N. Stojanovic, EP-SPARQL: a unified language for event processing and stream reasoning, in: Proc. 20th Int. Conf. World Wide Web – WWW '11, 2011, pp. 635–644.

[36] O. Saleh, S. Hagedorn, K.-U. Sattler, Complex event processing on linked stream data, Datenbank-Spektrum 15 (2015) 119–129.

[37] S. Komazec, D. Cerri, D. Fensel, Sparkwave, in: Proc. 6th ACM Int. Conf. Distrib. Event-Based Syst. – DEBS '12, ACM Press, New York, New York, USA, 2012, pp. 58–68.

[38] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, Linking building data in the cloud: integrating cross-domain building data using linked data, Adv. Eng. Informatics 27 (2013) 206–219.

[39] D. Le-Phuoc, H.Q. Nguyen-Mau, J.X. Parreira, M. Hauswirth, A middleware framework for scalable management of linked streams, Web Semant, Sci. Serv. Agents World Wide Web. 16 (2012) 42–51.

[40] A. Margara, J. Urbani, F. Van Harmelen, H. Bal, Streaming the web: reasoning over dynamic data, J. Web Semant. 25 (2014) 24–44.

[41] M. RobatJazi, M.Z. Reformat, W. Pedrycz, Ontology-based framework for reasoning with fuzzy temporal data, 2012 IEEE Int. Conf. Syst. Man, Cybern., IEEE, Seoul (2012) 2030–2035.

[42] G. Fenza, D. Furno, V. Loia, Hybrid approach for context-aware service discovery in healthcare domain, J. Comput. Syst. Sci. 78 (2012) 1232–1247.

[43] N. Díaz-Rodríguez, O.L. Cadahía, M.P. Cuéllar, J. Lilius, M.D. Calvo-Flores, Handling real-world context awareness, uncertainty and vagueness in real-time human activity tracking and recognition with a fuzzy ontology-based hybrid method, Sensors 14 (2014) 18131–18171.

[44] J. Zhang, Y. Wang, J. Vassileva, SocConnect: a personalized social network aggregator and recommender, Inf. Process. Manage. 49 (2013) 721–737.

[45] K. Taylor, L. Leidinger, Ontology-driven complex event processing in heterogeneous sensor networks, Semant. Web Res. Appl. (2011) 285–299.

[46] A. Gomez-Perez, M. Fernandez-Lopez, O. Corcho, Ontological Engineering: With Examples From the Areas of Knowledge Management, E-Commerce and the Semantic Web, Springer, 2010.

[47] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, et al., The SSN ontology of the W3C semantic sensor network incubator group, Web Semant, Sci. Serv. Agents World Wide Web. 17 (2012) 25–32.

[48] S. Kolozali, M. Bermudez-Edo, D. Puschmann, F. Ganz, P. Barnaghi, A knowledge-based approach for real-time IoT data stream annotation and processing, 2014 IEEE Int. Conf. Internet Things(iThings), IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput., IEEE (2014) 215–222.

[49] Y.-B. Kang, Y.-F. Li, S. Krishnaswamy, Predicting reasoning performance using ontology metrics, Semant. Web – ISWC 2012 (2012) 198–214.

[50] I. Horrocks, L. Li, D. Turi, S. Bechhofer, The instance store: DL reasoning with large numbers of individuals, Descr. Log. Work. (2004) 31–40.

Publication V

Evchina, Y., & Martinez Lastra, J. L. (2018). An approach to combining related notifications in large-scale building management systems with a rehabilitation facility case study. *Automation in Construction*, *87*, 106–116. https://doi.org/10.1016/j.autcon.2017.12.020

The original print of this publication is available online (DOI: 10.1016/j.autcon.2017.12.020).

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

An approach to combining related notifications in large-scale building management systems with a rehabilitation facility case study

Yulia Evchina[*], Jose Luis Martinez Lastra

FAST Laboratory, Tampere University of Technology, P.O. Box 600, FIN-33101 Tampere, Finland

ARTICLE INFO

Keywords:
Building management systems
Alarm management
Alarm classification
Alarm design
Notification management

ABSTRACT

ICT advances have enabled the incorporation of multiple devices that monitor various aspects of the environment into building management (BM) systems. The data from these devices is used to detect multiple abnormal situations, which require the awareness of system users and/or timely response. However, the number of abnormal situations is usually large, and delivering all of the associated notifications is overwhelming for users, rather than helping them to interpret the ongoing status of the environment. This work proposes a novel approach for combining ongoing notifications in the monitoring systems by their types, priorities, locations, and receivers. The approach is based on formal classification of possible alarms and runtime analysis of ongoing notifications with the aim of reducing repeating information pieces delivered as part of multiple notifications. The paper provides details of combination principles of notifications and applies them to real data from a rehabilitation facility. The results show a reduction in the users' information load of approximately 42% of the peak number of ongoing notifications. It is expected that the proposed approach will improve situation awareness in the managed facilities – enabling better and faster decisions on the ongoing status of the environment.

1. Introduction

Present day control and monitoring systems can be equipped with a wide range of heterogeneous devices capable of monitoring various aspects of the environment, the process of interest, and even the health and mental state of system users. This fact is especially evident in building management (BM) systems, where advancements in information and communications technology (ICT) have enabled the first steps toward integration of various building subsystems which have dramatically increased BM system capabilities along with the diversity of devices [1,2]. The number and diversity of devices, in turn, allow detection of multiple abnormal situations, which should, in principle, improve situation awareness and help the users to make correct and timely decisions. However, the number of notifications associated with abnormal situations is usually large and instead of helping the users to interpret the ongoing status of the environment, this can overwhelm a user's cognitive capacities and create situations where important information is unnoticeable or even lost. The result of this intensive information flow is reduced system safety due to an increased number of user errors. This fact is more studied in the domain of process industry, where downtimes and safety issues are costly and require close attention. For example, many sources report a range of 30–90% of all system failures as operator errors [3–5], while stating that the cost of such events is around 3–8% of plant capacity [6]. This problem is also relevant in other domains of control and monitoring systems, such as building management, where the number of data points can be in the tens of thousands, and facility operators must be assisted in order to easily locate the problem and provide a timely response to it [7–9]. There is therefore a great need to improve the situation awareness of the users of these systems by reducing the information load delivered to system users in the form of notifications.

The research community has addressed the problem of intensive information flows from several perspectives. One of the approaches is to enable delivery of the right information at the right time to the right person through context-aware systems [10]. The general idea behind context-aware systems is the introduction of an additional layer in the system architecture that is responsible for interpreting ongoing situations and inferring the most appropriate information for each user of the system in the ongoing context [11–13]. The concept of context-awareness is adopted in various domains ranging from industrial applications to building management and healthcare monitoring systems with the focus to either 1) appropriately react to the inferred situations of the environment, as in [11,14,15], or 2) conveniently present information to the user, as in [16–19]. While context-aware systems deal

[*] Corresponding author.
E-mail addresses: yulia.evchina@tut.fi (Y. Evchina), jose.lastra@tut.fi (J.L. Martinez Lastra).

https://doi.org/10.1016/j.autcon.2017.12.020
Received 2 August 2017; Received in revised form 3 November 2017; Accepted 9 December 2017

with the delivery of the correct information and improvement of user awareness of what is happening in the surrounding environment, they are not originally concerned with reducing information intensity to the user. A set of techniques exists, which are closely related to this topic, called alarm management. Alarm management is more commonly known and more formally described in process industry; it comprises the set of techniques, tools, standards, and procedures, with the goal of improving the effectiveness of alarm systems [20]. Recent research has proposed new algorithms for reducing information load, which comes in the form of nuisance and false alarms [21–23], number of alarm floods [24,25], wrong prioritization of alarms [26], and presenting large amounts of similar data to the users [27]. These works are usually based on analysis of alarm logs and related data, which results in detection of badly designed alarms and recommendations for reconfiguring alarms in the control and monitoring system.

This work proposes an approach for combining related notifications in the BM systems with the aim of reducing information load to the users. The approach covers the design and runtime phases of notification management. The design phase enables clear classification and correct priority assignment between alarms. The runtime phase implements combination principles to reduce information load to the users at system runtime. The combination principles relate ongoing notifications by type, priority, location, and notification receiver. The main objective is to reduce the number of notifications that have repeated information pieces. The runtime notification analysis is a major advantage of this approach in comparison to most of the existing solutions, which use log data for alarm analysis and are not able to address user needs during system operation.

The rest of the paper is organized as follows. Section 2 presents background and related work, with the focus on alarm management techniques and practices. Section 3 contains detailed description of the proposed principles for defining related notifications in the monitoring systems, with illustrative examples from the domain of building management. Section 4 gives details of the tools implemented to support the proposed principles of combining related notifications and describes the use case chosen for study. Section 5 presents the results of applying the proposed principles to the case study of a rehabilitation facility. Finally, Section 6 concludes the paper and gives directions for future work.

2. Background and related works

2.1. Notification types in monitoring systems

Monitoring systems produce notifications of various types and priorities. The difference between notification types depends on the domain of the particular monitoring system. For example, process industry is a highly automated domain with systems having a large number of process variables to control and monitor. The safety of such systems, however, critically depends on the operator's proficiency and his/her responses during abnormal situations. This domain is therefore well supported by alarm management standards and recommendations including EEMUA-191 (2007) Alarm Systems: A Guide to Design, Management, and Procurement, ANSI/ISA-18.2 (2009) Management of Alarm Systems for the Process Industries and its IEC 62682:2014 extension [28,29]. The ANSI/ISA-18.2 (2009) clearly defines four types of notifications: alarms, alerts, prompts, and messages. The alarms and alerts are used to inform about abnormal situations. The difference between them is in the required actions for the operator: alarm requires the response of the operator (action to improve situation), while alert requires only operator awareness of the abnormal condition (e.g., to pay closer attention, to check up other variables) [30].

Building management is less critical in a sense of downtime costs and incident consequences, therefore this domain is less supported by standards and recommendations [9]. The nature of abnormal situations is also quite different. In the process industry, everything is connected to the automated process, with numerous parameters to control and monitor; in building management, abnormal situations are quite independent in the sense that they usually happen in self-substantive units, such as apartments and buildings. The responses to abnormal situations also have differences: in the process industry, the response is supposed to be a well-defined process of actions in response to the ongoing condition whereas in building management, the responses are less procedural and the operators usually follow short instructions on how to react to certain types of high-priority event. In addition, the operators of such systems are the users, who have other roles and are often only required to have an awareness of what is happening in the environment, rather than to immediately act upon a situation. The difference between alarms and alerts in BM systems is thereby less critical in comparison to process industries; therefore, this paper does not distinguish between them – all notifications indicating abnormal conditions are referred to as alarms. This generalization does not significantly alter the meaning of alarms from process industry; however, it will enable the consideration of all possible abnormal situations and their related notifications in building management systems.

2.2. Alarm management approaches

Abnormal situations in the large-scale monitoring systems create intensive information flows. This problem is relevant in many domains of control and monitoring systems, but it is most studied in the process industry, where it is addressed through a series of practices called alarm management. Alarm management practices aim to increase the effectiveness of alarm systems in a plant; they can include a set of procedures ranging from identification and elimination of badly designed alarms to operator training on how to respond to alarms [31–33].

The research community has proposed numerous ways to improve operator effectiveness in relation to different aspects of alarm management in process industries. One of these approaches is to improve visualization tools, which aid the detection of badly designed alarms causing alarm floods (the condition where the rate of alarms is higher than the rate of the operator's comprehension of them [34]) and/or alarm chatter (the condition of certain alarms in which they activate and deactivate excessively [21]). For example, high-density alarm plot (HDAP) charts, alarm similarity color map (ASCM), and an index of alarm chatter are proposed in [21,34] with the following experimental studies showing their effectiveness in the detection of badly designed alarms [35]. Another way is to use plant data to predict equipment failures and improve operation set points, thus reducing the number of plant shutdowns and alarms [36]. While this is a promising approach for improving the work of plants, it usually requires high computational power to enable this approach at near real time. Finally, there are a number of works that attempt to analyze the similarity of alarms and their causes to offer a potential reduction in information load to the operators of the control and monitoring systems. Examples of such works include the proposal of new algorithms for detection of similar alarms [22,25], sequences of alarms [37], correlated alarms [38], and the same causes of alarms [39]. The common approach in these works is 1) to analyze available alarm log data and/or related information such as plant and system topology, 2) to detect problems in control and monitoring systems from an alarm management point of view, and, 3) to produce suggestions for improving alarm polices in a particular system.

While identification and possible elimination of badly designed alarms based on alarm logs is an important practice for improving alarm systems—and, thereby, user awareness—the related notifications in building management systems are not always a problem of alarm design, but rather a consequence of the possibility of having multiple, seemingly unconnected, triggers at the same time. For example, the warnings about an opened entrance door and low indoor temperature are two separate types of notifications, which can happen independently in different locations; however, when two of these conditions are met in one location, they might be related, due to the cold weather outside, rather than indicating the malfunctioning of the heating system. This example shows the importance of analyzing notifications and their possible relations at the time of occurrence—the focus of this work. The proposed approach is able to relate ongoing notifications based on their types, priorities, locations, and receivers, thereby reducing information load to the user and improving his/her situation awareness of what is happening in the environment.

The novelty of this work in comparison to the previous research lies in tackling particular problems and needs of BM systems in alarm management. Alarms and notifications of BM systems are connected to ever-changing environment conditions in multiple apartments, rather than having complex interdependencies as in industrial processes. The relation of notifications in BM systems is more intuitive, can be well formalized, and is readily implementable. Rather than using the bottom-up approach of analyzing alarm log data and finding alarm dependencies, this work proposes a top-down approach of formal classification of notifications and alarms with further runtime analysis of ongoing notifications.

3. Alarm combination principles

3.1. Classification of existing alarms

Before proceeding to analyze ongoing notifications to detect abnormal situations, we need to define several principles of classification for potential alarms in the system. This subsection formally introduces these principles with illustrative examples from the domain of building management.

The abnormal situations are detected if certain conditions in the system are met. In this paper, we refer to those conditions as notification conditions. Notification conditions typically consist of the description of what is required to cause this notification and the related notification parameters. Formally, it could be described as a set containing the following elements:

$$\mathbf{N_C}: = \{o_C, \boldsymbol{D_C}, ts_C, t_{l_C}, \boldsymbol{R_C}, t_C, p_C, \boldsymbol{HC_C}\} \tag{1}$$

where o_C is an informal description of the outcome, when the condition is fulfilled (e.g. "fire alarm", "nurse call"); $\mathbf{D_C}$ is the set of device condition elements describing condition fulfillment; ts_C is the timespan of the notification condition (i.e. the time during which the device condition elements should remain in order for the notification to become active); t_{l_C} is the location type (e.g. "Buildings", "Apartments", "Kitchen", "Dining Area"), which defines the scope of the nearby sensors; $\mathbf{R_C}$ is the set of characteristics of receivers of this notification, which describe who should receive it (e.g. user roles, locations); t_C is the type of the resulting notification; p_C is the priority of the resulting notification; and $\mathbf{HC_C}$ is the set of notifications of higher class, which will be described later in this section.

The device condition elements of the $\mathbf{D_C}$ set consist of the set of device conditions in the form:

$$\mathbf{d_C}: = \{t_{d_C}, c\} \tag{2}$$

where t_{d_c} is the type of the condition device; and c is the description of the device output value condition that causes the notification (e.g. "$v < 10$"). If $\mathbf{D_C}$ consists of several elements, then the notification is triggered only when there are a set of devices within the same location type t_{l_c}, which hold their values within specified conditions for at least the duration indicated by ts_C.

For better illustration of these notations, let us consider a simple example in this section. Let the hypothetical facility be an apartment building that has residents, nurses, and maintenance personnel on the site. Each apartment has a simple set of sensors: temperature sensor, door state sensor, and a smoke detector, and one common outside temperature sensor. Assuming that each apartment has a separate entrance from the street, the initial table of notifications could be written as demonstrated in Table 1.

As can already be seen, even with this simple set of sensors there are many options to track various situations. The notification conditions are different, but many of them use the same set of sensors, which might generate multiple notifications with similar information. The first step toward reducing repeat information is to classify possible notifications and define notification types (t_C), priorities (p_C), and higher classes ($\mathbf{HC_C}$) for each of them.

3.1.1. Assigning notification types

The type of a notification should follow directly from the description of its outcome. If the detection of a set of different notification conditions lead to the same outcome, then these notifications should be of the same type, even if the devices are different. More formally, it could be represented in the simple expression:

$$\forall\ \mathbf{N_{C1}}, \mathbf{N_{C2}}: o_{C1} = o_{C2} \rightarrow t_{C1} = t_{C2} \tag{3}$$

According to this expression, the example notifications of Table 1 could have the following types: 1 – FireAlarm, 2 – FireAlarm, 3 – HighIndoorTemperature, 4 – LowIndoorTemperature, 5 – OpenedDoor, 6 – OpenedDoor, 7 – OpenedDoorInColdWeather. Thus, for an indoor temperature sensor there are two different alarm types for high temperatures (HighIndoorTemperature and FireAlarm) rather than the same type with different priorities; while two different sensors capture the same notification type FireAlarm. The proposed classification rule is simple, but it allows notification types to focus on the meaning of the detected condition rather than the underlying devices supporting its detection.

3.1.2. Assigning notification priorities

The priorities show the importance of alarms and the degree of urgency, which must be considered by the users of the system. In the process industries, there are usually three to five degrees of importance, in line with the ISA 18.2 recommendation of using no more than four alarm priorities. For different alarm types, the assignment of priority is usually a long iterative task, involving analysis of each potential alarm, possibly with alignment to a safety risk matrix for each case (i.e. likelihood and severity of incidents). If no safety risk matrix is available, then a consistent policy of priority assignment should be adopted. As an initial approach, the following four-degree priority classification in the BM domain could be adopted: priority 0 – diagnostic messages (information, no threat to property or people), priority 1 – warnings (potentially leading to property damage), priority 2 – high-priority messages (potentially leading to personal injury), priority 3 – emergency (situations threatening the wellbeing of people).

While different notification types require careful consideration for their assigned priorities, within one notification type two simple indicators of priority relationships are notification timespan (ts_C) and the relation between devices. The formal description of this case is:

$$\forall\ \mathbf{N_{C1}}, \mathbf{N_{C2}}: t_{C1} = t_{C2}, \begin{cases} \forall\ \mathbf{d_{Ci}} \in \mathbf{D_{C2}},\ \exists\ \{\mathbf{d_{Cj}} \in \mathbf{D_{C1}} \mid t_{d_{Ci}} = t_{d_{Cj}}, c_j \subset c_i\} \\ \mathbf{D_{C1}} = \mathbf{D_{C2}}, ts_{C1} > ts_{C2} \end{cases}$$
$$\rightarrow p_{C1} > p_{C2} \tag{4}$$

or in other words, if there are two notification conditions of the same type along with either of the additional conditions that follow, then the priority of the first notification should be higher than the priority of the second notification. The conditions are: 1) the device condition values of the first notification are a subset of the device condition values of the second notification; or 2) the device conditions are the same for both notifications, and the timespan of the first condition is larger than the timespan of the second condition.

This rule regulates the severity principle for the same notification types and suggests higher priorities for notifications that include at least the same set of devices with more "severe" values ($c_j \subset c_i$ condition) or which have longer timespan. When applied to our example, we have two pairs of notifications of the same type (1, 2, and 5, 6). In the case of

Table 1
Initial description of notifications in the system.

N	Description $\boldsymbol{o_C}$	Sensors $\mathbf{D_C}$		Duration to detect $\boldsymbol{ts_C}$	Scope of detection $\boldsymbol{t_{l_c}}$	Receivers $\mathbf{R_C}$
		Device types t_{d_c}	Values c			
1	Detection of fire	Smoke detector	true	0.5 min	Apartments, Buildings	Fire department, residents, nurses
2	Detection of fire	Indoor temperature sensor	> 40 °C	1 min	Apartments	Fire department, residents, nurses
3	High indoor temperature	Indoor temperature sensor	> 30 °C	10 min	Apartments	Resident
4	Low indoor temperature	Indoor temperature sensor	< 19 °C	30 min	Apartments	Resident, maintenance
5	Opened door	Door state sensor	True	20 min	Apartments	Resident
6	Opened door	Door state sensor	True	60 min	Apartments	Resident, maintenance
7	Opened door in cold weather	Door state sensor Indoor temperature sensor Outdoor temperature sensor	True < 20 °C < 10 °C	10 min	Apartments	Resident, maintenance

FireAlarm, the notifications are caused by different devices and do not have common intersection, thus these two notifications should be of the same priority. In the case of OpenedDoor, the timespan of the second notification is larger than that of the first, so notification 6 should have higher priority than notification 5. In general, this approach reflects the increase in severity due to the prolonging of the situation or the worsening of the device values. In addition, such classification has important consequence of including all related device values into higher-priority notifications, which is used later on in combination principles.

3.1.3. Assigning higher classes

So far, we have assigned notification types, priorities, and relationships between priorities of the same type. However, some of the remaining notification types still seem to be related as a representation of more complex situations. The notion of higher classes reflects this relationship and takes into consideration the underlying device values and the combination of different notification conditions. This relationship could be represented as following:

$$\forall\, \mathbf{N_{C1}}, \mathbf{N_{C2}}\colon t_{C1} \neq t_{C2}, \left\{ \begin{array}{l} \boldsymbol{D_{C2}} \subset \boldsymbol{D_{C1}},\ \forall\, \boldsymbol{d_{Ci}} \in \boldsymbol{D_{C2}},\ \exists\ \{\boldsymbol{d_{Cj}} \in \boldsymbol{D_{C1}} \mid t_{d_{Ci}} \\ \quad = t_{d_{Cj}},\ c_i \cap c_j \neq \varnothing\} \\ \boldsymbol{D_{C2}} = \boldsymbol{D_{C1}},\quad \forall\, \boldsymbol{d_{Ci}} \in \boldsymbol{D_{C2}},\ \exists\ \{\boldsymbol{d_{Cj}} \in \boldsymbol{D_{C1}} \mid t_{d_{Ci}} \\ \quad = t_{d_{Cj}},\ c_j \subset c_i\} \end{array} \right\} \rightarrow t_{C1} \in \mathbf{HC_{C2}} \tag{5}$$

or in other words, if there are two notification conditions of different type along with either of the following conditions being present, then the type of the first notification should be included as a higher class for the second notification. The conditions are: 1) all of the devices of the second notification are a subset of devices of the first notification and the values of these devices intersect; or 2) the underlying devices are the same and the values of the first notification condition are a subset of the values of the second notification condition.

An illustrative example of such a case is the relationship between three notification conditions: the opened door, low indoor temperature, and opened door in cold weather. The OpenedDoor and LowIndoorTemperature notifications are two separate notifications that include the door sensor and indoor temperature sensor, respectively, while detection of OpenedDoorInColdWeather requires usage of the door state sensor, indoor temperature sensor, and outside temperature sensor, indicating the influence of the outside temperature. Since the device conditions of the latter notification intersect with those of the first two notifications, the opened door in cold weather condition acts as a notification of the higher class for both OpenedDoor and LowIndoorTemperature. In case of the high indoor temperature notification, where the same temperature sensor is used, the ranges of the notification triggering conditions do not intersect and, therefore, the OpenedDoorInColdWeather notification condition cannot be indicated as a higher-class notification for the HighIndoorTemperature condition. Again, the consequence of this classification principle is that notifications of a higher class include at least all of the sensors of the lower-class notifications. Also, since the higher-class notifications include more sensors ($\mathbf{D_{C2}} \subset \mathbf{D_{C1}}$), or the values for triggering the notification are more severe ($c_j \subset c_i$), the priority of higher-class notifications should be at least the same as the priority of corresponding lower-class notifications. This condition should be verified at consistency check for the final notification table.

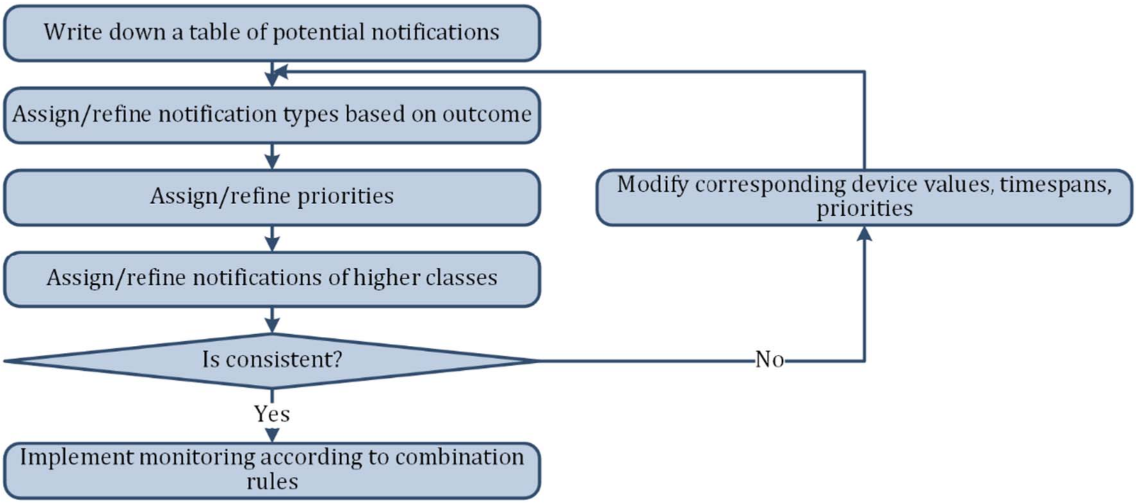

Write down a table of potential notifications
Assign/refine notification types based on outcome
Assign/refine priorities
Assign/refine notifications of higher classes
Is consistent?
No
Yes
Modify corresponding device values, timespans, priorities
Implement monitoring according to combination rules

Fig. 1. Alarm classification pipeline.

3.1.4. Alarm classification pipeline

The described principles of classification of abnormal situations can be summarized by the pipeline presented in Fig. 1.

The first step is to analyze the available devices in the system and propose potential abnormal situations for detection. At this stage, every tracked abnormal situation should be written as a separate notification, including those situations which have the same devices and outcomes, but require different actions (as in the example with notifications 5 and 6, which only differ in timespan and additional notification to maintenance personnel when abnormal condition is prolonged for a considerable amount of time). The next three steps assign notification types, priorities, and higher classes of notifications, as described in the previous sections.

The pipeline is iterative and should be repeated until notifications of the final table are consistent with their priorities, device values, and timespans. Specifically, all the higher-class notifications should have at least the same priorities as their corresponding lower-class notifications. The device values of notifications of the same type and with higher priorities should be included within the ranges of the corresponding lower-priority notifications or have longer timespan. Receivers of notifications of the same type and lower priority should be also receivers of higher-priority notifications. The resulting table of notifications for the illustrative example is shown in Table 2.

The proposed classification principles structure potential abnormal situations in a certain way. In particular, the adopted notions of higher-class notifications and assigning priorities for notifications of the same type contain at least the same device types as their lower-class and lower-priority notifications. With this in mind, the next section describes the principles of combining active notifications in the system at runtime.

3.2. Combination principles of ongoing notifications

When the system detects fulfillment of alarm conditions by a device or a set of devices, a new active or ongoing notification associated with this alarm is created. This active notification can be described as the following set of elements:

$$\mathbf{N_A} := \{id_A,\ \ ts_A,\ \ t_A,\ \ p_A,\ \ id_{l_A},\ \ \boldsymbol{HC_A},\ \ \boldsymbol{D_V},\ \ \boldsymbol{r_A}\} \tag{6}$$

where id_A is the ID of active or ongoing notification; ts_A is the timestamp of the notification; t_A is the type of active notification; p_A is the priority of active notification; id_{l_A} is the location ID of the place associated with notification; $\mathbf{HC_A}$ is the set of higher classes for the notification; $\mathbf{D_V}$ is the set of ongoing device values associated with the notification; and $\mathbf{r_A}$ is the set describing the receivers of the active notification (i.e. the users of the system).

The device values elements of the $\mathbf{D_V}$ set have the following form:

$$\mathbf{d_V} := \{id_{dV},\ \ t_{dV},\ \ v_{dV},\ \ ts_{dV},\ \ \boldsymbol{L_{dV}}\} \tag{7}$$

where id_{d_v} is the ID of a device; t_{d_v} is the type of device; v_{d_v} is the ongoing value of the device output; ts_{d_v} is the timestamp of the device value; and $\mathbf{L_{d_v}}$ is the set of location elements describing where the device is located (each element consists of the set $\mathbf{l} := \{t_l, id_l\}$, where t_l is the location type and id_l is the location ID).

The relationship between alarm condition and the associated active notification can be expressed in the following form:

$$if\ \exists \left\{ \mathbf{D_V}, \mathbf{N_C} \mid \begin{array}{l} \left(\forall\ \boldsymbol{d_C} \in \boldsymbol{D_C}, \exists \left\{ \boldsymbol{d_V} \in \boldsymbol{D_V} \mid \begin{array}{c} t_{dV} = t_{dC}, \\ v_{dV} \subseteq c \end{array} \right\} \right), \\ \exists\ \{\boldsymbol{l} \in \boldsymbol{L} \mid t_l = t_{lC},\quad \forall\ \boldsymbol{d_V} \in \boldsymbol{D_V},\quad id_l \in \boldsymbol{L_{dV}}\}, \\ \exists\ \{\boldsymbol{r} \in \boldsymbol{R} \mid \boldsymbol{R_C} \subseteq \boldsymbol{r}\} \end{array} \right\} \to \mathbf{N_A}$$

$$= \{id_A,\ \ ts_A,\ \ t_A = t_C,\ \ p_A = p_C,\ \ id_{l_A} = id_l,\ \ \mathbf{HC_A} = \mathbf{HC_C},\ \ \mathbf{D_V},\ \ \mathbf{r_A} = \mathbf{r}\} \tag{8}$$

or in other words, if there are a group of device values and notification

Table 2
Final description of the notifications in the illustrated scenario.

N	Description $\boldsymbol{o_C}$	Sensors $\mathbf{D_C}$		Duration to detect ts_c	Scope of detection $\boldsymbol{t_{l_c}}$	Receivers $\mathbf{R_C}$	Type $\boldsymbol{t_C}$	$\boldsymbol{p_C}$	Higher classes $\mathbf{HC_C}$
		Device types t_{d_c}	Values c						
1	Detection of fire	Smoke detector	true	0.5 min	Apartments, Buildings	Fire department, residents, nurses	FireAlarm	3	
2	Detection of fire	Indoor temperature sensor	> 40 °C	1 min	Apartments	Fire department, residents, nurses	FireAlarm	3	
3	High indoor temperature	Indoor temperature sensor	> 30 °C	10 min	Apartments	Resident	HighIndoorTemperature	2	FireAlarm
4	Low indoor temperature	Indoor temperature sensor	< 19 °C	30 min	Apartments	Resident, maintenance	LowIndoorTemperature	2	OpenedDoorInColdWeather
5	Opened door	Door state sensor	True	20 min	Apartments	Resident	OpenedDoor	0	OpenedDoorInColdWeather
6	Opened door	Door state sensor	True	60 min	Apartments	Resident, maintenance	OpenedDoor	1	OpenedDoorInColdWeather
7	Opened door in cold weather	Door state sensor Indoor temperature sensor Outdoor temperature sensor	True < 20 °C < 10 °C	10 min	Apartments	Resident, maintenance	OpenedDoorInColdWeather	2	

condition such that each device of the device group fulfills the notification condition, and these devices are located within the same location id, have the type specified in the notification condition, and there is at least one registered user in the system fulfilling the receivers requirements of the notification, then a new active notification is created in the system with parameters defined according to the associated notification condition and the ongoing device values. The important thing to notice here is that active notifications inherit their type, priority, and higher classes from the notification conditions described in the previous sections.

With these notations and rules defined for how active notifications in the system are created, the combination principles of ongoing notification can now be introduced.

3.2.1. Notifications of the same type and the same priority

When two active notifications exist with the same type, priority, location ID, and receiver, the device values of one notification are combined with the device values of the other notification, enabling the latter notification to be hidden for this particular receiver. More formally, this rule can be represented as follows:

$$\forall\ \mathbf{N_{A1}},\mathbf{N_{A2}}:\quad t_{A1}=t_{A2},\quad id_{l_{A1}}=id_{l_{A2}},\quad r_1=r_2,\quad p_{A1}=p_{A2}\ \rightarrow\ \mathbf{D_{V1}} = \mathbf{D_{V1}}\cap\mathbf{D_{V2}},\quad hide\ \mathbf{N_{A2}}\ for\ r_1 \tag{9}$$

The simple example of such a case is the detection of fire alarm by a smoke detector and temperature sensor in one apartment. The information from these devices would be combined into one notification for the appropriate users, reducing the information load while providing full information.

3.2.2. Notifications of higher priority

When two active notifications exist with the same type, location ID, and receiver, and the priority of one notification is higher than the priority of the other notification, the notification with lower priority should be hidden from the user. More formally, this rule can be written as follows:

$$\forall\ \mathbf{N_{A1}},\mathbf{N_{A2}}:\quad t_{A1}=t_{A2},\quad id_{l_{A1}}=id_{l_{A2}},\quad r_1=r_2,\quad p_{A1}>p_{A2}\ \rightarrow\ hide\ \mathbf{N_{A2}}\ for\ r_1 \tag{10}$$

The OpenedDoor notification is one such case. In case the door is opened for 1 h or more, instead of delivering additional notifications to users, it will appear as the same notification but with raised priority. In addition, since notifications of the same type and higher priority include at least the same set of devices as lower-priority notifications, hiding lower-priority notifications will not cause any information loss. This rule thereby offers the flexibility to describe situations of varying intensity without overwhelming users with additional information.

3.2.3. Notifications of higher class

When two active notifications exist with the same location ID and receiver, and the notification type of the first notification is an element of the higher-class set of the second notification, then the second notification should be hidden from the user. The formal definition of the rule is as follows:

$$\forall\ \mathbf{N_{A1}},\mathbf{N_{A2}}:\quad id_{l_{A1}}=id_{l_{A2}},\quad r_1=r_2,\quad t_{A1}\in\mathbf{HC_{A2}}\rightarrow hide\ \mathbf{N_{A2}}\ for\ r_1 \tag{11}$$

An illustrative example of this case is that active OpenedDoorInColdWeather suppresses active OpenedDoor and LowIndoorTemperature for the same location. In contrast to previous rules, where combination concerned only the same notification types, this rule suggests hiding notifications for different types. However, notifications of a higher class do also include at least the subset of devices with notifications of lower class and, therefore, hiding lower-class information will not lead to the loss of any data (i.e. delivered notifications always contain the values of the underlying devices that are causing it). This rule could be interpreted as the foremost delivery of more complex (higher-class) situations. In a way, the active status of higher-class notifications often partly suggests the reasons for lower-class situations (as with “OpenedDoorInColdWeather”, which might explain “LowIndoorTemperature” due to the outside door being opened, rather than breakage in the heating system). As a result, the user is exposed to less, but more essential, information, which still contains, as part of the notifications, information pieces of all abnormal situations.

It is worth noticing the connection of active notifications to the indicated receivers. Sometimes, the same notification could be shown to one user and hidden from another user. For example, let's consider two possible alarm types: “FireAlarm” and “AssistanceDuringFire”. The former is triggered if either of the devices detecting fire alarms is activated. The latter is triggered if either of the fire alarm devices is activated and it is detected that someone is present in the apartment with an active fire alarm. Thus, the “AssistanceDuringFire” is a higher class of “FireAlarm” notification. The receivers of the “FireAlarm” type notification could be all users of the system (or part of the users located in the building with the fire), while receivers of the “AssistanceDuringFire” notification type could be personnel of the buildings, such as nurses and maintenance. In the case where both of the notifications are active for the same location, personnel will receive “AssistanceDuringFire” notification, and “FireAlarm” will be hidden. However, “FireAlarm” will be still delivered to all residents of the building, because they are not indicated as receivers of “AssistanceDuringFire”, and the rules for combining notifications of a higher class will not be activated. As soon as “AssistanceDuringFire” is deactivated (e.g. a resident leaves the apartment), the lower-class “FireAlarm” notification will be visible again to the personnel.

4. Applying the approach to a case study

4.1. Building blocks of the implementation with enabling technologies

The proposed approach includes classification of existing alarms with runtime analysis of ongoing notifications and their relationships. In order to enable the implementation of this approach, the underlying software tools must support flexible modeling of relationships between various entities within the monitoring system alongside runtime analysis of ongoing data from devices. This subsection serves to introduce the backbone blocks of the implemented tools with a short introduction to the enabling technologies used in these blocks (Fig. 2). Each block is implemented as a separate Java web-based application and communicates with other applications through ZeroMQ[1] socket messaging library and web service implementation libraries.

4.1.1. Information source adapters

This block transforms input data from various devices and data sources to the unified format adopted within the functional blocks. It then publishes newly available data to the upper blocks of the architecture. The block includes the various adapters and can be extended to include other adapters depending on the required application case.

4.1.2. Knowledge model provider

The knowledge model provider block holds the knowledge model of the system with a reasoner, capable of analyzing this knowledge model and inferring related facts about it. The block provides access to the knowledge model for external applications through web services with the ability to query and modify the underlying knowledge model.

The knowledge model itself is the standardized representation of entities and their relationships, related to the purposes of the data and notification management in the monitoring systems. Generally

[1] ZeroMQ messaging library: http://zeromq.org/

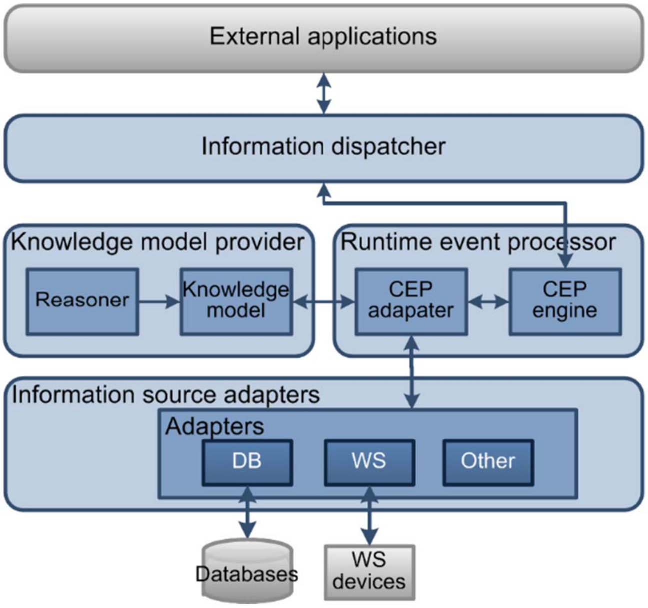


Fig. 2. Functional blocks of the implemented tools.

speaking, any representation technique could be used for capturing such relationships, as long as it has sufficient means for expressing the required relationships; however, many researchers indicate ontologies as the most flexible way of capturing object properties and their relationships with well supported tools and reasoners [13,40]. This work uses ontologies and SPARQL[2] querying for the purposes of modeling and extracting data from the model. All manipulations with knowledge model are done with the Jena[3] library, which provides an API for accessing and querying the model. The examples of knowledge model structure and queries can be found in [41,42].

4.1.3. Runtime event processor

The runtime event processor enables analysis of the ongoing data coming from the devices against the rules for detecting abnormal situations. Complex event processing (CEP) technology is used for these purposes. CEP is a defined set of tools and techniques for analyzing complex series of interrelated events and matching them with predefined patterns or rules [43]. This technology is well supported by a wide range of available CEP engines with APIs. Drools[4] rule engine is used in this work to provide CEP functionality with the CEP adapter block responsible for transforming incoming data from the devices and knowledge model into data that is understandable by the Drools engine. More detailed examples of rules and combination principles of data from ontology and external devices can be found in [44].

4.1.4. Information dispatcher

The information dispatcher is an intermediate block, which serves as a link between the implemented tools and external HMI applications. It receives data from the runtime event processor block, distributes it according to receivers of notifications, and then transforms the data into a format required by the external display application.

The aforementioned architecture blocks enable flexible modeling of different relationships between the elements of the monitoring system and runtime analysis of various abnormal situations and their relations to each other. These blocks have been used to implement the proposed approach for the case study of a rehabilitation facility, presented in the next subsection.

4.2. Rehabilitation facility case study

The rehabilitation facility under study is situated in Tampere, Finland. It provides rehabilitation services for people requiring treatment and is designed as a temporary residence for such groups as elderly people and people with physical disabilities. The main purpose of the facility is to provide services that support the recovery process after health-distorting events, and to prepare residents for independent living in their households following the period of therapy.

The facility is operated by two user groups: maintenance and nurse personnel. The first group is responsible for the overall functioning of the facility and reporting any major malfunctioning to the outside companies, while the maintenance personnel are physically present in the facility only when some of the equipment is malfunctioning (the maintenance services are subcontracted by the facility). The nurses are responsible for the overall wellbeing of the residents, their schedule, and for reporting any malfunctioning of the equipment raised by the residents to the maintenance personnel. Thus, the monitoring system of the facility must support the various needs of its personnel, assist in their daily tasks, and deliver timely and consistent information according to what is being detected in the environment.

The facility comprises five buildings, two of which are administrative buildings with office and common spaces, such as a dining room, kitchen, and several meeting rooms. The other three buildings contain 30 separate living apartments each having floor area of approximately 40 m^2. The apartments are equipped with various sensors, including IR, door state sensors, temperature sensors, vacuum cleaner inlets, water leak sensors, nurse call buttons, electric stove sensors (Fig. 3). In addition, if a resident's specific needs require it, the apartment can be equipped with supplementary sensors for monitoring specific user parameters, such as fall detectors and other wearable devices.

The devices in the facility are connected to a SCADA system, which provides control and monitoring of the basic building functions and keeps a one-year log of data from all devices. While the default system configuration supports the most important functionalities and monitoring of crucial abnormal situations in the facility, it does not fully exploit the capabilities of the sensors provided for the monitoring of abnormal situations, which involve values from multiple sensors. For example, a combination of IR and door state sensor might be used for detection of presence in the apartments, which may, in turn, be used for detection of other situations; however, due to the difficulties in configuration, these capabilities are not supported by the system in the facility.

5. Results

The architecture described in the previous section enables monitoring of multiple abnormal situations with various number and complexity of involved devices, and simplifies the reconfiguration of such systems [44]. Although detection of multiple abnormal situations of a different nature increases the specificity of information, it also increases the information flow to the users of the system. This fact becomes especially important when the monitoring system is a supportive tool for the personnel in daily tasks, rather than the primary tool for monitoring process status as in process industries. The proposed approach for combining ongoing notifications was tested in the rehabilitation facility previously described. The implemented tools have enabled runtime analysis of ongoing notifications, and their relations to each other, as described in Section 3.2. The one-year log data has been used for this purpose. The log data was interpreted as runtime data from devices enabled by the pseudo clock functionality of Drools CEP.

The approach for combining related notifications includes two phases: classification of existing alarms in the system; and a combination of ongoing notifications based on this classification and the status of the environment. These two phases were applied to the case of the rehabilitation facility.

[2] SPARQL Protocol and RDF Query Language: http://www.w3.org/wiki/SPARQL

[3] Apache Jena (framework for building semantic web and linked data applications): https://jena.apache.org/

[4] Drools rule management solution: http://www.drools.org/

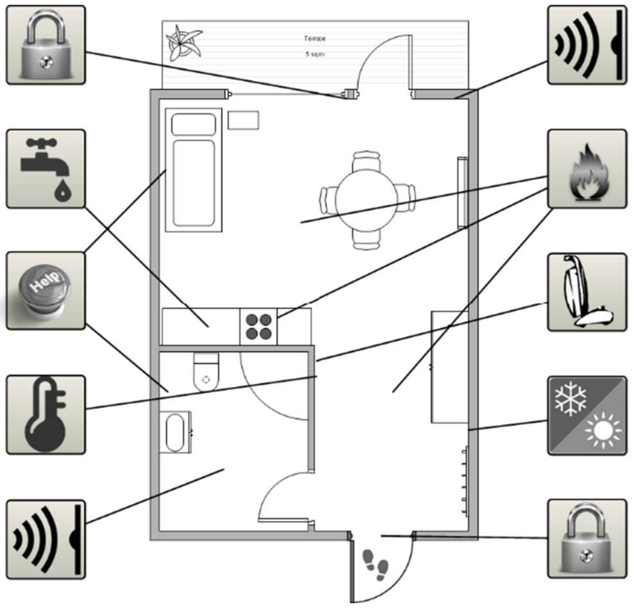


Fig. 3. An example of the installation of devices in an apartment in the rehabilitation facility.

5.1. Classification of alarms

The rehabilitation facility was thoroughly analyzed based on existing devices in the system, user groups, and responsibility of these user groups within the facility [45]. Based on this study, 25 conditions of abnormal situations of different types and priorities were identified for this system, some of which are specific to maintenance personnel in the tasks of running the facility. Such conditions include detection of high indoor temperature, water leaks, and opened doors in cold weather. Others are specific to the nurses group and are used to detect the general wellbeing of the users. Such conditions include monitoring of activities of the users inside the apartments based on IR sensors (ResidentHealthIssues notification type), triggering of nurse call buttons, and the detection of an active oven while the resident is not in the apartment. Finally, many notification types are intended for all user groups in the system, e.g. fire alarms, and opened doors notifications. It is worth noting that the original monitoring system supported only some of those notification types; specifically, only ten abnormal conditions were originally detected by the system—detection of the other notification types and priorities was enabled by the software tools, implemented with the general architecture described in Section 4.1.

The notification conditions were classified according to the principles proposed in Section 3.1. Based on these principles, the conditions were grouped by priorities and higher classes; in total, four priorities (0 to 3) and five notification types of a higher class have been identified. One-year log data from all devices in the facility was analyzed to detect abnormal situations within the facility. The log data was interpreted as runtime data from devices with pseudo clock functionality provided by Drools CEP engine. The examples of number and alarm type distribution for the most- and least-active months, in terms of notifications, are presented in Figs. 4 and 5 respectively. These figures represent the total number of active notifications per day as well as the distribution of different notification types.

As can be seen from Fig. 4, the facility had the maximum number of high indoor temperature notifications and, correspondingly, opened doors, because the beginning of August was hot. January was the least-intensive month in terms of notifications, with a maximum of around 70 notifications per day detected, in comparison to around 180 notifications per day in August. Considering that the monitoring of system data is not the primary task of the facility personnel (i.e. the personnel are actually working with the residents and/or doing repair work), the notification rates seen are already intensive, and there therefore needs to be a way of reducing the amount of ongoing notifications. The next subsection presents the results of applying combination principles to ongoing notifications in the system.

5.2. Combining ongoing notifications

The proposed combination principles enable a reduction in the information flow of ongoing notifications in the facility. The principles are based on combining active notifications according to their types, priorities, receivers, and overall status in the system and guarantee no information loss for users of the system, i.e. that data from all devices that are part of the notifications will be delivered to the users. The reduction of information flow is achieved by reducing the duplication of data pieces coming from the same devices.

The results of combination of notifications for August and January can be seen in Figs. 6 and 7, respectively. In these figures, the maximum number of simultaneously delivered notifications per day is depicted in bars and is categorized by priorities. This is compared to the information flow when combination principles are applied (dotted bars of Figs. 6 and 7). As can be seen, the information flow can be significantly

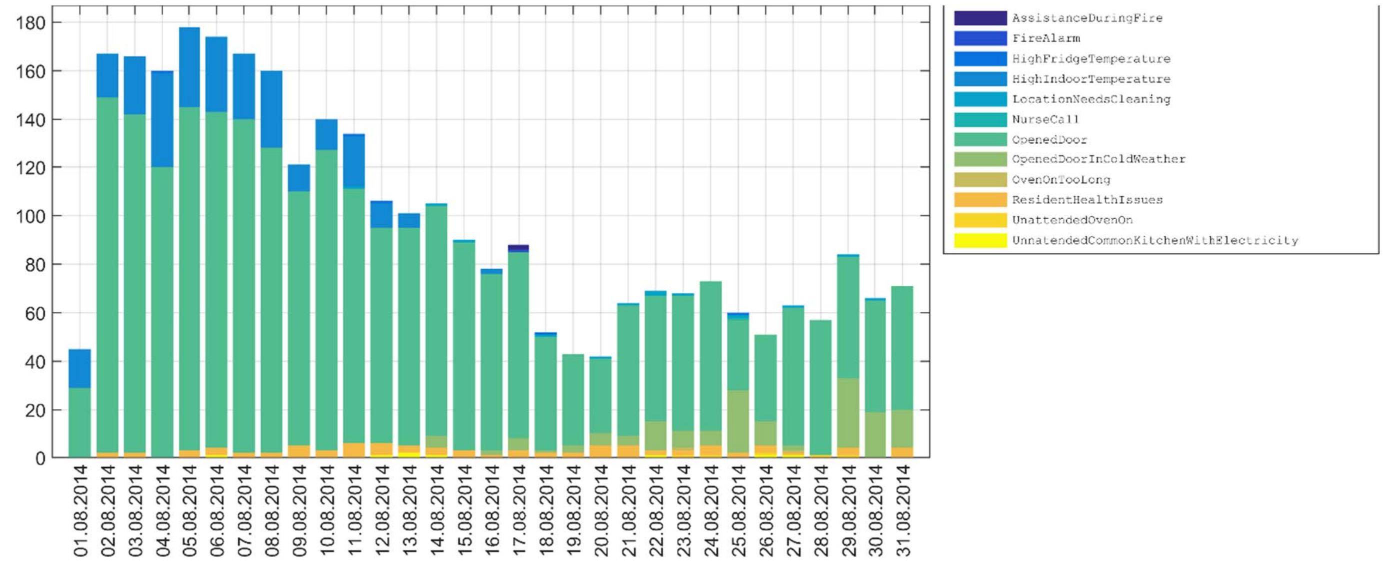


Fig. 4. Alarm type distribution for August 2014. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

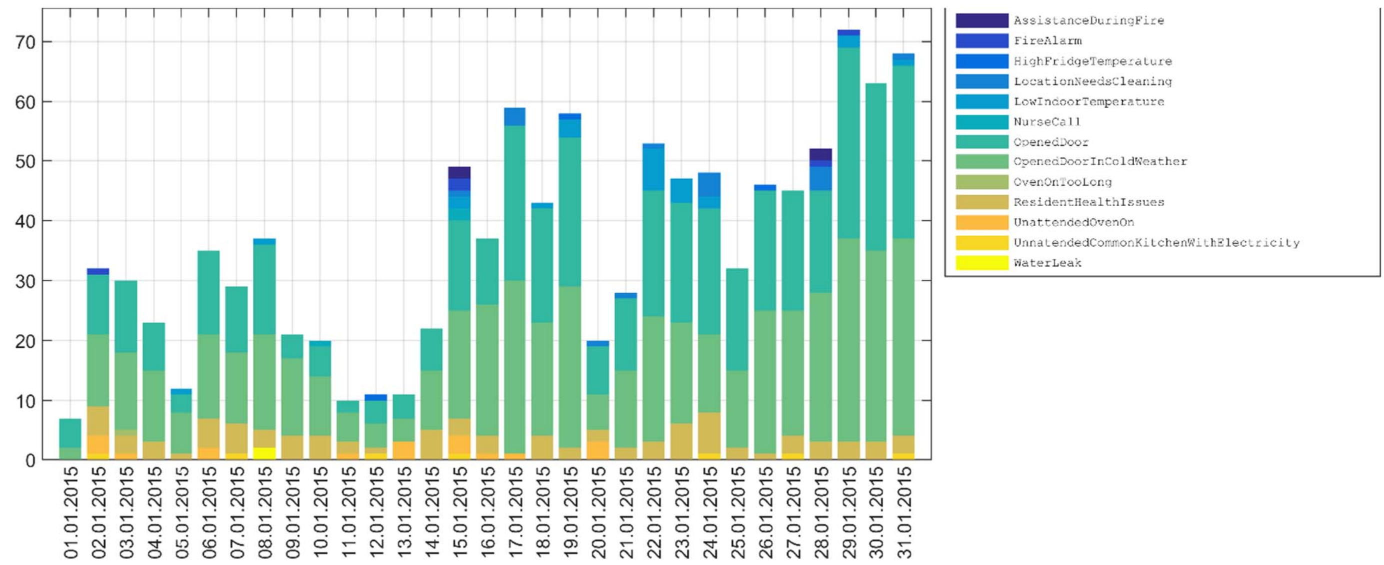


Fig. 5. Alarm type distribution for January 2015. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

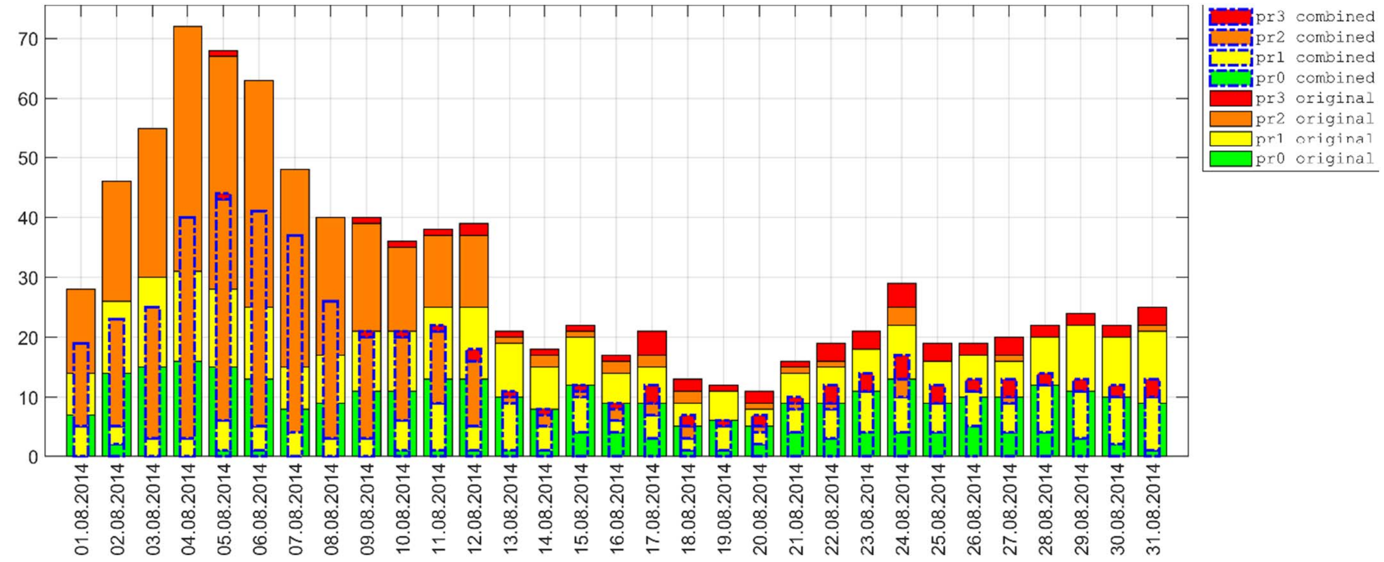


Fig. 6. Maximum simultaneous notifications per day in August 2014. Normal bars represent original notification rate, dotted bars represent combined notification rate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reduced, especially for notification-intensive months such as August. The reduction is made possible by combining related notifications of higher priorities and classes, which lead to reduction of delivered notifications of lower priority and classes. The average reduction of ongoing notifications for the whole year was about 42%; the maximum reduction rate of 51% was observed for June, and the minimum of 38% was observed for April and September.

Finally, the analysis of the combination of notifications was also performed on an hourly basis. Each hour was analyzed for the maximum number of ongoing notifications before and after the application of combination principles. The number of notifications was fitted to one of the four categories and pie charts were formed. The summary analysis is presented in Figs. 8 and 9 for August and January, respectively. These results show that the combination principles enable a significant shift from a high-intensity information flow (the number of active and visible notifications is > 20) to a lower-intensity information flow, even for an active notification month such as August. When the number of notifications is lower (such as in January), the combination principles enable the avoidance of any high-intensity hours, in terms of notification delivery (Fig. 9).

To summarize, the proposed approach covers the design and runtime phases of notification management. Clear design guidelines enable consistent classification and priority assignment for existing notifications and alarms, while runtime combination principles ensure a reduced information load for the users of the system. The main advantage of the proposed method is its ability to address the problem of information load in BM systems at runtime—the majority of existing solutions for alarm management perform offline analysis of log data, with no ability to support users during system operation.

6. Conclusion

This paper addresses the problem of intensive notification flows detecting abnormal situations in large-scale BM systems. The paper presents a novel approach for the classification of alarms in these systems and the combining of active related notifications at runtime, thereby reducing the information load for the users of the system.

The combination principles rely on classifying possible abnormal situations in the system and subsequently analyzing, during runtime, the active notifications by their type, priority, location, receivers, and

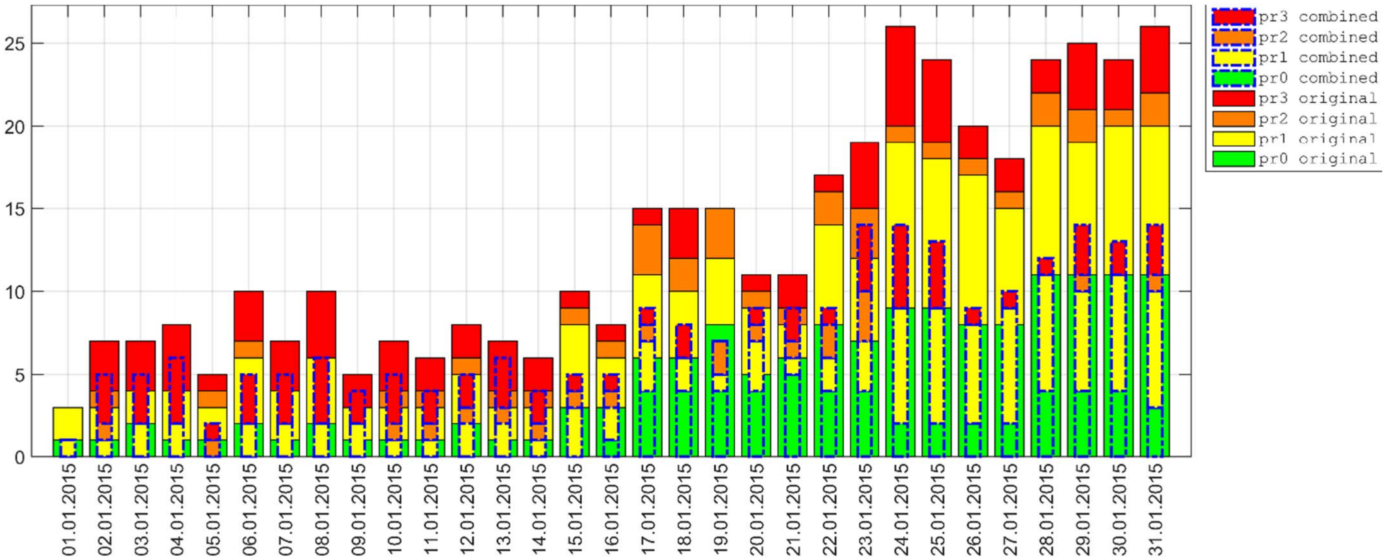


Fig. 7. Maximum simultaneous notifications per day in January 2015. Normal bars represent original notification rate, dotted bars represent combined notification rate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

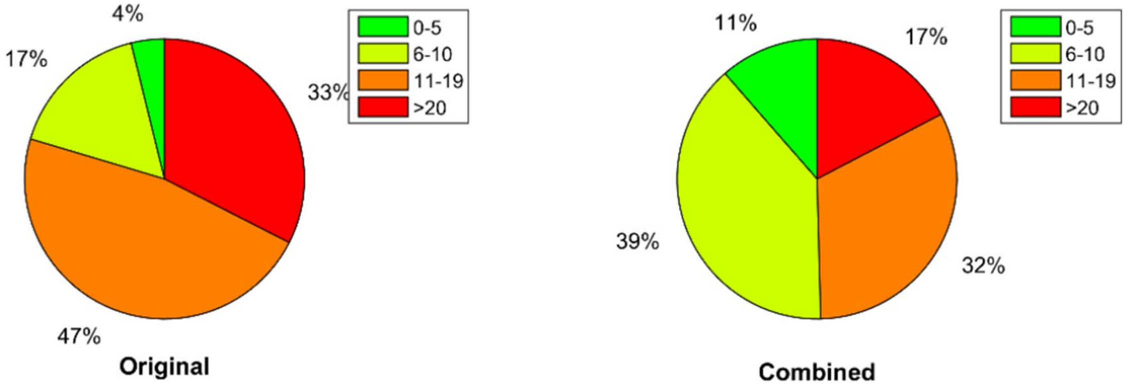


Fig. 8. Summary of maximum simultaneous notifications per hour in August 2014. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

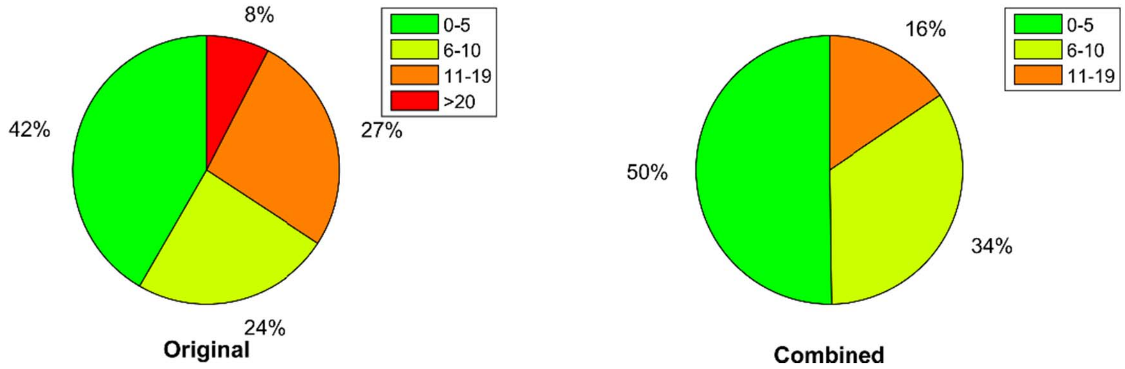


Fig. 9. Summary of maximum simultaneous notifications per hour in January 2015. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the ongoing status of the environment. The reduction in notification flows is achieved through a reduction in duplicated data pieces; thereby guaranteeing no information loss, meaning that each data piece from all devices that detect an abnormal situation will be delivered as part of the higher-priority and/or higher-class notification.

The paper provides details and formal description of the classification and combination principles of ongoing notifications, and presents the results of applying these principles to one-year log data from all devices at the rehabilitation facility under study. The log data was treated as runtime data from devices, thereby making it possible to analyze ongoing notifications and relate them to each other. The results of the notification analysis for the considered case study suggests that a reduction in information load to the users of the system of 42% of the peak number of ongoing notifications can be achieved.

It is expected that proposed approach will improve the awareness of the operators of the monitoring systems by reducing their information load by delivering only the most-relevant notifications according to ongoing conditions. Future work can be oriented in two directions: first, the proposed approach will be tested during the operation of the facility in order to explore the effects of information flow reduction on the actual decisions of the operators during runtime; and second, the approach will be adapted and tested in the domain of discrete manufacturing, which is, in many ways, similar to BM systems (as shown in [1]) and also suffers from delivery of excessive information flows to the system operators.

Acknowledgements

This work was supported by the Doctoral Programme of the President of the Tampere University of Technology.

References

[1] T. Sauter, S. Soucek, W. Kastner, D. Dietrich, The evolution of factory and building automation, IEEE Ind. Electron. Mag. 5 (2011) 35–48, http://dx.doi.org/10.1109/MIE.2011.942175.

[2] A. Shahi, M.N. Sulaiman, N. Mustapha, T. Perumal, Naive Bayesian decision model for the interoperability of heterogeneous systems in an intelligent building environment, Autom. Constr. 54 (2015) 83–92, http://dx.doi.org/10.1016/j.autcon.2015.03.015.

[3] P.H. Seong (Ed.), Reliability and Risk Issues in Large Scale Safety-critical Digital Control Systems, Springer, 2009, , http://dx.doi.org/10.1007/978-1-84800-384-2.

[4] Improve Operations Performance, Emerson Process Management, (2016), pp. 1–16 http://www3.emersonprocess.com/operationsperformance/whitepapers.html (accessed March 2, 2016).

[5] K. Larson, Operator effectiveness: the next frontier of process automation, ABB (2014) 1–4 https://www.yumpu.com/en/document/view/36262918/operator-effectiveness-the-next-frontier-of-control-global , Accessed date: 6 March 2015.

[6] R. Viswanathan, Auditing the Alarm Management System, InTech Magazine, https://www.isa.org/intech201412web01/, (2014) , Accessed date: 25 March 2015.

[7] J. O'Donnell, M. Keane, E. Morrissey, V. Bazjanac, Scenario modelling: a holistic environmental and energy management method for building operation optimisation, Energ. Buildings 62 (2013) 146–157, http://dx.doi.org/10.1016/j.enbuild.2012.10.060.

[8] K.L. Skillen, L. Chen, C.D. Nugent, M.P. Donnelly, W. Burns, I. Solheim, Ontological user profile modeling for context-aware application personalization, lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 7656, LNCS (2012) 261–268, http://dx.doi.org/10.1007/978-3-642-35377-2_36.

[9] J. Sinpoli, Don't Be Panicked by Alarm Management of Building Systems, AutomatedBuildings.com, (2011) http://www.automatedbuildings.com/news/may11/articles/sinopoli/110425011606sinopoli.html (accessed February 29, 2016).

[10] G. Fischer, Context-aware systems, Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12, ACM Press, New York, USA, 2012, pp. 287–294, , http://dx.doi.org/10.1145/2254556.2254611.

[11] A. Schmidt, Context-aware computing: context-awareness, context-aware user interfaces, and implicit interaction, in: M. Soegaard, R.F. Dam (Eds.), The Encyclopedia of Human-Computer Interaction, 2nd ed., The Interaction Design Foundation, 2014, http://www.interaction-design.org/encyclopedia/context-aware_computing.html , Accessed date: 30 April 2015.

[12] M. Baldauf, S. Dustdar, F. Rosenberg, A survey on context-aware systems, Int. J. Ad Hoc Ubiquitous Comput. 2 (2007) 263, http://dx.doi.org/10.1504/IJAHUC.2007.014070.

[13] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive Mob. Comput. 6 (2010) 161–180, http://dx.doi.org/10.1016/j.pmcj.2009.06.002.

[14] R. Velik, G. Zucker, Autonomous perception and decision making in building automation, IEEE Trans. Ind. Electron. 57 (2010) 3645–3652, http://dx.doi.org/10.1109/TIE.2009.2038985.

[15] J.A. Fraile, Y. De Paz, J. Bajo, J.F. De Paz, B. Pérez-Lancho, Context-aware multiagent system: planning home care tasks, Knowl. Inf. Syst. 40 (2013) 171–203, http://dx.doi.org/10.1007/s10115-013-0635-9.

[16] X. Xue, J. Chang, Z. Liu, Context-aware intelligent service system for coal mine industry, Comput. Ind. 65 (2014) 291–305, http://dx.doi.org/10.1016/j.compind.2013.11.010.

[17] M. Naderpour, J. Lu, G. Zhang, An intelligent situation awareness support system for safety-critical environments, Decis. Support. Syst. 59 (2014) 325–340, http://dx.doi.org/10.1016/j.dss.2014.01.004.

[18] A. Copetti, J.C.B. Leite, O. Loques, M. Fritsch Neves, A decision-making mechanism for context inference in pervasive healthcare environments, Decis. Support. Syst. 55 (2013) 528–537, http://dx.doi.org/10.1016/j.dss.2012.10.010.

[19] U. Varshney, A model for improving quality of decisions in mobile health, Decis. Support. Syst. 62 (2014) 66–77, http://dx.doi.org/10.1016/j.dss.2014.03.005.

[20] Alarm Management What, Why, Who and How? Honeywell, 2011, pp. 1–9 www.honeywellprocess.com/library/marketing/whitepapers/HoneywellAlarmManager_AlarmManagements4Ws_WP639.pdf (accessed February 29, 2016).

[21] S.R. Kondaveeti, I. Izadi, S.L. Shah, D.S. Shook, R. Kadali, T. Chen, Quantification of alarm chatter based on run length distributions, Chem. Eng. Res. Des. 91 (2013) 2550–2558, http://dx.doi.org/10.1016/j.cherd.2013.02.028.

[22] Y. Cheng, I. Izadi, T. Chen, Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm, Chem. Eng. Res. Des. 91 (2013) 1085–1094, http://dx.doi.org/10.1016/j.cherd.2012.11.001.

[23] R. Srinivasan, J. Liu, K.W. Lim, K.C. Tan, W.K. Ho, Intelligent Alarm Management in a Petroleum Refinery, http://scholarbank.nus.edu.sg/handle/10635/89257, (2004).

[24] J. Zhu, Y. Shu, J. Zhao, F. Yang, A dynamic alarm management strategy for chemical process transitions, J. Loss Prev. Process Ind. 30 (2014) 207–218, http://dx.doi.org/10.1016/j.jlp.2013.07.008.

[25] K. Ahmed, I. Izadi, T. Chen, D. Joe, T. Burton, Similarity analysis of industrial alarm flood data, IEEE Trans. Autom. Sci. Eng. 10 (2013) 452–457, http://dx.doi.org/10.1109/TASE.2012.2230627.

[26] O.M. Foong, S.B. Sulaiman, D. Rohaya, B. Awang, N. Syazwani, B. Abdullah, ALAP: alarm prioritization system for oil refinery, World Congress on Engineering and Computer Science, San Francisco, USA, 2009, pp. 1–6 (ISBN:978-988-18210-2-7).

[27] F. Higuchi, I. Yamamoto, T. Takai, M. Noda, H. Nishitani, Use of event correlation analysis to reduce number of alarms, Comput. Aided Chem. Eng. 27 (2009) 1521–1526, http://dx.doi.org/10.1016/S1570-7946(09)70644-3.

[28] Alarm Management Standards–Are You Taking Them Seriously? Honeywell, 2013, pp. 1–15 www.honeywellprocess.com/library/marketing/whitepapers/Alarm-Management-Standards-Taken-Seriously-Whitepaper.pdf (accessed February 29, 2016).

[29] T. Vardy, A Guide to Effective Alarm Management, Honeywell, 2015, pp. 1–12 www.honeywellprocess.com/library/marketing/whitepapers/Guide-to-Effective-AM-whitepaper_July2013.pdf , Accessed date: 29 February 2016.

[30] Alarm Management, Emerson Process Management, (2015), pp. 1–32 http://www2.emersonprocess.com/en-us/brands/deltav/documentation/pages/whitepapers.aspx (accessed February 29, 2016).

[31] J. Wang, F. Yang, T. Chen, S.L. Shah, An overview of industrial alarm systems: main causes for alarm overloading, research status, and open problems, IEEE Trans. Autom. Sci. Eng. 13 (2016) 1045–1061, http://dx.doi.org/10.1109/TASE.2015.2464234.

[32] D.G. Dunn, N.P. Sands, Diagnosing Your Alarm System, InTech Magazine Jul–Aug, https://www.isa.org/intech/20150805/, (2015) (accessed October 16, 2015).

[33] P. Urban, L. Landryova, Identification and evaluation of alarm logs from the alarm management system, 17th International Carpathian Control Conference (ICCC), IEEE, 2016, 2016, pp. 769–774, , http://dx.doi.org/10.1109/CarpathianCC.2016.7501199.

[34] S.R. Kondaveeti, I. Izadi, S.L. Shah, T. Black, T. Chen, Graphical tools for routine assessment of industrial alarm systems, Comput. Chem. Eng. 46 (2012) 39–47, http://dx.doi.org/10.1016/j.compchemeng.2012.06.042.

[35] W. Hu, M.S. Afzal, G. Brandt, E. Lau, T. Chen, S.L. Shah, An application of advanced alarm management tools to an oil sand extraction plant, IFAC-PapersOnLine, Elsevier Ltd., 2015, pp. 641–646, , http://dx.doi.org/10.1016/j.ifacol.2015.09.040.

[36] A.C.B. Garcia, C. Bentes, R.H.C. de Melo, B. Zadrozny, T.J.P. Penna, Sensor data analysis for equipment monitoring, Knowl. Inf. Syst. 28 (2011) 333–364, http://dx.doi.org/10.1007/s10115-010-0365-1.

[37] J. Folmer, B. Vogel-Heuser, Computing dependent industrial alarms for alarm flood reduction, International Multi-Conference on Systems, Signals & Devices, IEEE, 2012, pp. 1–6, , http://dx.doi.org/10.1109/SSD.2012.6198008.

[38] Z. Yang, J. Wang, T. Chen, Detection of correlated alarms based on similarity coefficients of binary data, IEEE Trans. Autom. Sci. Eng. 10 (2013) 1014–1025, http://dx.doi.org/10.1109/TASE.2013.2248000.

[39] M. Schleburg, L. Christiansen, N.F. Thornhill, A. Fay, A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system, J. Process Control 23 (2013) 839–851, http://dx.doi.org/10.1016/j.jprocont.2013.03.010.

[40] C. Hoareau, I. Satoh, Modeling and processing information for context-aware computing: a survey, N. Gener. Comput. 27 (2009) 177–196, http://dx.doi.org/10.1007/s00354-009-0060-5.

[41] Y. Evchina, A. Dvoryanchikova, J.L. Martinez Lastra, Semantic information management for user and context aware smart home with social services, IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, IEEE, 2013, pp. 262–268, , http://dx.doi.org/10.1109/CogSIMA.2013.6523856.

[42] Y. Evchina, J.L. Martinez Lastra, Hybrid approach for selective delivery of information streams in data-intensive monitoring systems, Adv. Eng. Inform. 30 (2016) 537–552, http://dx.doi.org/10.1016/j.aei.2016.07.006.

[43] D. Luckham, The power of events: an introduction to complex event processing in distributed enterprise systems, Rule Representation, Interchange and Reasoning on the Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, p. 3, , http://dx.doi.org/10.1007/978-3-540-88808-6_2.

[44] Y. Evchina, J.L. Martinez Lastra, Semantic-driven CEP for delivery of information streams in data-intensive monitoring systems, 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), IEEE, 2015, pp. 1251–1256, , http://dx.doi.org/10.1109/INDIN.2015.7281914.

[45] Y. Evchina, J.L. Martinez Lastra, An ICT-driven hybrid automation system for elderly care support: a rehabilitation facility study case, J. Hous. Elder. 30 (2016) 52–74, http://dx.doi.org/10.1080/02763893.2015.1129382.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4146-9

ISSN 1459-2045