Department of Computer Science

Web Technology based Smart Home Interoperability

Hannu Järvinen

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 183/2015

Web Technology based Smart Home Interoperability

Hannu Järvinen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the auditorium AS1 of the school on 11 December 2015 at noon.

Aalto University
School of Science
Department of Computer Science
Web Services Group

Supervising professor
Prof. Petri Vuorimaa

Thesis advisor
Prof. Petri Vuorimaa

Preliminary examiners
Prof. Jukka Vanhala
Tampere University of Technology, Finland

Prof. Petri Pulli
University of Oulu, Finland

Opponents
Dr. Dominique Guinard
Co-founder & CTO at EVRYTHNG, United Kingdom

Aalto University publication series
DOCTORAL DISSERTATIONS 183/2015

ISBN 978-952-60-6509-0 (printed)
ISBN 978-952-60-6510-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6510-6

Unigrafia Oy
Helsinki 2015

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Hannu Järvinen

Name of the doctoral dissertation
Web Technology based Smart Home Interoperability

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 183/2015

Field of research Media Technology

Manuscript submitted 18 August 2015 **Date of the defence** 11 December 2015

Permission to publish granted (date) 12 October 2015 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

A smart home commonly refers to a home equipped with devices and systems that are connected to each other and can be centrally managed. Remote management can also be provided using, e.g., mobile devices. Research on smart homes has been conducted for decades, but solutions have failed to become popular among common people. One of the problems has been weak interoperability between different devices. However, technological advances in various fields have made smart homes more possible than earlier. We are carrying with us mobile devices not only suitable for the system management but also for sensor measurements, such as automatic positioning. In addition, the required infrastructure for networking is common in modern buildings in forms of wireless and wired local area networks. Also, remote access can be realized over Internet connection.

This dissertation focuses on improving interoperability of smart home devices and systems using Web technologies. The proposed solutions are mainly based on the REST architectural model. They exploit the HTTP protocol commonly used by Web browsers. In addition, Semantic Web technologies and multi-agent system intelligence are proposed for the higher-level interoperability.

The dissertation presents an interoperability model that extends commonly known interoperability layers with functional interoperability. Derived from the cooperation between different devices, operational logic and automation are commonly considered as objectives of home and building automation systems. Modeling functional interoperability, this dissertation aims to promote these objectives and getting solutions eventually available for the end users.

The dissertation presents interoperability layers with existing related research on each of the layers. Web technology based solutions are proposed for improved interoperability. Proof of concept implementations are reviewed against the interoperability model. Interoperability solutions provided by the implementations are described in detail.

In the findings, different interoperability solutions are summed up with the used Web technologies. Implementations can be considered to prove that interoperability can be achieved with the used technologies.

As a research result, interoperability model emphasizes the importance of functionality in smart home interoperability. Results from the implementations show that Web technologies are well suited for implementing interoperability in smart homes.

Keywords Web service, smart home, interoperability

ISBN (printed) 978-952-60-6509-0 **ISBN (pdf)** 978-952-60-6510-6

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2015

Pages 162 **urn** http://urn.fi/URN:ISBN:978-952-60-6510-6

A! **Aalto-yliopisto**

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tiivistelmä

Tekijä
Hannu Järvinen

Väitöskirjan nimi
Yhteentoimivuus älytaloissa käyttäen Web-teknologioita

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 183/2015

Tutkimusala Mediatekniikka

Käsikirjoituksen pvm 18.08.2015 **Väitöspäivä** 11.12.2015

Julkaisuluvan myöntämispäivä 12.10.2015 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Älytalolla tarkoitetaan tavallisesti kotia, jonka laitteet ja järjestelmät ovat kytketty toisiinsa ja niitä voidaan hallita keskitetysti. Myös etäkäyttäminen voi olla mahdollista esimerkiksi mobiililaitteiden avulla. Tutkimusta älytaloista löytyy jo vuosikymmenien takaa, mutta ratkaisut eivät kuitenkaan ole yleistyneet tavallisen kansan keskuudessa. Yhtenä ongelmana on ollut eri laitteiden välinen heikko yhteentoimivuus. Puitteet älytalojen yleistymiselle ovat kuitenkin nyt paremmat kuin koskaan. Kannamme jatkuvasti mukanamme älylaitteita, jotka soveltuvat tällaisten järjestelmien hallintaan ja jotka mahdollistavat myös esimerkiksi käyttäjien automaattisen paikannuksen. Lisäksi tarvittava verkkoinfrastruktuuri on moderneissa taloissa valmiina langallisten ja langattomien lähiverkkojen muodossa. Myös etäyhteys on toteutettavissa Internetin välityksellä.

Tässä väitöskirjassa keskitytään älytalojen eri laitteiden ja järjestelmien yhteentoimivuuden parantamiseen Web-teknologioita käyttäen. Esitetyt ratkaisut perustuvat pääasiassa REST-arkkitehtuurimalliin ja pohjautuvat verkkoselaintenkin käyttämään HTTP-protokollaan. Lisäksi korkeamman tason yhteentoimivuuteen esitetään käytettäväksi Semanttisesta Webistä tuttuja teknologioita ja moniagenttijärjestelmien tarjoamaa älykkyyttä.

Väitöskirjassa esitellään yhteentoimivuusmalli, joka kuvaa perinteisten yhteentoimivuustasojen lisäksi myös toiminnallista yhteentoimivuutta. Erilaisten rakennus- ja taloautomaatiojärjestelmien tavoitteina pidetään usein laitteiden välisen yhteistyön kautta saavutettavaa toimintalogiikkaa ja automaatiota. Kuvaamalla toiminnallista yhteentoimivuutta, väitöskirjatutkimus pyrkii edistämään näiden tavoitteiden saavuttamista ja lopulta ratkaisujen saattamista loppukäyttäjien ulottuville.

Tutkimuksessa kuvataan eri yhteentoimivuuden tasot ja esitellään niihin liittyvää aikaisempaa tutkimusta. Yhteentoimivuuden parantamiseksi esitetään Web-teknologioihin perustuvia ratkaisuja. Konseptit todennetaan toteutuksin, joita peilataan yhteentoimivuusmallia vasten. Toteutusten tarjoamat yhteentoimivuuden ratkaisut käydään yksityiskohtaisesti läpi.

Tutkimuksen tuloksissa yhteentoimivuusratkaisut kootaan yhteen ja toteutuksissa käytetyt Web-teknologiat käydään läpi. Toteutuksia voidaan pitää todisteena siitä, että yhteentoimivuuden toteuttaminen on mahdollista käytettyjen teknologioiden avulla.

Tutkimustuloksena yhteentoimivuusmalli korostaa toiminnallisuuden merkitystä älytalojen yhteentoimivuudessa. Toteutuksista saatujen tuloksien valossa Web-teknologiat soveltuvat hyvin älytalojen yhteentoimivuuden toteuttamiseen.

Avainsanat www-sovelluspalvelu, älytalo, yhteentoimivuus

ISBN (painettu) 978-952-60-6509-0 **ISBN (pdf)** 978-952-60-6510-6

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2015

Sivumäärä 162 **urn** http://urn.fi/URN:ISBN:978-952-60-6510-6

Preface

I would like to express my gratitude to Professor Petri Vuorimaa, who has supervised this dissertation. He gave me the opportunity to learn how to do research, write scientific articles, and finally write this dissertation. Without him, this work would never have been materialized.

I would like to thank my colleagues in the Web services research group who have shared this journey with me. During the years, help has always been available in different forms. When problems have been encountered, numerous conversations have guided the research toward the right direction. Also, relaxing moments during the lunch breaks and with the table hockey gave a proper balance to the hard work. Thanks belong to Dr. Jari Kleimola, Markku Laine, Pia Tukkinen, and Evgenia Litvinova. I also want to thank the pre-examiners for their professional feedback on this dissertation.

I am deeply grateful to co-authors Professor Petri Vuorimaa, Professor Ulises Cortés, Dr. Dario Garcia-Gasulla, and Andrey Litvinov. Without their contribution this work would have been impossible to complete.

I also want to thank Professor Paul F. M. J. Verschure for the opportunity to visit the Pompeu Fabra University for collaboration, and Dr. Ulysses Bernardet for the professional guidance during my visit. I want to thank again Professor Ulises Cortés for the opportunity to visit Polytechnic University of Catalonia, and for the fruitful collaboration during my visit.

Most importantly, I want to thank my parents Marja-Leena and Sulo Järvinen, my sisters Ulla Rikkonen and Leena Kärras, and my brother Matti Järvinen for their unconditional love and for always being there for me. Finally, I would like to express my deepest appreciation and love to Sanna-Mari. I love you.

Espoo, October 29, 2015,

Hannu Järvinen

Contents

List of Publications

This dissertation consists of an overview and of the following publications, which are referred to in the text by their Roman numerals.

I Hannu Järvinen, Andrey Litvinov, and Petri Vuorimaa. Integration Platform for Home and Building Automation Systems. In *Proceedings of the eighth IEEE Consumer Communications and Networking Conference*. Pages 292-296. IEEE. 2011.

II Hannu Järvinen and Petri Vuorimaa. Anticipatory Lighting in Smart Building. In *Proceedings of the ninth IEEE Consumer Communications and Networking Conference*. Pages 390-394. IEEE. 2012.

III Hannu Järvinen, Dario Garcia-Gasulla, and Ulises Cortés. A Push-based Agent Communication Model Empowering Assistive Technologies. In *International Journal on Artificial Intelligence Tools*. Vol. 23. No. 1. Pages 1-30. World Scientific. 2014.

IV Hannu Järvinen and Petri Vuorimaa. Interoperability for Web Services based Smart Home Control Systems. In *Proceedings of the tenth International Conference on Web Information Systems and Technologies*. Pages 93-103. SCITEPRESS. 2014.

V Hannu Järvinen and Petri Vuorimaa. Semantic Interoperability for Web Services based Smart Home Systems. In *Proceedings of the seventh International Conference on Agents and Artificial Intelligence*. Pages 141-148. SCITEPRESS. 2015.

Author's Contribution

Publication I: Integration Platform for Home and Building Automation Systems

Publication I describes a method for connecting smart home devices to an integration platform using Web services. Comparison is made between three different polling methods, and the system is evaluated with the conducted efficiency measurements. The solution provides interconnection and syntactic interoperability for smart homes.

The author designed and implemented the first integration platform prototype. The author wrote the paper in cooperation with the second author, contributing half of the text while getting proofreading help from Prof. Petri Vuorimaa.

Publication II: Anticipatory Lighting in Smart Building

Publication II presents a general smart home system model with distributed intelligence and an implementation of that model. The implementation includes an indoor tracking based prediction module for anticipating lighting conditions and a generic rule engine handling the rest of the required logic. The system is evaluated with experiments conducted in laboratory environment with participants. The solution provides interoperation for Web services based smart home systems.

The author designed and implemented the entire work. The author wrote all of the text, while getting proofreading help from Prof. Petri Vuorimaa.

Publication III: A Push-based Agent Communication Model Empowering Assistive Technologies

Publication III describes a WebSocket Message Transport Protocol for agent platforms with an implementation for the JADE platform, and a push-based subscription model for the communication. Based on the measurements, WebSocket communication is evaluated against the HTTP MTP. A senior living use case considers a smart home system with agent platforms handling pill dispensing, access control, and communication. The solution provides interoperability on interconnection and syntactic layers.

The author came up with the idea, and designed and implemented the WebSocket Message Transport Protocol. The author made the measurements and the analysis. The author wrote the paper in cooperation with the second author, contributing half of the text while getting feedback and proofreading help from Prof. Ulises Cortés.

Publication IV: Interoperability for Web Services based Smart Home Control Systems

Publication IV presents a solution for enabling interoperation in smart home systems using XML based rules. To ensure the interoperability, requirements are defined for Web services based building automation control systems. A standard building automation guideline, OBIX, is used to provide interoperability on the low level, but also adopted for the rule management and description on the high level. The resulting rule engine architecture and implementation are evaluated against the requirements.

The author designed and implemented the entire work. The author wrote all of the text, while getting proofreading help from Prof. Petri Vuorimaa.

Publication V: Semantic Interoperability for Web Services based Smart Home Systems

Publication V presents a solution for Web services based smart home system to publish semantic data, and to communicate with agents systems. Communication of browser-based agents with a smart home system is demonstrated. Efficiency of the system is measured and the results provided. The presented solution provides semantic interoperability, and a basis for exploiting intelligence from multi-agent systems in smart home system control.

The author designed and implemented the entire work. The author wrote all of the text, while getting proofreading help from Prof. Petri Vuorimaa.

List of Abbreviations

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ACL	Agent Communication Language
AI	Artificial Intelligence
AP	Agent Platform
BA	Building Automation
BACnet	Building Automation and Control Networks
BAS	Building Automation System
BEMS	Building Energy Management System
BMS	Building Management System
CABA	Continental Automated Buildings Association
CoAP	Constrained Application Protocol
CORBA	Common Object Request Broker Architecture
DDC	Direct Digital Control
DLNA	Digital Living Network Alliance
DPWS	Devices Profile for Web Services
EAI	Enterprise Application Integration
ECA	Event-Condition-Action
EHS	European Home Systems Protocol
EIB	European Installation Bus
EIF	European Interoperability Framework
EMS	Energy Management System
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
FIPA	Foundation for Intelligent Physical Agents
FMS	Facility Management System
HA	Home Automation
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HVAC	Heating, Ventilating, and Air Conditioning
IA	Intelligent Agent
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization

JADE	Java Agent DEvelopment Framework
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LAN	Local Area Network
LDA	Linear Discriminant Analysis
LED	Light-Emitting Diode
M2MXML	Machine-To-Machine Extensible Markup Language
MAS	Multi-Agent System
MTP	Message Transport Protocol
NCS	Networked Control System
OASIS	Organization for the Advancement of Structured Information Standards
OBIX	Open Building Information Xchange
oFMS	open Facility Management Server
OPC	Object Linking and Embedding for Process Control
OSGi	Open Services Gateway initiative
OSI	Open Systems Interconnection
OWL	Web Ontology Language
PLC	Programmable Logic Controller
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
REST	REpresentational State Transfer
RPC	Remote Procedure Call
RTT	Round-Trip Time
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
Turtle	Terse RDF Triple Language
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WoT	Web of Things
WSA	Web Services Architecture
WSDL	Web Services Description Language
WS-*	Big Web services
WWW	World Wide Web
XIM	eXperience Induction Machine
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

1 Introduction

The main theme of my research during the last years has been the interoperability of RESTful Web services based smart home systems. This dissertation unifies that research under one topic. It considers a layered interoperability model, and presents the research done on each of the layers. Figure 1 depicts a high-level overview of the research area and the relations of the publications to the related topics.

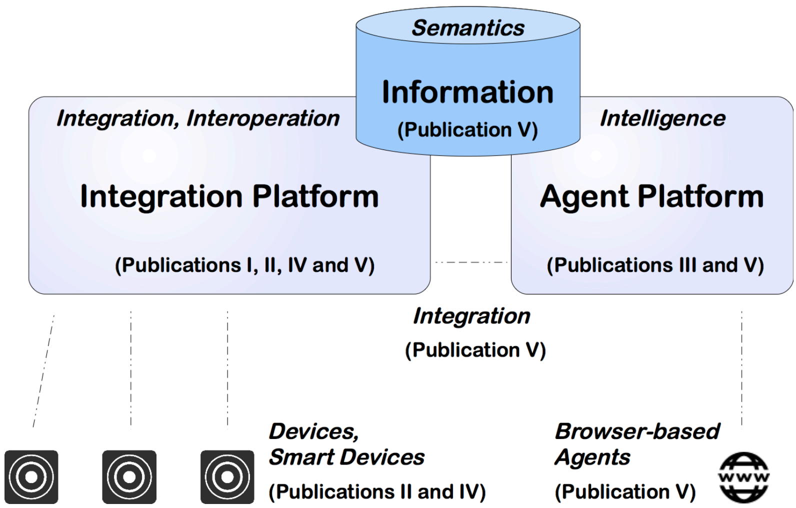

Figure 1: High-level overview of the dissertation domain and the relations of each publication to the themes.

The research presented in each publication has both a strong connection to the theme of the dissertation, but also a distinct specific focus based on the type of interoperability presented in that research. Unifying factors in the publications are interoperability, Web services, and, of course, smart homes.

The scope of this dissertation is defined as follows. Discussed Web services are mainly limited to RESTful Web services. In addition, REST definition (Fielding, 2000) is not strictly followed, as some of the REST constraints are not always fulfilled. Smart home interoperability is considered as defined in this dissertation, while also compared with similar smart home and smart home interoperability models. Focus of smart homes is kept mainly on the actual home environment, omitting enterprise applications, *e.g.*, remote senior care services, from the larger discussion. The main focus is kept on different layers of the smart home interoperability model, and on the contributions made on these areas to realize interoperability using RESTful Web services, and other Web and agent technologies.

1.1 Motivation

Domestic technology has long been undervalued. It has been referred to as the Cinderella technology, emphasizing the lack of glory it has received. It simply has been a relatively uninteresting domain for the male engineers and researchers. (Cockburn, 1997) Fortunately, smart home research finally presents futuristic scenarios already known from the science fiction, making the domestic technology research of today more interesting. The area holds a huge potential that is to be unleashed in the very near future.

Today, various manufacturers are already offering smart home solutions for their customers. These products generally provide solutions to individual Home Automation (HA) tasks, while cooperation between different devices and manufacturers is less common. However, technological advances in various fields have made multi-vendor smart homes more possible than earlier. Separate smart devices, ubiquitous standard networking technologies, and on the software side, interoperable applications, are the required components that already exist today. Thus, as the technology and infrastructure are now available for the smart homes, they can finally become reality in the very near future (Yamazaki, 2006).

Smart homes can appear in different forms. One of the most significant factors affecting the smart home feature set is the interoperability. Only if the system components are able to interoperate, the capability of the system is more than the sum of its parts. In agent technologies, one of the basic requirements of an Intelligent Agent (IA) is the ability for social behavior with other agents (Wooldridge and Jennings, 1995). Among other capabilities, this cooperation allows Multi-Agent Systems (MASs) to perform tasks that are too complex to be solved by one centralized control system (Sánchez *et al.*, 2011). Similarly, the degree of interoperability of the system components defines the final capabilities of the smart home.

The essential problem today is that the smart homes are not interoperable. The use of non-standard, closed, and proprietary protocols can be considered as the underlying cause. In turn, financial interests of the companies are one of the reasons for this. A common standard used by several manufacturers could lead to interoperability between different solutions. However, a higher governing body might be required to enforce the manufacturers to adopt the standard. Another solution to the problem would be to use only one system. In that case, the system could be closed and use proprietary protocols. However, as a single company would handle all the development, the system could be built interoperable by design. In both of the approaches, the protocols would have to be technologically capable of providing required means to achieve different types of interoperability.

Numerous protocols have been developed for the device communication. Hence, a modern Building Automation System (BAS) comprises of variety of different technologies. A more recent research trend is to concentrate on high-level integration and allow the heterogeneity of the low-level device protocols. At the same time, the World Wide Web (WWW) has become extremely popular. The Web has proven to be a good platform for integrating and composing applications that run on heterogeneous hardware and software connected to the Internet (Guinard, 2011). Accordingly, Web services have been successfully applied to the integration of distributed systems. While earlier middleware and Enterprise Application Integration (EAI) platforms have been used rather successfully for application integration in certain settings, problems in cross-organizational interactions and homogeneity of middleware platforms are found as their weaknesses. With Web services, better integration can be achieved using Service-Oriented Architectures (SOA), redesign of middleware communication, and open standards. (Alonso *et al.*, 2004) (Linthicum, 2003) Web services can be now considered as the *de facto* standard for solving interoperability problems in large networks. This success of Web services as an integration technology has led to the development of Web services based high-level Building Automation (BA) standards. The approach is promising, since on the Web heterogeneous applications on diverse platforms are able to communicate together.

Internet of Things (IoT) is a term used to describe a network of humans, animals, and physical objects that have integrated services and Internet connectivity alongside with a uniquely identifiable IP address. The idea is to provide better services and exchange of data through this connectivity. However, IoT lacks generally agreed standard protocols and data formats that realize higher-level interoperability between things and thus enable, *e.g.*, composite applications. Web of Things (WoT) is an effort for exploiting Web technologies as a platform for integrating smart things. It is based on REST and has the objective of providing an application layer for the IoT. While Internet connects computers operating on heterogeneous hardware and software platforms from all around the world, the Web has shown to provide efficient and scalable integration and composition of applications on top of the Internet using relatively simple open standards, including HTTP, URI, HTML, XML and JSON. REST and Web technology have high availability of development tools, built-in functionality for interaction with browsers, and have gained worldwide acceptance. Simplicity and scalability of the Web have provided it with a huge success. (Guinard, 2011) Thus, Web might well be the future platform for smart things (Guinard *et al.*, 2010b).

Exploiting Web services as an integration technology in smart home research can provide solutions for the interoperability. However, development of interoperable smart home systems is not trivial. In addition to the problem of connecting devices into the same system, interoperability can be extended to consider interoperations between these devices, and intelligent behavior that the system is expected to induce. Some other aspects, such as finding and sharing Web based devices and services have already been addressed earlier (Guinard, 2011). Hence, the research presented in this dissertation focuses on interoperability based on data representation capabilities and functional properties, and on providing interoperability using Web services as a common integration technology. While there is earlier research on the field, maturity of the required technology and available infrastructure make it a relevant research domain. The research is also justified, as despite of earlier research efforts, truly interoperable smart home systems are not yet reality for the general public.

1.2 Research Questions

As discussed above, heterogeneous building and home automation devices are not working in an interoperable fashion. Despite of numerous research projects and commercial products, true interoperability can only be expected in situations where one vendor provides all the devices. Research in this dissertation presents how Web technologies could be used to improve smart home interoperability in different ways. Accordingly, the main research question in this dissertation is:

> *How can Web technologies be used to improve smart home interoperability?*

To answer this question, the following sub-questions are considered:

> **Q1:** What are the key elements required for the smart home interoperability?

> **Q2:** How can Web technologies provide solutions for the different elements of the smart home interoperability?

1.3 Research Methods

Design science research considers different analytical techniques and perspectives for performing research in information systems domain. The work of design science researchers concentrates on understanding, explaining, and improving information systems (Hevner and Chatterjee, 2010). The main purpose of design science research is to achieve knowledge and understanding of a problem domain by building and application of a designed artifact (Hevner *et al.*, 2004). Similarly, in constructive research the key idea is the construction based on the existing knowledge used in novel ways. Artifacts such as models, diagrams, system designs, algorithms and artificial languages are typical constructs. These solutions are mostly designed and developed, and not in the first place discovered, while inspiration undeniably sometimes comes from the nature. Constructive research is used to build an artifact for solving a domain specific problem in order to create knowledge about how that problem can be solved, understood, or modeled in principle. (Crnkovic, 2010)

The design science research presented in this dissertation is constructive research where the *constructs* are system models, software implementations, and an extended interoperability model.

The work investigates application of Web services for enhancing interoperability in smart home systems. The nature of the research requires specific forms of validation, as comparisons to existing solutions are not always possible or relevant. However, in each phase the development of the software is based on the review of the existing solutions. Decisions concerning the development are then made trying to promote simple, open, and standard techniques, preferably based on Web technologies. The actual validation of the software includes both quantitative and qualitative methods.

The validation of the software is done in laboratory tests. Some of the tests are simulations, while the rest are made with voluntary participants. The applied quantitative methods are questionnaires, measurements, and statistics. In practice, the data is gathered during and after the experiments and then analyzed off-line. The data includes results for Likert items, network traffic packets, logs, Round-Trip Times (RTTs), and other information recorded by software specially developed for that purpose. Statistical analysis of this type of data produces information about the performance, functioning, and delay of the system. Also, it provides means for comparisons between different experimental conditions, where, *e.g.*, used protocol is an independent variable. In contrast to the rest of the quantitative data, Likert item data has to be considered as subjective opinions of the test subjects, instead of objective measurements.

Alongside with the quantitative methods, qualitative methods are also applied. Qualitative methods are used to find out how a certain setting is experienced among the participants, and what are the reasons for those experiences. The applied qualitative methods include observations and questionnaires. Observations are made during the experiments and recorded as written notes or fetched afterwards from the recordings. Questionnaires are used to collect data from the participants after the experiments.

Quantitative data was gathered for the research in all of the publications. Qualitative data was gathered using open-ended questions in a questionnaire in research described in Publications II and IV.

Research environments used for tests were laboratory environments. User experiments with the anticipatory lighting in Publication II were conducted in the eXperience Induction Machine (XIM), a space equipped with numerous sensors and actuators (Bernardet *et al.*, 2010). The environment is highly adaptive for different research purposes related to user experience, and has been used also for, *e.g.*, studying user experiences in virtual environment navigation (Järvinen *et al.*, 2011).

1.4 Contributions

The main contributions of the research are the interoperability model, system models, and the developed software. The interoperability model is extended from the commonly used models to include additional functional layers that build on top of the traditional capability layers. The interoperability model is presented in Chapter 3, and system models with the implementations in Chapter 4. Details of the software implementations are described in the publications. Using Web technologies, presented software implementations provide solutions for the different layers of the interoperability model. The software contributions present the following generalizable ideas:

> *A standalone integration platform that provides home for external heterogeneous system components through Web services interface;*
>
> *A rule engine that operates in distributed device environments over Web services interface;*
>
> *A smart device that cooperates with a rule engine and other devices through Web services interface;*
>
> *WebSocket communication for multi-agent systems;*
>
> *Transformation of smart home data into semantic representation based on an ontology; and*
>
> *Integration of a smart home system with a multi-agent system.*

The contributions demonstrate how Web technologies can be used to provide interoperability for smart home systems. In addition to Web technologies and smart homes, the research also contributes to other research domains, such as, Semantic Web and agent technologies.

1.5 Organization of the Dissertation

The main contributions of the research are presented in the publications included in the article based dissertation. This summary presents the interoperability model and reviews the implementations in relation to that model.

The dissertation is organized as follows. This chapter gives an introduction to the research presenting motivation, research questions and methods, and contributions. The following chapter shortly describes the history and background of the building automation domain, then focuses on the smart homes and Web services, and finally introduces different smart home Web service protocols. Chapter 3 presents the smart home interoperability model, describes in detail different interoperability layers, and presents related research on each of them. Chapter 4 describes the software contributions and reviews them from the interoperability point of view. Chapter 5 summarized the current situation on smart home interoperability and discusses the movement toward Internet of Things, Web of Things, and smart homes. Finally, the last chapter concludes the work by summing up the contributions and describing the future work on the smart home interoperability.

2 Background

Smart homes have a long history. According to different definitions, first smart buildings are from the mid-1900s. For example, Mozer (1998) argues that since the mid-1940s, the home automation industry has promised to revolutionize our living environments. Similarly, Levermore (2000) suggests that the earliest ancestor of the BAS was the hard-wired, central system that first appeared in the 1960s and was employed in large buildings.

The history of smart homes is in the building automation, and the future may be in the Internet and the Web. Development has been slow, but recent advancements in communication technology and the popularization of the Web may speed up the process. In this chapter, I first give a short overview of the building automation history, and then concentrate on the smart homes and recent Web technology solutions.

2.1 Building Automation

Several different names with similar meanings are used for building automation systems, for example, Building Management System (BMS), Building Energy Management System (BEMS), Energy Management System (EMS), and Facility Management System (FMS). While having slightly different definitions, they all refer to a similar technological solution. For simplicity, this dissertation refers to all of these as BASs. (Levermore, 2000) (Ihasalo, 2012)

Building Automation (BA) is the objective of a BAS. It considers automation achieved by various systems and devices, *e.g.*, Heating, Ventilating, and Air Conditioning (HVAC), lighting, and security. Energy management has often been the primary justification for installing a BAS (Eyke, 1988). More recently, increased user comfort at reduced operation cost has also been considered as the key driver for the BA market (Kastner *et al.*, 2005).

Industrial automation, as well as information technology has influenced the development of BASs, and recent increased use of BASs is shaped by rapid advances in communications and networks (Levermore, 2000). For example, early network protocols and control techniques were adopted from the industrial automation for the BA (Schickhuber and McCarthy, 1997). Computerization of the BASs has followed the information technology advancements. First computerized BA control systems were introduced in the 1960s (Månsson and McIntyre, 1997). Digital devices with unique addresses have also provided possibility to move from discrete wiring to sharing a common communication path (Eyke, 1988).

From the early 1980s, Direct Digital Control (DDC) has been used in BASs to directly control environmental conditions. DDC is a system where a digital computer is used to automatically control a device. DDC can be considered as one of the early enablers of a modern BAS. Before DDC, control was achieved with electromechanical systems, *e.g.*, air and oil pressure systems. (Eyke, 1988) (Schickhuber and McCarthy, 1997)

Another important factor for the advent of BASs is fieldbus, the network system for the field devices. In a fieldbus system, a control station can communicate with several devices connected to one serial bus. Fieldbus systems were first developed for the industry automation. Later, Networked Control Systems (NCS) opened the market for home and building automation systems. A fieldbus system consists of slave devices, *e.g.*, sensors and actuators, and master devices, *e.g.*, Programmable Logic Controllers (PLCs). There is no direct interoperation between slave devices, as only a master device can control them. Main limitations of the serial fieldbus systems consider network expansion, network topology, and transmission media. Later, large number of different control system protocols and the slow standardization process of those protocols (Felser and Sauter, 2002) led to the development of gateways between fieldbus protocols, and between fieldbus and higher-level information systems. (Schickhuber and McCarthy, 1997)

Building automation system architecture can be thought to consist of several layers based on, *e.g.*, control or communication network. Levermore (2000) presents a four-level control model for BASs. The model consists of field, control, operational, and management levels. Field level devices are connected to unintelligent outstations on the control level. Central stations operate on operational level and have the ability to calculate and make decisions based on the data from the outstations. Management applications operate on the management level and communicate with the central stations via gateways. However, they note that the nature of BAS is becoming more difficult to define, as even the smallest components in the system can have integrated microchips. Kastner *et al.* (2005) presents another BAS model consisting of three levels. Their model includes field, automation, and management levels. Sensors and actuators interact with the physical world on the field level. Automatic control, *i.e.,* logic of the system, takes place on the automation level. In addition to separate control sequences for the individual devices, horizontal communication can occur between different data from the field level. Data from the field level can also be processed here for, *e.g.*, preparing data histories. At the management level, operator can manually access and manage the system via unified user interface.

A modern BAS should provide both energy-efficiency and user comfort. Unified user interface should combine the management of different heterogeneous devices and subsystems. Integrating together these devices is one of the biggest challenges in the building automation today. The problem is that devices often use domain specific, non-standard, proprietary and complex protocols, and communicate over different physical layers.

Using standards, different manufacturers can produce devices that are compatible with each other. Unfortunately, historical reasons and business interests of the manufacturers have led to the situation where we have several competing home and building automation standards. Earlier protocols concentrated on the field level communication. They defined low-level data formats and used specific communication channels. In recent years, new high-level standards, which make use of the general purpose Internet Protocol (IP) network, have appeared (Maile *et al.*, 2007). It has allowed building automation to benefit from available Internet technologies, such as Web services. These new technologies are intended to simplify the high-level integration of different BAS parts with management and enterprise applications.

2.2 Smart Homes

While BASs concentrate mostly on the energy efficiency of large buildings, the focus of home automation is more on living comfort, convenience, and security of residential households. Smart home extends this goal requiring intelligent behavior and interactions between devices and appliances (Mozer, 1998).

The term *smart house* was first used in an official way by the American Association of House Builders in 1984. However, hobbyists already built first wired homes in the early 1960s. This development defined that incorporating interactive technologies makes a smart home smart. (Harper, 2003) However, this is a vague description and accordingly various definitions for *smart home* have later been presented. Oxford Dictionaries (Oxford, 2014) defines a smart home as:

> *"A home equipped with lighting, heating, and electronic devices that can be controlled remotely by smartphone or computer."*

By this definition, a smart home consists of devices that are required to be remotely controllable. Such smart home is, however, still quite simple and might be considered to be not that smart. In scientific literature, Valtchev and Frankov (2002) define a smart home as:

> *"A house or living environment that contains the technology to allow devices and systems to be controlled automatically."*

This definition requires automatic control of the smart home devices. In practice, controlling each other based on some logic, devices realize the interoperation of the smart home. Another, more goal oriented definition by Cook and Das (2007) defines a smart environment as:

> *"One that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment."*

Koskela and Väänänen-Vainio-Mattila (2004) defines a smart home similarly as a space that:

> *"Adjusts its functions to the inhabitants' needs according to the information it collects from the inhabitants, the computational system, and the context."*

Last two definitions consider intelligence as a requirement for a smart home. Similarly to an Intelligent Agent (IA) (Wooldridge and Jennings, 1995), smart home should perceive the environment (and the inhabitants as a part of the environment) through its sensors, and control the environment through its actuators based on some logic.

Since the above-presented definitions differ greatly from each other, there is a need to specify a smart home definition to be used in this dissertation. Thus, a smart home is considered as:

“A home equipped with interoperable devices.”

To understand this definition, the term *interoperability* needs further to be defined in the scope of smart homes. For this, an interoperability model is considered. The model consists of technical, interconnection, syntactic, interoperation, semantic, and intelligence layers. The interoperability model is presented thoroughly in Chapter 3. Now, smart home is simply defined as a home that contains some level of interoperability. Thus, minimally, when two or more devices exhibit technical interoperability, the system fulfills the definition. As can be noted, the standpoint of this smart home definition sets high importance for the interoperability. The reason for this is that integration, interaction, and intelligent cooperation of the system components are considered as some of the most important challenges in the smart home research today.

Aldrich (2003) proposed a classification of five hierarchical classes of smart home: homes which contain intelligent objects; home which contain intelligent, communicating objects; connected homes; learning homes; and attentive homes. Three first classes clearly represent different levels of interoperability from independent and separate devices to integrated systems and to homes with local network connected to the Internet. Learning homes represent the ability to learn from the recognized patters from the signals given by the user and various sensors. Finally, attentive homes represent context aware systems that require understanding of the semantics of the information. Similarly to our definition for the smart home, this classification emphasizes the importance of integration, communication and intelligent cooperation. In addition to these, other important subjects in the smart home research and development consider, *e.g.*, safety, security, authentication, authorization, usability, user centric design, and different application scenarios.

Although the first wired homes were built over 50 years ago, and the term smart house was defined decades ago, smart homes have not become reality for the common people. Some reasons for this are high cost of the systems, lack of required cabling, lack of common protocol, and poor design and usability of the products. (Aldrich, 2003) Today, the required infrastructural support for the smart home systems exists. Wired and wireless Local Area Network (LAN) cablings are common in modern buildings and can be used as a backbone network for the smart home. This can significantly decrease the initial investment needed for installing a system.

The aim of IoT is to use commonly available Internet cabling as an infrastructure for connecting things together. Numerous low-power network protocols have recently been developed for this purpose. Unfortunately, the systems are generally incompatible with each other. As an effort for solving this problem, several service platforms have been developed. However, service platforms are again commonly incompatible with each other and have not succeeded to gain wider acceptance. (Guinard, 2011)

OSGi is a Java based middleware for the integration of heterogeneous systems together. It is based on a component model that provides modularity and dynamicity. Software components, called bundles, can be installed and managed without rebooting the system. (OSGi, 2007) OSGi has been used, *e.g.*, to enable smart environment by integrating heterogeneous devices using bundles (Lee *et al.*, 2003), and with mobile agent technology, SOAP Web services, and several distributed OSGi systems together for providing service oriented smart home architecture (Wu *et al.*, 2007). However, for embedded devices in smart homes, OSGi has been considered as too heavy and complex platform, as the same benefits can be gained using Web services, which are operating system and programming language independent (Guinard, 2011) (García-Valls *et al.*, 2010). Jini (Arnold *et al.*, 1999), sometimes called as Apache River, is another Java based approach for building distributed systems. It uses Java Remote Procedure Call (RPC) for the communication between the client and the service. Similarly to OSGi, Jini supports basic interconnectivity but is unable to provide more complex and intelligent functionalities (Kaldeli *et al.*, 2013).

In the European market, KNX presents one of the traditional standards for the intelligent buildings. It was created in a convergence of European Installation Bus (EIB or Instabus), European Home Systems Protocol (EHS), and BatiBUS standards. It has several options for the physical communication media from twisted pair, radio and power-line to infrared and Ethernet. The main medium for device-to-device communication is twister pair, but through EIBnet/IP tunneling also remote maintenance is made possible. (Kastner *et al.*, 2005) In 2015, KNX Association had 367 members from 37 countries, almost 47000 partners from 133 countries, and the number of KNX certified devices was over 7000. In 2012, KNX had 74% share in European smart home market. (KNX, 2015) However, in practice, only supported medium in KNX device communication is twisted pair (Alam *et al.*, 2012). Regarding current trend of Web enabling devices for the WoT, KNX can thus be considered outdated. Still, using a middleware solution, KNX devices can be connected to modern Web based smart home systems.

As mentioned before, a plethora of different standards have been developed for the building and home automation device control. Also, independent solutions presented in research publications often propose new protocols that are never truly adopted by the community outside of that particular project scope. For example, Superior Home Automation Protocol (SHAP) in DOMOSEC home automation system (Zamora-Izquierdo *et al.*, 2010) might be a suitable protocol in some scenarios, but by introducing one more protocol for the domain it only manages to further complicate the interoperability problem.

As standardization of communication protocols has so far failed to popularize smart homes, I believe that an additional catalyst is needed to reach faster the tipping point. Ecosystems, maintained by large organizations or companies, are probable breeding grounds for the popularization of smart home devices. As commonly customers already own several devices from major mobile technology companies, smart home device ecosystems from Apple, Google, and Microsoft can be expected to gain popularity in the future.

In research, smart home projects have commonly concentrated on providing solutions to certain use cases. These use cases include, *e.g.*, energy efficiency (Han and Lim, 2010) (Jahn *et al.*, 2010), and health related issues, such as senior living, telemedicine, eHealth, telehomecare, home-based eHealt, and assistive devices (Chan *et al.*, 2008). However, increased standard of living, available network infrastructure, and prevalence of mobile devices can also motivate people to consider smart home systems for easily increasing their living comfort. In a long-term user study on living in a smart home, after a few weeks the tenants adopted the technology well and reported good user experiences (Kaila *et al.*, 2008).

In addition to popular communication protocol development, research on other aspects, *e.g.*, user interface development for smart homes has also been presented in the literature. Lazovik et al. (2009) described a Smart Home Visualization and Simulation (ViSi) environment based on Google SketchUp. The system is mainly targeted on reducing testing costs by replacing actual devices with virtual ones. However, they also connected one physical SOAP based device to the system. More recently, the system has been extended to use OSGi platform as a middleware and UPnP as device control protocol (Kaldeli et al., 2013). ViSi is used for testing and evaluation of the system, while development toward usage of actual devices is currently underway. Besides offering a simulation and visualization platform, ideas from ViSi could be exploited in smart home user interface development.

Popularity of social networking applications with rapidly aging population have motivated research to consider social inclusion of the elderly with their families and caregivers through social networks. For example, Pensas et al. (2013) presented a social network application framework for elders with support for different user groups: elders, family members, and third parties. Specialized touch based user interface was developed for the elderly, while others could interact using PC and mobile device interfaces (Kivimäki et al., 2013).

Aging population has also motivated research on new types of user interfaces from the perspectives of senior citizens. Using fixed cameras and projectors, remote assistance in cooking can be given for demented elderly using visual prompts (Ikeda et al., 2011). Similarly, wearable technology, including camera, picoprojector and accelerometer, can provide assistance for daily activities of senior citizens (Pulli et al., 2012). As the technology is always carried with the user, projections can be made on any surface, and the accelerometer can recognize tap actions made by the user to interact with the user interface. For better usability, Smart Glasses project integrates microphone, speaker, camera, and small LED-indicators into eyeglasses

(Firouzian et al., 2015) (Lääkkö et al., 2014). Such device facilitates remote assistance for navigational tasks for senior citizens. Tervonen et al. (2013) also proposed providing navigation aid using body worn laser device. Using the device, people suffering from dementia can navigate in both indoor and outdoor environments with the help of laser-projected arrows. Additionally, the device can be used to help identifying real objects with a laser pointer.

2.3 Web Services

Generally a Web service can refer to any service available on the Web. One basic form of this kind of a service is a web page. However, software developers usually interpret Web service as something else. For them, a Web service is a software application providing functionality to other applications through a Web protocol interface. Accordingly, Web services have been used to develop distributed computing environments and for integrating existing separate systems together. Web services provide interoperability for heterogeneous computing devices. The technology is independent of any hardware, operating system, or application.

There are various definitions for a Web service. The Web Services Architecture (WSA) by World Wide Web Consortium (W3C) is an interoperability architecture that provides a common implementation-free Web service definition (Booth *et al.*, 2004). It defines that:

> *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*

While not specifying how Web services are implemented, the definition refers to certain techniques. It requires Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) to be used. Additionally, it suggests the use of HyperText Transfer Protocol (HTTP) and Extensible Markup Language (XML). Earlier version of the WSA had a more open definition and did not bind Web services to SOAP and WSDL techniques (Booth *et al.*, 2003).

WS-* technology stack, sometimes called Big Web services, is a family of technology standards for developing Web services. It refers to technologies and standards, such as SOAP, WSDL, WS-Addressing, and WS-Security (Weerawarana *et al.*, 2005) (Zeeb *et al.*, 2007). In addition to Web services complying with the W3C definition and WS-* standards, more lightweight RESTful Web services exist that typically do not use SOAP, and are rarely described with WSDL. Commonly, SOAP based Web services have been used by enterprises, while RESTful Web services and Web APIs have become more popular for developing basic and *ad hoc* integration solutions. (Benslimane *et al.*, 2008) (Pautasso *et al.*, 2008)

SOAP based Web services have found their place in the integration of intra- and inter-organization enterprise systems. Wide range of standards and tools are able to fulfill the requirements of such systems. At the same time, protocol stack has become relatively complicated and a part of developers have sought alternatives for developing more lightweight Web services. As typically implemented using only HTTP without additional application layer protocols, RESTful Web services provide this type of simple solution. In comparison, WS-* technologies use Web as a transport layer instead of treating it as an application architecture, like WoT. Naturally, both WS-* and RESTful Web services have their strengths and are a right technology of choice for some situations. For example, RESTful Web services are more loosely coupled and thus exhibit better agility for adapting to changes. Respectively, being more tightly coupled, WS-* technologies can provide better performance optimization. (Pautasso and Wilde, 2009) (Wilde, 2007)

REST (Fielding, 2000) is an architectural style for developing large-scale distributed hypermedia systems. HTTP application protocol is the most commonly known implementation of the REST style, and is thus often confusingly used as synonymous to REST. (Pautasso *et al.*, 2008) Fielding, as one of the principal authors of the HTTP 1.0 specification (Berners-Lee *et al.*, 1996) and as the main author of the HTTP 1.1 specification (Fielding *et al.*, 1999), specified several constraints for guiding the development of hypermedia systems according to REST style (Fielding, 2000). These constraints are aimed to provide, *e.g.*, scalability, generality, modularity, and security (Feng *et al.*, 2009).

SOA is a paradigm, where services are considered as the main system elements (Zhang and Zhang, 2010). Web services instead, are now the most common implementation of an SOA. In an SOA, applications can be accessed through a common programming interface, enabling existing IT infrastructure to be more easily reused, replaced, and modernized. As a concept, an SOA is nothing new. For example, J2EE and Common Object Request Broker Architecture (CORBA) have separated software interface from the implementation supporting service definitions. However, the strength of an SOA with Web services is a more complete separation of the service interface from the execution environment, pervasiveness of Web services, simplicity, and platform neutrality. (Newcomer and Lomow, 2004) (Santos and Carreira, 2014)

2.4 Smart Home Web Services

SOA provides four clear benefits for the system development: reuse, efficiency, loose technology coupling, and division of responsibility. All of these can be beneficial for the development of smart home systems. Device and system services can be reused in multiple applications. New services can be quickly and easily composed from old and new services without concentrating on the underlying implementation details. Devices and systems can run on their ideal executing environments and cooperate with the control logic units using agreed standards and protocols. To realize these benefits, Web services are deployed as an ideal platform for SOA. (Newcomer and Lomow, 2004)

Kindberg *et al.* (2002) were one of the first to propose a comprehensive approach of building on top of the Web for bridging the physical and virtual worlds. The idea later matured to be now known as the Web of Things (WoT) (Wilde, 2007) (Guinard, 2011). In the WoT, people, places, and things have virtual representations on the Web, and users can interact with them using REST principles (Kaldeli *et al.*, 2013). The WoT provides an application layer on top of the IoT. To become popular, the concept may still need additional standards, *e.g.*, for agreed descriptions of the things and their control. In 2014, the W3C showed interest in WoT by organizing the W3C Workshop on the Web of Things to discuss different topics regarding the WoT, including, *e.g.*, core technologies, challenges, semantics, security, and privacy. Related to interoperability, the workshop report states that there was a broad agreement among the participants about the problems of fragmentation of products developed in isolation, plethora of IoT protocols, and lack of shared approach to services (Heuer and Raggett, 2014).

Continental Automated Buildings Association (CABA) started planning of Open Building Information eXchange (OBIX) in 2003. Currently, the standardization process is handled by the Organization for the Advancement of Structured Information Standards (OASIS). Development on smart home Web services based solutions presented in this dissertation is originally based on OBIX Version 1.0 Committee Specification 01 (Frank, 2006). Updated OBIX Version 1.1 Committee Specification Draft 02 (Gemmil, 2013) has been separated into multiple documents. In addition to the main specification, separate documents for common encodings, HTTP binding, SOAP binding, and WebSocket binding are published. The most significant changes include introduction of delete request, binary and JSON encodings, WebSocket binding, and server pushed watches. One of the strengths of OBIX is that it lacks the burden from legacy and proprietary systems. It has been designed for the Internet from the beginning.

In addition to OBIX, other similar proposals have been presented. Some of the existing lower level automation system standards have been extended to support Web services. For example, the popular communication standard, Building Automation and Control Networks (BACnet), now includes a definition of generic Web services interface for automation data independent from the underlying protocol (Neilson *et al.*, 2012). Also, The Object Linking and Embedding for Process Control (OPC) Foundation has published an OPC Unified Architecture (OPC UA) specification, which describes Web services as one of the means for communication (OPC Foundation, 2012). OPC concentrates mostly on the industrial automation and OPC UA is designed for connecting industrial devices to control and monitoring applications (Hadlich, 2006). While the standard addresses primarily control systems for industrial processes, it can also be applied for building automation (Fernbach *et al.*, 2011).

OBIX basically defines an XML data format to represent the information and a RESTful Web services interface to access it. However, real-time communication can be an important requirement in some smart home use cases. In OBIX, it can be achieved using either server pushed watches, or in a more elegant way using OBIX WebSocket binding. In BACnet/WS, client sets a callback URI for the server to send updates. Thus, real-time updates are only achieved when client implements also server functionality.

In home networking protocols, UPnP is an exception. It has managed to obtain relatively high level of support from different device manufacturers, increasing the number of devices supporting it. It is promoted by the UPnP Forum including over a thousand companies. On the transport layer, it uses both TCP and UDP protocols, and on the application layer HTTP and XML. However, similarly to Jini, it is targeted on basic interoperability, including device discovery, communication, and providing data and services in a home environment. Support for intelligence and enterprise applications is outside the scope (Kaldeli *et al.*, 2013). Supporting UPnP is in any case a good starting point for a home gateway system entering the market. Research on the area has been done for, *e.g.*, UPnP home middleware (Kim *et al.*, 2002), sensor network management using UPnP (Song *et al.*, 2005), and converging UPnP with other middleware protocols in single home, interconnected homes, and Internet scale scenarios (Kleimola and Vuorimaa, 2008).

Digital Living Network Alliance (DLNA) organization has similarly defined guidelines, *e.g.*, (DLNA, 2014), for sharing digital media between multimedia devices at home. The guidelines specify a set of existing standards to be used in a restricted way for compatibility between devices. This is why DLNA is also partly based on UPnP, as it uses UPnP Device Control Protocol Framework for device discovery with Simple Service Discovery Protocol, and UPnP Audio/Video (AV) specification for media controls (Bergwik, 2014). By restricting existing standards, DLNA can better ensure the interoperability between different devices. Parks Associates estimated that by 2015 almost 4 billion DLNA certified devices have been sold (Parks Associates, 2013). While DLNA is targeted only for home network environments, research has been done for extending it further. Sharing and accessing DLNA device content over the Internet using proxies have been suggested by, *e.g.*, (Kim *et al.*, 2007) and (Venkitaraman, 2008). Similarly, two-way sharing of content between P2P networks and DLNA using OSGi has been suggested by (Lai *et al.*, 2010).

Devices Profile for Web Services (DPWS) (Driscoll and Mensch, 2009) is a specification designed to provide common requirements for devices to promote interoperability between resource-constrained Web service implementations. It defines basic requirements for security, dynamic discovery, description, and event subscription based on WS-* technologies. Zeeb *et al.* (2007) analyzed DPWS and presented a toolkit for developing devices conforming to it. They pointed out a couple of issues in DPWS in need for improvements. For example, lack of clarity in messaging section can lead to interoperability issues, and semantics of device types are not clearly defined. De Souza *et al.* (2008) presented SOCRADES architecture for business integration. In their work, DPWS is used with OPC UA to allow the middleware to connect industrial manufacturing devices with the high-level enterprise applications such as Enterprise Resource Planning (ERP) systems.

DPWS requires a service to at least support push delivery mode with WS-Eventing. WS-Eventing specification (Box *et al.*, 2006) describes a method for Web services to subscribe to an event source to receive instant notifications. However, it does not describe a communication technique for implementing a server push method between a client and a server. This implies that while DPWS is targeted for embedded devices with restricted computing capabilities, devices need to implement server side functionalities to conform to the specification. Such solution is described by, *e.g.*, Liu *et al.* (2006). In contrast to this, in Chapter 4.5 and Chapter 4.7 a novel solution using WebSockets to allow clients to only implement client side functionalities and still receive server-push notifications in real-time is presented. Similarly, also Guinard (2011) suggests extending REST with server-push techniques for the WoT.

3 Smart Home Interoperability

Merriam-Webster Encyclopedia[1] defines interoperability as *the ability of a system to work with or use the parts or equipment of another system.* Smart home interoperability can thus be similarly defined as:

> *"The ability of smart home devices to work with or use the services or equipment of other smart home devices."*

This definition brings up questions regarding the amount of interoperability that a certain system entails. Depending on the system, different devices may, *e.g.*, be connected to the same control system, share information directly with each other, or build a knowledge base and reason about the environment using goal-directed behavior.

To categorize different features contributing to interoperability, a layered interoperability model is presented (Figure 2). Earlier, interoperability models have been defined by, *e.g.*, The European Telecommunications Standards Institute (ETSI) (van der Veer and Wiles, 2008) and European Interoperability Framework (EIF) (EIF, 2004). The model presented here extends commonly known three interoperability layers from these models with three additional layers that focus on realizing the potential of the known capabilities with corresponding derived functionalities. The goal of a smart home system should be automation that can be achieved with enhanced cooperation between devices. This motivated the extension of the common model with functional layers that represent operational levels that can be achieved. Commonly presented interoperability layers are technical, syntactic, and semantic layers. On the bottom, technical interoperability concentrates on establishing a communication channel between the systems, commonly implemented as a common network protocol. In the middle, syntactic interoperability increases the interoperability by introducing a

[1] Merriam-Webster Online: Dictionary and Thesaurus - http://www.merriam-webster.com/

common data format for the messaging. On top, in semantic interoperability, data is presented semantically, categorized using context ontologies, and linked to other semantic data.

Functional layers respectively present operations that can be built on top of these commonly presented interoperability layers. Interconnection layer represents the lowest level of functional interoperability. Separate systems can communicate but the content of the messages might not be correctly interpreted by the receiving system, or the communication could just happen in one integrated system. Interoperation layer considers inter-device logic that enables actions to be executed in one component based on the information from another component. Intelligence layer provides services for controlling the smart home in more sophisticated fashion, including, *e.g.*, adaptive logic for automatizing the smart home behavior and programming.

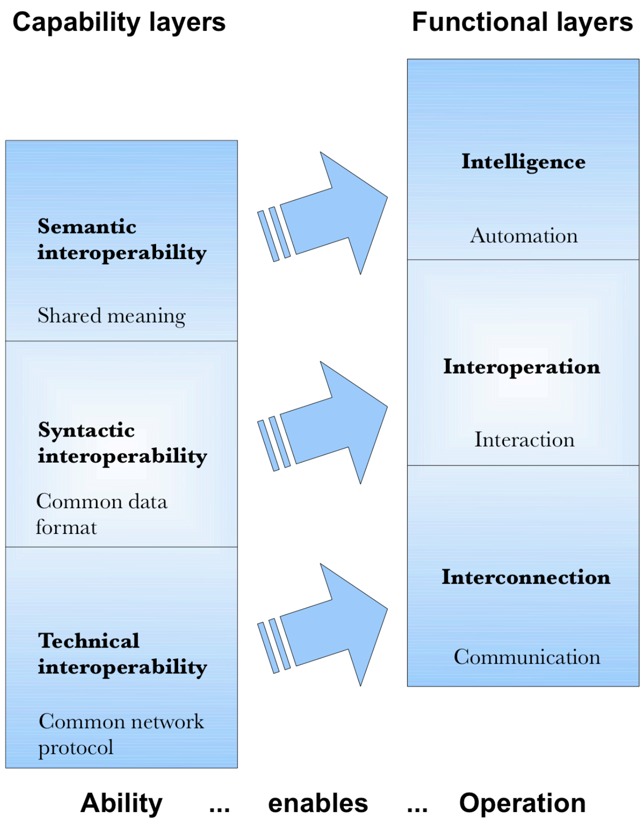

Figure 2: The interoperability model consists of six layers. Commonly known capability layers enable operations on functional layers.

The model is built to depict different levels of the smart home interoperability. However, naming of the layers and the corresponding descriptions are deliberately kept as generic as possible, so that the model can be generalized to describe interoperability also in other technical domains. Figure 3 depicts the interoperability model side by side with the well-known Open Systems Interconnection (OSI) model (Zimmermann, 1980) from the International Organization for Standardization (ISO) and Internet Protocol Suite, *i.e.*, TCP/IP model (Leiner *et al.*, 1985) (Braden, 1989). Precise and straight comparison between the layers of the interoperability model and the two other models is, however, not completely meaningful as the models characterize different things. OSI and TCP/IP models depict the internal functions of communication systems, while interoperability model additionally covers other capability and operational areas. In practice, OSI and TCP/IP models can be understood as models for standard protocol stacks that enable interconnection and syntactic interoperability, while interoperability model provides a broader view on how well systems understand each other and what functionality can be built upon them.

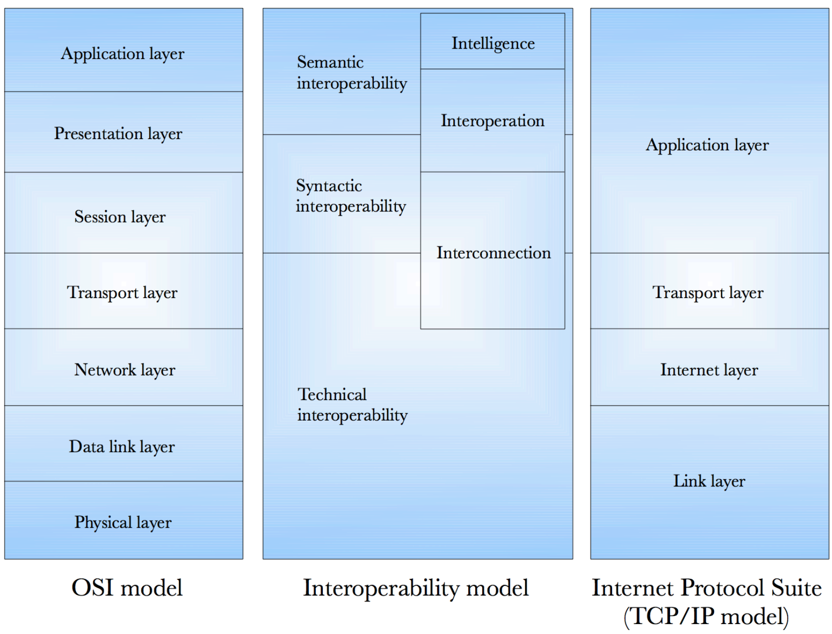

Figure 3: OSI, TCP/IP, and interoperability models presented side by side.

A similar model to the one presented here, a layered conceptual architecture for Semantic Smart Homes (SSH) was proposed by Chen *et al.*

(2009). They considered physical layer as sensor and actuator hardware, data layer as data storage, semantic layer for homogenization of the data, intelligent service layer for providing advanced processing and presentation capabilities and services, and application layer for providing the high level tools for finally enabling the smart home scenarios. The model follows a generic smart home model with two additional layers added for the semantic aspect.

Another generic layered model, the Context Stack, proposed by Zhang *et al.* (2005) is a general guideline for context-aware systems. Their model consists of acquisition, representation, aggregation, interpretation, and utilization layers. The focus in their model is on context-awareness.

Cook and Das (2007) also presented a model of smart environment components in their survey on state of the art on smart environments. Their model consisted of four layers: physical, communication, information, and decision. They present the work done for each of the layers. However, presented research does not consider any work on semantic technologies, MASs, or the use of Web technologies.

Another, three-layer interoperability model was described by Perumal *et al.* (2008). From the bottom-up, their low-level model consists of basic connectivity interoperability, network interoperability, and syntactic interoperability layers. Basic connectivity considers the physical cabling, network considers the transport, *e.g.,* TCP/IP, and syntactic interoperability considers the application and presentation, *e.g.,* HTTP and SOAP. In contrast, our model takes into account higher-level interoperability layers, and considers technical interoperability to include both physical cabling and the network protocol.

Guinard (2011) presented a layered architecture for the WoT in his dissertation. The architecture consists of four layers: accessibility, findability, sharing, and composition. Accessibility layer concentrates on providing access to smart things, publishing them as web pages, and modeling functionality and services using RESTful principles. Findability layer, in turn, considers searching and finding services, and adding semantic annotations for search engines. Sharing layer utilizes social networks as a sharing mechanism for smart things. Finally, composition layer facilitates the creation of composite applications and physical mashups manually, using a widget based portal, and using an end-user mashup editor. In his work, the nature of Web based devices has been addressed thoroughly based on these layers. In contrast to the challenges he tackled, our research takes a different point of view by approaching interoperability based on data representation capabilities and functional properties. Still, both aspects should be kept in mind when developing Web based device environments. Smart home environment requires solutions from all of the WoT architecture layers, and similarly WoT needs to consider ways to enhance interoperability.

Kamilaris (2012) presented work on enabling smart homes using Web technologies in his dissertation. The work considers many common issues in smart home system development, *e.g.*, discovery and sharing, and also discusses briefly about security. He presented three-layer application framework architecture for smart homes, composed of device, control, and presentation layers. The device layer is responsible of device drivers connecting embedded devices to the system, and provides a uniform interface for accessing the device data. Control layer handles the core system functions. It initializes the system and controls the whole framework. The presentation layer runs a Web server and provides the device data and services through REST interface.

As our smart home definition indicates, this dissertation and the presented model consider smart home from the interoperability point of view. Only technical interoperability is required for the system to be considered as a smart home. However, this is only a technical definition and the smartness of such a system is very limited. The three functional layers form their own stack, as they are optional when the capability stack is considered. This is, because none of the layers of the capability stack requires functional interoperability. The functional stack however, has the same requirement as the capability stack, that the lower layer is always a prerequisite for the upper layer. Following subchapters describe the interoperability layers in detail.

3.1 Technical Interoperability

ETSI (van der Veer and Wiles, 2008) describes technical interoperability as:

> *"Usually associated with hardware/software components, systems and platforms that enable machine-to-machine communication to take place. This kind of interoperability is often centered on*

(communication) protocols and the infrastructure needed for those protocols to operate."

Technical interoperability is the lowest level of interoperability that a system needs to hold to be even considered interoperable at all. It is the compatibility between different subsystems. In practice, it comprises a combination of hardware and software that unifies the physical communication channel and the low-level protocol structure. Being widely accepted, TCP/IP is considered as the solution for technical interoperability. Thus, the state of the technical interoperability can generally be considered as fully developed. (Kubicek *et al.*, 2011)

In smart homes, TCP/IP presents a viable option for implementing technical interoperability. However, it only provides half of the solution *per se*. Traditionally, home automation devices have lacked both hardware and software capabilities for TCP/IP communication. Even though the software support for TCP/IP would exist, devices also need to provide physical connectivity, let it be either wired or wireless. Thus, for legacy home automation devices technical interoperability usually requires the addition of appropriate hardware solutions. In some cases, the device may provide a different kind of interface without the support for TCP/IP. In such cases, protocol converters or gateways may be used to add this support.

The power of technical interoperability is often exaggerated. It is purely a capability layer to enable the exchange of messages between subsystems. Message content structure and interpretation are disregarded (Kubicek *et al.*, 2011). Thus, a message exchange of two technically interoperable systems is comparable to a conversation between two people from different countries speaking their own languages to each other. Messages are delivered but hardly understood.

While technical interoperability alone does not enable any useful functionality for applications, it is an important layer when thinking interoperability as a whole. Technical interoperability is required before any other type of interoperability can occur. It is usually also the only layer that requires hardware.

3.2 Interconnection

Interconnection is the first functional interoperability layer. It is enabled only by technical interoperability, and can be seen as more abstract than other functional layers. On this layer, two or more subsystems can simply communicate with each other. As the structure of the message content is not agreed between the subsystems on this layer, interconnection merely builds basis for better interoperability on higher layers.

When two or more separate subsystems are integrated on the code level, they can also be considered interconnected. In this case, communication between the subsystems takes place within the program code, *e.g.*, through function calls. However, as the structure of such function calls and the argument types are explicitly defined, this is considered as higher-level interoperability.

However, in addition to the message exchange functionality, there are other advantages of interconnection, even though the subsystems cannot understand each other. Interconnected devices can utilize the same network infrastructure available in the environment. This can already radically lower the costs of installing different systems in a building, as only one type of cabling is needed. Due to the success of the Internet, wireless and wired TCP/IP networks are currently available in most modern buildings.

Available network infrastructure and TCP/IP together present another advantage of interconnection. While subsystems fail to understand each other, subsystem services can be semi-integrated into one unified control center. For example, independent user interfaces for the services can be presented as widgets on a desktop computer application.

3.3 Syntactic Interoperability

ETSI (van der Veer and Wiles, 2008) describes syntactic interoperability as:

> *"Usually associated with data formats. Certainly, the messages transferred by communication protocols need to have a well-defined syntax and encoding, even if it is only in the form of bit-tables. However, many protocols carry data or content, and this can be represented using high-level transfer syntaxes such as HTML, XML or ASN.1."*

While technical interoperability is responsible for transport protocols, syntactic interoperability concerns a common data format used to describe the messages. It defines the syntax for the content. Such syntax can form, *e.g.*, a complex and flexible high-level language, or a primitive and specific bit-table format for a narrower purpose. Syntactic interoperability layer provides the capability of two or more systems to interpret the message content of the exchanged data. It also enables better integration of services into one unified user interface on a central control station.

There are three common methods for implementing syntactic interoperability. It can occur directly between two or more devices either using their native protocol, or via protocol converters. In a more sophisticated fashion, it can be implemented in a middleware through protocol adapters that abstract the underlying diverse protocols and unify the data according to a common model.

In building automation, development has traditionally concentrated mostly on low-level field devices. Manufacturers have introduced their own diverse communication protocols and device specific data formats. This has caused numerous different protocols and data formats to be developed without concerning the interoperability between devices and systems from different manufacturers or even between different products from the same manufacturer. The lack of interoperability has slowed further development of BA and HA industry. To solve this, a standardization process of automation protocols was started. If manufacturers used the same protocol, it would be possible, for instance, to replace a broken or an old device in a system with a similar device of another brand. Obviously, each company tried to promote the protocol they already adopted to become a standard. In the industrial automation, similar disputes between supporters of different protocols sometimes even grew into political issues (Felser and Sauter, 2002). No final agreement has been made and, as a result, today there are a number of competing national and international standards of field level protocols for BASs (BACnet, KNX, LonWorks, X10, *etc.*).

The variety of standards has left the interoperability issue unresolved. A modern BAS often has to combine several subsystems talking different protocols. This requires using bridges in order to convert messages from one protocol into another. It results in complex automation network topologies and makes the integration process for each BAS to be a cumbersome task.

In addition, protocol conversion has other weaknesses, as complete conversion is not always possible and some information may be lost in the process (Kaila *et al.*, 2008).

Numerous bridges have been developed for connecting systems operating with different protocols, *e.g.,* KNX-ZigBee (Lee and Hong, 2009), LonWorks-UPnP (Chemishkian and Lund, 2004), and KNX-OBIX (Neugschwandtner *et al.*, 2007). More recent solution is to use a gateway middleware with protocol adapters to convert heterogeneous device protocols into a unified data format. The benefit of such gateway is that each protocol needs to be converted only into the common format and back, resulting in a linearly growing number (2n) of adapters. This is tolerable, compared to the polynomial growth in number (n(n-1)) of converters needed for a system with bi-directional bridges between each protocol.

A number of integration platforms have been developed to address the incompatibility problem caused by the excessive amount of different protocols. One such system is a SOAP based middleware for integration of various systems of the building (Perumal *et al.*, 2010). In this solution, a central building management server using a set of event-based triggers coordinates the work of subsystems.

More complex architecture for a BAS based on OPC standard is proposed in (Wang *et al.*, 2007). It consists of two integration layers. On the low level, different subsystems are connected to the OPC server through OPC gateways and LAN is used as a backbone. The OPC server, in turn, provides a Web services interface for higher-level integration. On the high level, Building Management Server (BMS) uses this Web services interface to control remotely several OPC sub-networks over Internet. As Web services suppose classic client-server communication where only a client can initiate a connection, authors introduce ”mini-Web services” at the BMS side. These Web services are used by OPC servers to deliver event messages on demand. Hence, both BMS and OPC server in this architecture act as a client and a server simultaneously. This solution solves the problem of pushing data from end devices to BMS, but as a drawback the whole system becomes more complicated.

Similar research on integration platforms has been done for the home automation domain. For instance, in an expandable integration framework (Tokunaga *et al.*, 2002), each type of home appliances is connected to the common network with help of Virtual Service Gateways (VSG). These gateways convert device specific protocols into a common language, based on SOAP. Each gateway can publish device service (act as a server) and use services from other gateways (work as a client). Thus, home appliances can communicate directly with each other. There is also a separate service repository inside the network, which is used to detect available devices. In the prototype it was implemented as UDDI registry (Bellwood *et al.*, 2002). The proposed architecture is a less centralized solution, as there is no single management server, which controls the whole system. Instead, any node in the network can provide a user interface to the system.

Another architectural solution, DomoNet, is presented in (Miori *et al.*, 2006) and its further development is discussed in (Miori *et al.*, 2010). The difference with previous model is that device gateways (called TechManagers) do not provide Web services by themselves, but publish them to a separate server, called domoNetWS. TechManagers provide an interface, which is used by domoNetWS in order to forward calls of corresponding Web services methods. It means that each TechManager should implement server functionality and have a known IP address.

DomoNet XML language, DomoML, is also used in (Witte and Telkamp, 2006). In this project, home automation system is connected to the Internet with a residential gateway to enable remote control, for example, using a mobile device. The residential gateway converts DomoML messages into another protocol. It means that messages are converted twice while travelling from an automation device to a user application: First, from a device specific protocol into DomoML, and then from DomoML into the language understandable by the UI application.

Latest tendency in the integration of devices is to use IP networks not only as a backbone, but also to communicate directly with end devices (Guinard and Trifa, 2009). Previously it was too expensive to implement the whole IP protocol stack inside each device, but as microcontrollers become smarter and cheaper, more and more devices appear, which are able to speak to the Internet (Knauth *et al.*, 2008). With the advent of the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) (Shelby and Bormann, 2009) standard, it became possible to integrate an IP interface even into a small battery-powered wireless sensor (Hui and Culler, 2008). Therefore, future smart home system can be a simple IP network of intelligent devices with no intermediate protocols. Such system will become a part of the Internet, benefiting from all available Web technologies. For instance, a user interface for controlling the whole system can be created as a Web mashup of services provided by each device (Guinard and Trifa, 2009). However, while consumer electronics are increasingly getting Internet connectivity, unfortunately the products still commonly fail to exhibit syntactic interoperability as they use proprietary interfaces (Guinard *et al.*, 2010b).

OSGi (2007) framework provides a complete and dynamic component model allowing remote installation, upgrade, and management of independent components without requiring a reboot. It runs in a single Java Virtual Machine (JVM), making the system more efficient but also less tolerant for failures. OSGi is a modularity framework, as it enforces better modularization of the software. It has been used by, *e.g.,* smart phone industry and car manufacturers.

Wu *et al.* (2007) presented a service-oriented smart home architecture based OSGi and mobile agent technology. They identified the problems in smart homes caused by centralized architecture and dynamic nature of the configuration, and the need of proactivity in the system. As a solution, they distribute the system over several OSGi platforms, use SOAP Web services, and mobile agents in their system. Despite of the usage of mobile agents and a distributed architecture, they do not use any common multi-agent system, and leave semantic interoperability as future work.

System integration in smart homes can cover domains outside the home. For example, Vuorimaa *et al.* (2012) presented assistive Active Life Home portal for home care. The system integrates Web service based assistive devices with high level Personal Health Record (PHR) and user interfaces for various different systems into one portal.

3.4 Interoperation

Interoperation provides interaction for the system. By interoperating, components can perform operations toward each other. While syntactic interoperability provides an abstraction layer for the heterogeneous subsystems by unifying different data representations, interoperation layer provides functional logic for the subsystems to actually interoperate using this unified data. The key characteristic of this interoperability layer is an action that happens based on perceived data.

It is common that the interaction with the smart home system is limited to the human interaction with the devices, *i.e.*, to user interfaces. Here, the term interoperation is used to cover both human-machine interaction and especially machine-to-machine communication. However, besides communication, interoperation needs the logic that makes the communication to actually happen. Interoperation is one of the most important interoperability layers, as it can provide inter-device functionality that is perceived as automation.

Perumal *et al.* (2014) have developed an Event-Condition-Action (ECA) control mechanism for a SOAP based home and building automation system. In their SOA based system, both control of the integrated devices and the management of the rule engine are possible with SOAP messages. The system is comparable with our rule engine presented in Publication IV, having the main difference in the use of Big Web services instead of our RESTful approach.

Kaldeli *et al.* (2010) presented a home system based on three architecture layers. Pervasive layer uses OSGi to handle the communication with heterogeneous devices, and user layer provides the user interface for the home. The most interesting layer is named composition layer, and is responsible for coordinating the device states and the functioning of the system. It consists of, *e.g.*, composition, rule engine, and orchestration modules. The functioning of the composition module is based on AI planning, and the orchestration module controls the execution. Rule engine perceives changes in the home environment and notifies the composition module accordingly when required. The system thus provides interoperations between the home devices, and probably also intelligence. Details of the communication between the modules in the composition layer are not presented. Unfortunately, the rule engine interface for rule creation and management is thus not described. For better interoperability between different control systems, we propose that also the management of the control systems, *e.g.*, rule management, should be exposed as RESTful Web APIs.

In the Web 2.0, mashups are user-generated composite applications that combine data from several online sources to create new richer service that displays the mixed data in one view (Guinard *et al.*, 2009). Similarly, services mashups aim to develop modern Web applications based on simple end-user service compositions (Benslimane *et al.*, 2008). While mashups and services mashups seem relevant considering interoperation on the Web, they usually concentrate only on combining data from different sources, instead of providing a possibility to program real-time logic for the Web services. Extending the concept, physical mashups have been proposed for similar purposes in the WoT context. In addition to concentrating on combining existing services, also functional real-time applications have been proposed (Guinard *et al.*, 2009) (Guinard, 2011).

3.5 Semantic Interoperability

EIF (EIF, 2004) describes semantic interoperability as:

> *"Concerned with ensuring that the precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose. Semantic interoperability enables systems to combine received information with other information resources and to process it in a meaningful manner."*

Semantic interoperability layer concerns the meaning of the data and sharing of that meaning. Adding semantics to the data, information is created (Kubicek *et al.*, 2011). In practice, on this layer data can be presented semantically, categorized using context ontologies, and linked to other semantic data. In addition to the use of standard Semantic Web technologies, other techniques are also considered. In addition to syntax, low-level device protocols and Web based data formats can also provide support for describing semantics. In simple terms, any method adding understanding of the meaning of the data contributes to the semantic interoperability. However, the benefits of the semantics are primarily gained when building intelligence that exploits this information. Still, semantics can also be used, *e.g.*, to improve findability of device services for search engines (Guinard *et al.*, 2010a).

Separation of syntax and semantics in the context of computing has been done long before the birth of the Semantic Web concept. For example, already forty years ago Woods (1975) presented many of the ideas that are now applied in the Semantic Web, described syntax as a form of describing things, and semantics as relations between things and what they mean.

Berners-Lee *et al.* (2001) have predicted that the future of home automation exploits Semantic Web technologies. They state that current techniques for connecting devices and services, such as UPnP and Jini, only concentrate on the syntactic level, while Semantic Web allows shared understanding with the exchange of ontologies. Kubicek *et al.* (2011) say that only if the semantics are defined and shared, data from one system can be automatically recognized and processed by another system.

Several formats have been proposed for adding semantic information to the Web. The most widely used, *de facto* format among Semantic Web developers is Resource Description Framework (RDF) (Manola *et al.*, 2004). Basic idea in RDF is to describe relations between things in triples, consisting of subject, predicate, and object, where the predicate presents the relationship between the subject and the object. In addition, Uniform Resource Identifiers (URI) are used to name all these parts. As a whole, triples form a directed and labeled graph. For representing RDF descriptions, different serialization formats exist, *e.g.*, XML and Terse RDF Triple Language (Turtle). However, RDF has sometimes been considered too complicated even for developers (Nowack, 2006). A lightweight version of RDF, RDF in Attributes (RDFa) (Herman *et al.*, 2015) attempts to facilitate the use of simple RDF in HTML and XML documents. In addition to other widely known formats, such as Microformat (Khare and Çelik, 2006) and Microdata (Hickson, 2013), less known formats, such as Simple HTML Ontology Extensions (SHOE) (Heflin and Hendler, 2000) have been proposed.

Ontologies are controlled vocabularies used to define concepts and relations between those concepts on a certain domain (Guarino *et al.*, 2009). When information is linked toward ontologies, logical inferences can be made from it (Zhou *et al.*, 2006). The W3C Web Ontology Language (OWL) is a knowledge representation language for Semantic Web ontologies. It is used to describe classes, properties, individuals and data values in ontologies. Similarly to RDF, OWL can be serialized and exchanged in various syntaxes. (Horrocks *et al.*, 2012) The idea is that ontologies provide the meaning of things in a form of a graph of concepts and their properties, and RDF links the actual data from the RDF graph toward the concepts.

In smart home domain, various studies exist for modeling context ontologies, *e.g.*, (Kim and Choi, 2006), (Chen *et al.*, 2003) and (Xu *et al.*, 2009). Commonly, these solutions use context ontologies for modeling situational information of the entities in the space. Ontologies can help to automatically compose appropriate services and dynamically adjust the environment.

General Semantic Web research and related tools have commonly concentrated on scenarios where queries are made against static or rarely updated linked data, *e.g.*, (Bizer *et al.*, 2009) and (Hyvönen *et al.*, 2005). However, some applications would benefit form being able to use streaming sources as well. Linked data query languages could offer, for example, publish-subscribe mechanisms to facilitate the development of such applications. In addition to smart home research, need for such features have been reported, at least, in smart grid research (Wagner *et al.*, 2010). While approaches have traditionally also concentrated on read-only access to the semantic data, update query functionalities have been introduced to support scenarios with frequently changing information (Hert *et al.*, 2010) (Seaborne and Manjunath, 2007).

3.6 Intelligence

Intelligence layer provides services for controlling the smart home in a more sophisticated fashion, including, *e.g.*, adaptive logic for automatizing the smart home behavior and programming. As this layer provides functionality for the system control in a form of sophisticated logic, it can be considered as a more advanced type of interoperation. However, here the focus is on the logic of the interoperations. This logic is built on top of information or knowledge, while interoperation commonly relies on data. The characteristic of this layer is intelligent smart home behavior that happens based on Artificial Intelligence (AI) techniques.

Levermore (2000) defines intelligent device to have a microprocessor incorporated with the device. He suggests that the building becomes more intelligent when different microprocessor-controlled systems of the building share common bus systems and intercommunications. However, I consider smart home intelligence as the ability of the system to learn from occurring events, predict upcoming events, and adapt the system behavior to the changes in the human behavior. Still, as Vainio *et al.* (2008) noted, while automation is needed for the smart home, the user must always have the ability to override all of the decisions made by the system intelligence.

MASs are potential candidates for providing intelligence for the smart home control. Ideally, intelligent agents could communicate with Web services based systems without any changes in MAS or Web services specifications and implementations. Accordingly, solutions such as AgentWeb Gateway (Shafiq *et al.*, 2006) have been proposed to enable this interoperability. Ramos *et al.* (2008) note that Ambient Intelligence (AmI) is not achieved without AI technologies, and that agents are a good way to model and represent entities such as rooms and persons.

Research of Multi-Agent System for Building cOntrol (MASBO) (Qiao *et al.*, 2006) describes the control of a building using MASs. In our interoperability model, MASBO directly fits into the intelligent interoperability layer. However, it does not consider representation of semantic information, nor does it use Web technologies for the integration and interoperability.

Rashidi and Cook (2009) presents CASAS, an adaptive smart home system utilizing machine learning to discover patterns in smart home activities aiming to automate the home. Learning methods are described in detail but information on the implementation techniques, *e.g.*, possible standards and protocols, is missing. Thus, the system certainly provides intelligent functionality for the smart home, but syntactic interoperability with any other devices or systems is not guaranteed.

In a long-term user study on living in a smart home, the tenants reported that the future smart home control systems should be adaptive and proactive (Kaila *et al.*, 2008). Vainio *et al.* (2008) presented such an adaptive smart home system with proactive fuzzy control for lighting. Their solution was able to learn from the user behavior, distinguish routines from random actions, and provide temporary override control for the user. Interestingly, they found that users do not usually have strict demands on lighting conditions. It only becomes a problem when the lighting is clearly wrong.

In general, various projects, *e.g.*, (Youngblood *et al.*, 2005) (Rashidi and Cook, 2009) (Choi *et al.*, 2005), have proposed solutions for implementing learning in smart homes. However, it is common that the projects integrate AI and related techniques directly with the home systems, ignoring Semantic Web and Web services technologies. This inhibits interoperability with Web technology based control systems, as some of the middle interoperability layers are not implemented with common Web based techniques or the implementation is completely missing.

4 Proof of Concept Implementations

This chapter presents the implementations produced within the dissertation work. Only high-level descriptions are given here, and the details can be found in each corresponding publication. Instead, the implementations are here reviewed in relation to the interoperability model presented in Chapter 3. These relations are depicted in Figure 4.

The presented implementations demonstrate a Web based smart home interoperability architecture. However, this collection of solutions for different layers tries to describe why and how interoperability should be improved. Thus, these implementations are showing the direction for the future development of interoperable smart home systems.

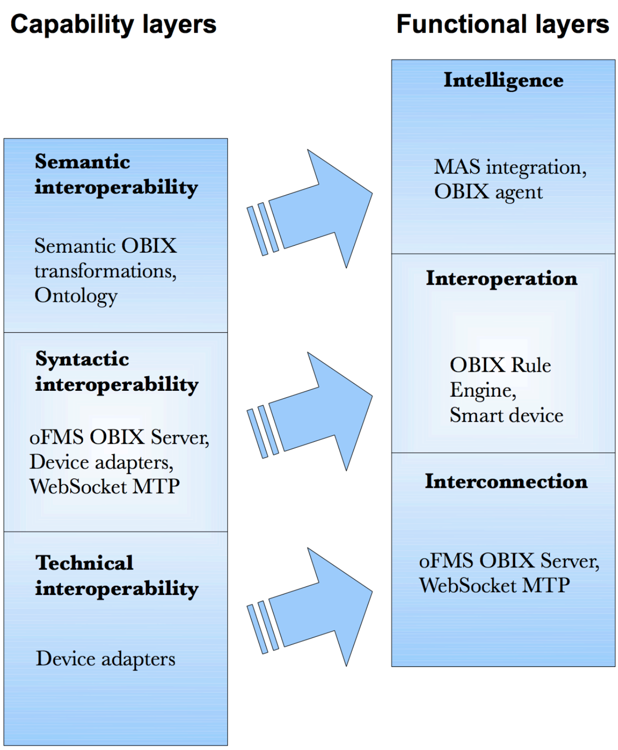

Figure 4: Implementations in relation to the interoperability model.

4.1 Technical and Syntactic: Device Adapters

Various device adapters were developed to provide technical and syntactic interoperability for the devices. As depicted in Figure 5, each device adapter functioned as a converter between the device specific and common communication protocol and data format. In practice, all device adapters implemented OBIX client side functionalities and another communication protocol needed for the communication with the device. Adapters were also responsible for building device data descriptions into OBIX data format.

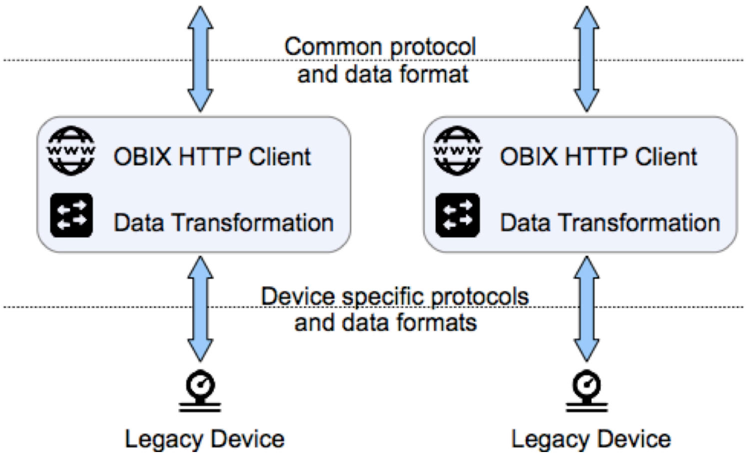

Figure 5: Device adapters provide technical and syntactic interoperability by converting device specific protocols and data format into OBIX HTTP communication and OBIX XML.

Ensto Smart home control system provided a limited native M2MXML language support for the communication through RS-232 serial port. Unfortunately, the used device was unable to send data to the port but successfully received all the commands sent to it. The adapter described the device data in OBIX format and converted the commands from OBIX to M2MXML. Another device, Enervent Pingvin also used serial port. However, it supported a lower-level Modbus protocol for the communication. The support was fully implemented and the communication functioned in both directions. (Järvinen, 2007)

Three devices were used in Publication II and Publication IV. Tracking system, lighting system, and a virtual TV device had corresponding adapters implemented for them. Tracking system adapter communicated with the tracking system controller via UDP to poll for user position. The adapter also implemented a method for predicting user moving from one room to another, and was thus categorized as a smart device. Similarly, the lighting system was manageable via UDP. Virtual TV device was implemented with XForms and JavaScript to run in a browser and play videos from YouTube.

All the described device adapters provided both technical and syntactic interoperability by unifying both the communication protocol and the data model. Next subchapter describes how the adapters connected to an OBIX server to enable interconnection, and to build basis for higher-level interoperability.

4.2 Syntactic and Interconnection: oFMS

An integration platform called open Facility Management Server (oFMS) was developed to provide interconnectivity and syntactic interoperability for the devices. Device adapters discussed in the previous subchapter were able to register their services and data to the server, through which the devices could be controlled. A general description of the implementation is given in Publication I, and it has been exploited and further developed in Publication II, Publication IV, and Publication V. Early development stages and additional examples of developed device adapters are presented in (Järvinen, 2007).

The software is based on Open Building Information eXchange (OBIX) specification version 1.0 (Frank, 2006), and it functions as an integration platform for the home and building automation devices. The platform uses Web services technology for connecting devices to a stand-alone server, and for the user agents to access it. More specifically, it provides OBIX HTTP interface for all interactions. Standard OBIX server functionality is extended with a signUp operation to allow clients, *e.g.*, devices or device adapters to add data specific to their functioning to the server. As depicted in Figure 6, the platform integrates devices into a centralized OBIX server, unifies the data representations, and provides a standard interface for managing the data.

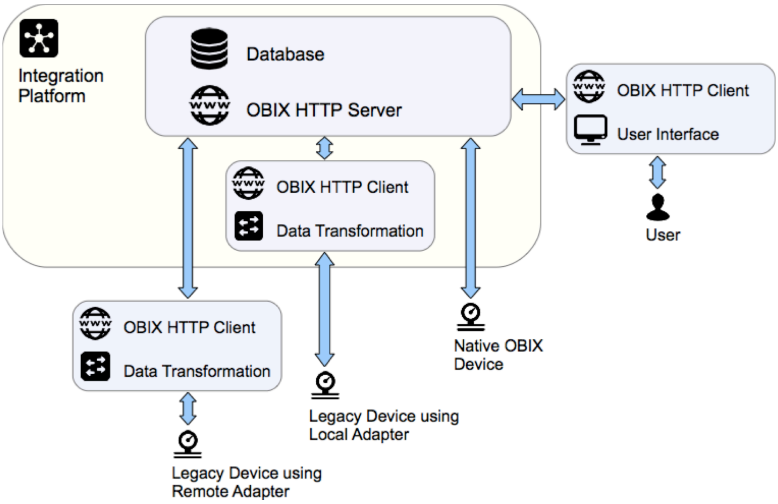

Figure 6: OFMS integration platform provides interconnection and syntactic interoperability by connecting heterogeneous devices to the server and unifying the representation of the data.

The goal of oFMS is to enable legacy devices to be presented on the Web using common building information XML format and to provide device management through common Web services interface. Legacy devices are connected via device adapters that implement OBIX client functionality. Similarly, OBIX ready devices could implement the same functionality out-of-the-box. Because all of the devices are connected through the OBIX interface, they can be local or remote. This allows developers to implement OBIX clients on the platform of their choice. One downside of the used OBIX 1.0 specification is that it does not offer any server push method. In real-time sensitive application scenarios, this can lead to a lot of unnecessary polling.

4.3 Interoperation: OBIX Rule Engine

Open Facility Management Server originally provided only interconnection and syntactic interoperability. To add better functional interoperability support, OBIX Rule Engine was developed and integrated into the server. It extends the server functionality by enabling definition of simple interoperations, *i.e.*, rules. A general overview of the implementation is presented with experimentation in Publication II, and a more detailed description with a practical use case and analysis is given in Publication IV.

OBIX Rule Engine was developed from the interoperability point of view. In practice, seven requirements were set to ensure that the description format, interaction in distributed environments, knowledge base updates, and system management were implemented accordingly to facilitate the interoperability. As a result, adoption of new data format and interface is avoided as rules are described using OBIX format, and the rule management is done through the OBIX HTTP interface. In addition, interoperations can function in distributed environments, as the system can perceive the environment and command actions with web resource requests.

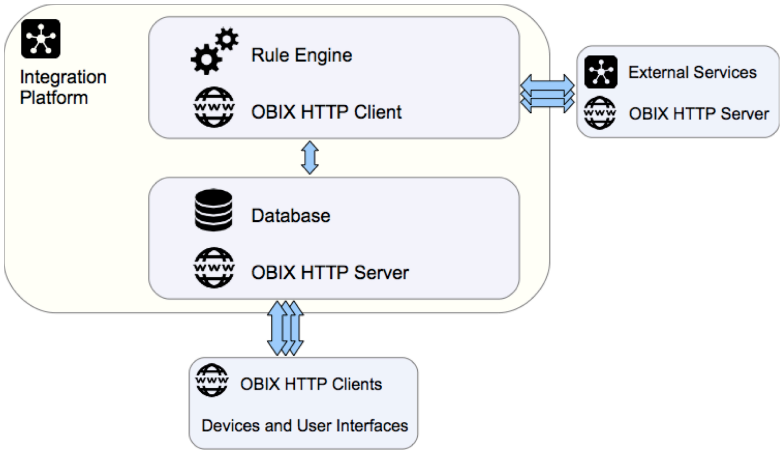

Figure 7: OBIX Rule Engine is integrated into oFMS integration platform and provides interoperations for distributed device environments.

As depicted in Figure 7, the rule engine is integrated into oFMS integration platform. OBIX server is extended to provide a Smart Application service accessible from the OBIX lobby, a standard location for common services. The smart application service provides an operation for creating new rules that are represented in OBIX XML. Further rule management is done by directly manipulating the rule parts via their URIs and OBIX interface.

The interoperation support makes it possible to program the system to automatically execute behavior that the users desire. Based on changes in one or more subsystem devices, the rule engine can command changes in other subsystems. This type of functional interoperability between devices can be considered as a first level of interoperability where devices function in cooperation rather than as individual units. The rule engine handles the coordination of operations between the devices. Furthermore, in OBIX Rule

Engine all device information is requested via URIs, thus allowing devices to be distributed into multiple OBIX servers.

4.4 Interoperation: Smart Device

A smart device was developed that demonstrates functional interoperability between multiple system components. The device provides tracking system service on a basic level but also implements an anticipation method for predicting the intention of a user to move from one room to another. An overview of the smart device with a practical use case is presented in Publication IV, and the implementation and experimentation are detailed in Publication II.

Figure 8 illustrates the interoperability of the smart device. OBIX Rule Engine is running on one integration platform. All devices, including the tracking system smart device, publish their data on another, external OBIX integration platform. Smart device is tracking the position of the user in the space, reports it to the integration platform, and is able to predict the intention of walking from one room to another. Based on this prediction, it is able to decide when the lights in the rooms should be lit and when to switch them off. This anticipatory lighting represents a simple example of proactive computing in a smart environment where the system takes the initiative and anticipates events (Vanhala *et al.*, 2005).

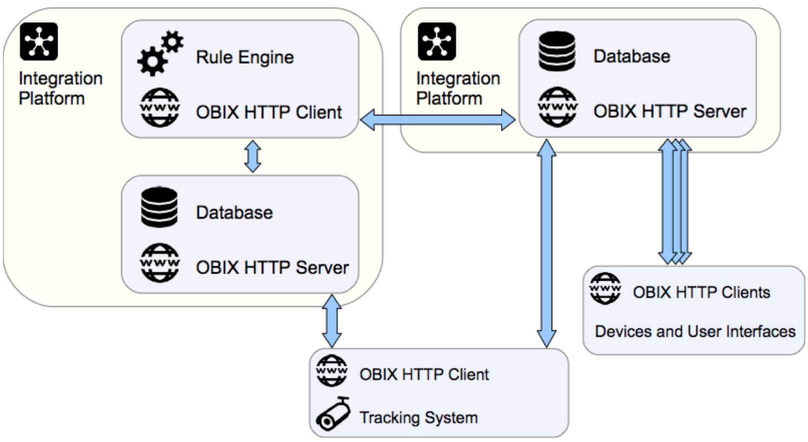

Figure 8: The tracking system functions as a smart device being able to proactively make decisions to command other OBIX devices.

The goal of the anticipation was to remove the system delay, and thus make the interaction more pleasant. Light controls of the rooms were batched using simple rules on the rule engine. Thus, the smart device was not commanding individual lights separately, but used the rule engine for that. In addition to the interaction with the rule engine, tracking system also published current user location data on the external OBIX integration platform. This data was used by additional rules to provide backup lighting functions for correcting possible errors done by the prediction logic.

The goal of the system was not on developing a superior prediction algorithm but on demonstrating cooperation of the proof-of-concept smart device with the rest of the system. The applied Linear Discriminant Analysis (LDA) method is far from the optimal solution. However, it performs well enough to provide improvement to the user comfort.

From the functional and communicational point of view, the smart device can be considered as another control system comparable with the rule engine. Thus, the system demonstrates the capability of functional interoperability and cooperation between multiple control systems. This type of cooperation of equal system components for implementing intelligent systems is also the goal of many MAS applications.

4.5 Syntactic and Interconnection: WebSocket MTP

Cooperating intelligent agents in MASs can as a whole perform complex tasks that are difficult to be managed by one centralized control system. Such system can be built from simple autonomous agents that each handle reasonably simple tasks but by interoperating with others can provide intelligent solutions. Combining MASs with Web services technologies could thus help to provide intelligence for Web services based smart home systems. Unfortunately, MASs and smart home systems have evolved to use differing protocols thus complicating their integration. WebSocket Message Transport Protocol (MTP) is a step toward a better mutual communication method between the two. However, *per se* it provides interconnection and syntactic interoperability between agents systems. WebSocket MTP and an implementation for the Java Agent DEvelopment Framework (JADE) are presented with a detailed analysis and a use case in Publication III, and integration with the OBIX platform is presented in Publication V.

Preliminary research on the utilization of WebSocket technology in MAS communication is presented by Järvinen and Garcia-Gasulla (2012).

MTPs are a FIPA compliant method in agent systems to establish the inter-platform agent communication. Regardless of the location of the receiver, individual agents can use the standard messaging interface provided by the platform, as MTPs handle the complexity of the message transport. An agent platform can support several different MTPs simultaneously. For example, in JADE agent platform the default MTP is HTTP, while other options are supported both natively and as extensions.

WebSocket MTP provides another, WebSocket based message transport protocol for the agent platforms. One the one hand, providing full-duplex bi-directional communication, WebSocket communication fits better to the agent communication paradigm that HTTP. On the other hand, with a true server push feature, it enables communication with peers that can only implement client side functionality. In practice, agents can be implemented to run within browsers but also communicate with external agents running in traditional agent platforms. Figure 9 illustrates the interoperability provided by the WebSocket MTP.

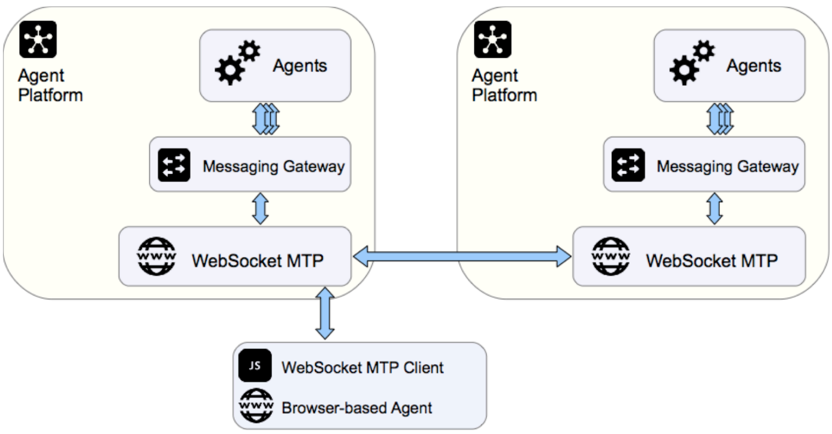

Figure 9: WebSocket MTP provides interconnection and syntactic interoperability with a WebSocket based communication protocol between agent systems. Exploiting WebSocket MTP JavaScript library, browser-based agents can communicate with traditional agent systems.

4.6 Semantic: Semantic OBIX

OBIX information is normally presented in XML syntax introduced in the OBIX specification. To add semantic interoperability support for oFMS, a transformer was developed for converting the OBIX XML information into RDF. A detailed description and evaluation of the solution are given in Publication V.

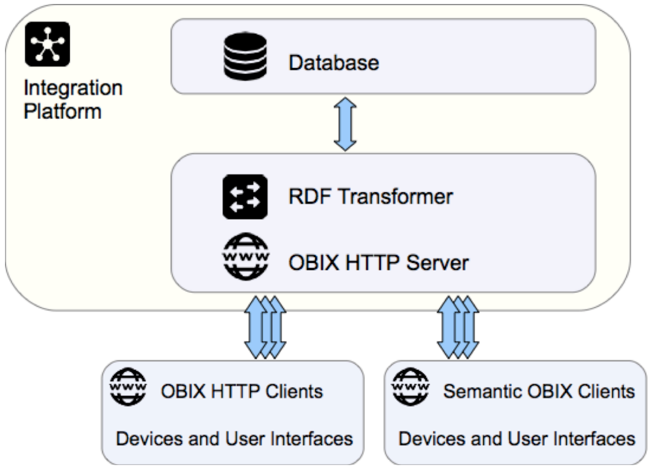

Figure 10: OBIX RDF Transformer provides semantic interoperability for oFMS integration platform.

Figure 10 illustrates the integration of semantic transformer into oFMS. Traditional and semantic OBIX clients are both able to communicate with the integration platform over HTTP. The transformer is integrated with the OBIX HTTP server, and used to convert information from common OBIX XML into RDF/XML when required. Clients define a URI parameter to inform the server when the RDF format is preferred.

In practice, two XSLT templates are used to handle the transformation process. OBIX ontology is defined to represent the smart home information in an explicit semantic format. Both XSLT transformations generate very similar OWL/RDF/XML data, and the only difference is that in addition to OBIX ontology, the other one defines a namespace also for FIPA-RDF ontology used in MASs. This prepares the support for interoperability on the intelligence layer.

4.7 Intelligence: OBIX Gateway Agent

As mentioned before, cooperating intelligent agents in MASs can as a whole perform complex tasks that are difficult to be managed by one centralized control system. Therefore, smart home systems could benefit from the advancements of MASs in a form of intelligent control. OBIX Gateway Agent provides a basis for interoperability on the intelligence layer by integrating smart home and multi-agent systems. A detailed description of the integration is presented with performance measurements in Publication V.

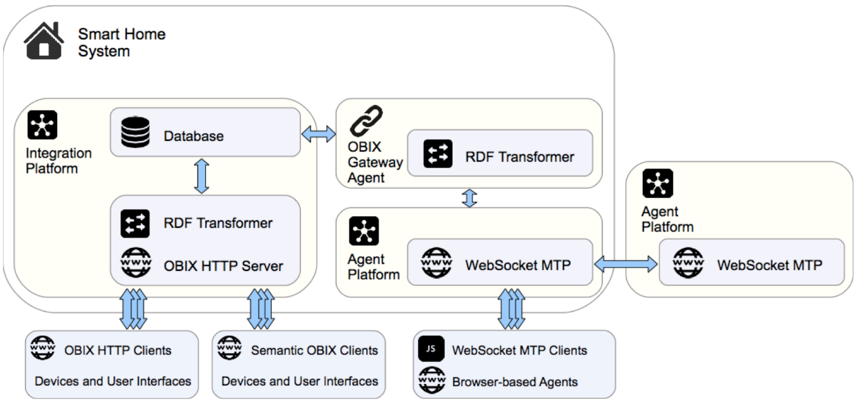

Figure 11: OBIX Gateway Agent enables intelligence layer interoperability by integrating oFMS and agent platforms.

Figure 11 depicts the interoperability provided by the resulting system integration. OBIX Gateway Agent functions as a bridge between the OBIX and agent platforms. Now, both OBIX clients and FIPA compliant agents are able to provide for and consume services of the smart home system. Regular and semantic OBIX clients can access the system through the OBIX server, and agents through the agent platform. The agent platform further provides a WebSocket MTP interface, enabling cooperation with external agent platforms and browser-based agents.

The solution has another advantage concerning the communicational capabilities. While OBIX 1.0 specification provides a feature called watches for reducing the traffic load, it does not offer any true real-time notification method to implement server-push communication. However, OBIX Gateway Agent communicates directly with the integration platform database, and provides a standard subscription method for the agents. Using this feature, agents can receive updates on specific data from the server immediately when available. Automatic notifications followed by the subscriptions also prevent the possibility of occasionally not registering some frequently changing data values.

5 Discussion

The work presented in previous chapters concentrated on the interoperability model for smart homes. The model consists of six layers, each contributing toward better interoperability. Publications present interoperability solutions on these layers using RESTful Web services, and other Web and agent technologies. In this dissertation, these solutions were reviewed concentrating especially on interoperability, and categorized based on the interoperability layers.

In this chapter, current situation in smart home interoperability, and then the movement toward IoT, WoT, and smart homes are discussed in general.

5.1 Smart Home Interoperability

SOA and WS-* technologies have been successfully applied to enterprise integration and interoperability problems. Smart home and enterprise environments have similarities in their distributed nature of services and the need for interoperability between various heterogeneous entities. Enterprise application integration has heavily relied on WS-*, namely SOAP, WSDL, UDDI, and related technologies. There have been attempts to apply these solutions also to smart homes. Unfortunately, the smart home industry has lacked the desire to advance interoperability over company boundaries. It seems to have been in business interests of the companies to make their products incompatible with products from their competitors on the market. Also, the solution to the problem in smart homes has to be different from the enterprise integration, as there are no similar resources available as the companies have. Integration of various heterogeneous systems in companies requires the work of, *e.g.*, software integration architects and system developers, and each company can use a different set of systems (Newcomer and Lomow, 2004). Each smart home owner, however, can only put very limited resources on the integration of the home devices and systems. Interoperability needs to be built-in. RESTful Web services might provide a more suitable, simple, and lightweight platform for realizing this, as presented in this dissertation. As opposed to WS-* technologies, RESTful Web services use HTTP as a true application protocol (Guinard, 2011). Loose coupling of REST approach is more suitable for the global integration scenario in the WoT context. It also provides better adaptability in constantly changing smart home environments (Pautasso and Wilde, 2009).

This dissertation concentrates mostly on open standards provided and managed by international standards bodies, *e.g.*, W3C and OASIS. However, to date open solutions have failed to provide widely adopted solution for interoperable smart homes. Maybe the future of smart homes is not completely open, while many of the proposed solutions could still be utilized. Highly possible breeding grounds for the smart homes are ecosystems. A large body in each ecosystem, a central company or organization, *e.g.*, Apple or Google can define the protocols and data structures that are used in that ecosystem. Developers can implement support for the selected ecosystems for their products, and smart home owners can choose on which ecosystem they base their home system. Such ecosystems can be expected to be built by and be based on the platforms from the biggest international mobile device companies. There are already more mobile devices in the world than there are human beings. We can expect mobile phones to be used as primary user interface for smart homes (Kaila *et al.*, 2008). Thus, we have already started to build our smart homes, as we are continuously carrying devices that can be used for multiple smart home purposes, including system management, remote control, sensing, and location awareness.

5.2 Toward IoT, WoT and Smart Homes

As discussed earlier, work on IoT has long concentrated on defining and standardizing the low-level protocols for the communication between heterogeneous distributed entities (Guinard, 2011). This has led to a situation where products are developed in isolation and are rarely using same high-level protocols and data formats. However, Solutions for embedding IP stack in small low-power devices, *e.g.*, 6LoWPAN (Shelby and Bormann, 2009) and CoAP (Shelby *et al.*, 2014), are important and a required part of the process.

W3C has recognized the excessive attention given to the IoT transport layer, and reports a growing awareness that the actual business will be in the services. Now, the focus needs to be shifted from the IoT transport layer to the WoT, the application layer on top of the IoT. (Heuer and Raggett, 2014) W3C has a strong support for starting new standards work on WoT (Heuer and Raggett, 2014). Hopefully the work will result in standards that are widely adopted by the WoT developers. Common protocols and data formats are the basis from which interoperability among things can be built, and IoT and WoT realized. However, that requires wide commitment to these standards, and development of services and tools based on these standards. In addition, layer stack of WoT application architecture model is not strictly defined, as the layers can overlap. It is an ecosystem of services to ease the development of WoT applications. (Guinard, 2011) To reach the goals of the WoT, standard techniques are required for each layer.

WoT application framework architecture builds on and extends the REST architecture to fit the requirements of the physical world. It focuses on device accessibility, findability, sharing, and composition. (Guinard, 2011) These issues are important and should be kept in mind when developing applications for the WoT. However, similarly to the Web based smart homes, I want to highlight the importance of the interoperability for the WoT. While lower interoperability layers relate better to the WoT application framework architecture layers, higher semantic and functional interoperability layers need to be addressed as well. This can turn the data about things into information (Kubicek *et al.*, 2011) and enable automatized operations between things. Also, things from the WoT can provide services for the Web based smart homes, and *vice versa.*

Similarly as Guinard (2011) describes the WoT as an application layer on top of the IoT, Kamilaris *et al.* (2011) describes the use of Web-based standards to achieve interoperability in smart homes. This work also focuses on the lower level interoperability, but also suggests the use of Web mashups for programming functionality.

Due to the similarity, WoT research can contribute toward the research and deployment of Web based smart homes in multiple ways. The two can be considered as highly overlapping domains. Both the WoT and the Web based smart homes have the same idea of taking the Web and extending it so that anyone can plug devices into it (Kamilaris *et al.*, 2011) (Wilde, 2007). However, there are also differences. For example, because LAN is generally used in the home environment, discovery mechanism can rely on broadcasting based solutions. Also, security and privacy issues need different kind of consideration.

While the amount of data communicated over the Internet and the Web is already rapidly increasing, the IoT and the WoT will accelerate the growth even further. Thus, the domains need also to address several issues known from the Big Data research, including preprocessing, collecting, analyzing, and visualizing the data. In addition, data produced by physical devices introduce new challenges for the Big Data research. The nature of this type of data is continuous, streaming, real-time, and dynamic with spatiotemporal dependency. Solutions for automatically enriching semantics of the data, binding time-specific data with the location, and for continuous querying of semantic data are required. Also, appropriate methods for reducing the amount of the data sent over the network need to be studied. This considers, at least, preprocessing of the raw data by filtering, and handling the short-term and long-term storage of the data. (Barnaghi *et al.*, 2013)

Huge amount of research on smart homes exists. Nevertheless, the fact is that our homes today are generally not very smart. Reasons for this are found in lack of standard protocols and low collaboration between device manufacturers. Today, as the required technology and infrastructure are truly ready to be exploited, smart homes might finally become reality for the people. If WoT first catches on and succeeds to develop common standards for the communication and data representation, that progress can also penetrate into the smart home domain. This would also facilitate the cooperation between the smart home and the WoT services. In fact, by sharing the technological practices, smart homes could be part of the WoT.

6 Conclusions

In this last chapter, contributions are summarized together with the answers for the research questions given in Chapter 1.2, and finally the future work on the smart home research is discussed.

6.1 Contribution

The main contributions of the research presented in this dissertation are the extended smart home interoperability model described in Chapter 3, and the advances in the smart home interoperability described in the publications and reviewed in Chapter 4.

As presented in Chapter 1.2, this dissertation studies *how Web technologies can be used to improve smart home interoperability*. The problem is formulated into two research questions. Here, the findings answering to these questions are summed up.

The first research question (Q1) considers the problem of defining *the key elements required for the smart home interoperability*. Answer to this question is shaped based on the experiences and results from the publications, and depicted in a form of the smart home interoperability model described in Chapter 3. I believe, that focus in developing smart home devices and systems need to be turned more toward functional interoperability. Separate smart devices are not enough in the future. To gain added value, devices need to interoperate and provide real automation of the system as a whole. Thus, I believe that smart home system developers should consider implementing solutions for all identified interoperability layers. In this dissertation, these layers are defined as the key interoperability elements: technical interoperability, syntactic interoperability, semantic interoperability, interconnection, interoperation, and intelligence. Technical and syntactic interoperability provide capability for communication with common network protocol and data format.

Semantic interoperability ensures capability of system components to have shared meaning of the data. Interconnection, interoperation and intelligence present different layers of functionality that can be built on top of the first three layers. It should be noted that discovery, sharing, and security are important parts that do contribute to the interoperability. The model presented in this dissertation, however, focuses on different levels of interoperability based on the data representation and enabled functional properties. Discovery, sharing, and security instead penetrate through the interoperability model, as solutions for them are needed on several layers. Also, as this work is mainly based on Web technologies, and Guinard (2011) has already thoroughly presented solutions for these aspects for the WoT, discovery, sharing, and security have been intentionally left outside of the scope of this dissertation.

The second research question (Q2) considers *how Web technologies can provide solutions for the different elements of the smart home interoperability.* As the six identified interoperability layers were defined as the key interoperability elements, this question requires an answer that covers these six smart home interoperability layers. Again, the answer can be composed based on the experiences and results from the publications. Research in the publications present solutions for integrating devices into a smart home system using Web services (Publication I), cooperation of a smart device with the rest of the system using Web services (Publication II), MAS communication over WebSocket protocol (Publication III), rule engine with management, sensing, and actuating via Web services (Publication IV), and integrating semantically enriched smart home system with a MAS (Publication V). The proposed solutions demonstrate different ways to achieve interoperability using Web technologies. However, presented proof-of-concept implementations need to be considered only as examples of simple ways to realize different interoperability layers in Web based smart homes. As discussed in this dissertation, in addition to techniques applied in this work, a variety of similar, and other applicable techniques exist. For example, while RDF was here used as a semantic data format, RDFa can be more suitable format in some scenarios. Similarly, OBIX turned out to be a good candidate for a smart home language and interface, but many other proposals for the same purposes exist as well.

Device adapters were implemented for connecting real and virtual devices to integration platforms. These adapters provide technical interoperability by implementing both device specific communication protocols and a common OBIX client interface. Adapters also provide syntactic interoperability by converting diverse device data formats into standard OBIX XML format. In both interoperability solutions, Web technologies were used. For technical interoperability, OBIX HTTP binding was used as a common communication protocol. Similarly, the XML syntax of the common data format is well suited for the Web. As OBIX promotes a standard XML format and Web services guideline, it was a natural choice for the implementation. However, as discussed in Chapter 2.4, many other options exist as well. Using Web technologies, technical interoperability for smart home devices can be implemented with converters from device specific protocols into any common Web protocol, *e.g.*, HTTP or SOAP. Also, use of a common Web friendly syntax for the data, *e.g.*, XML or JSON, provides syntactic interoperability.

In Publication I, I presented a standalone integration platform that provides home for heterogeneous distributed system components through Web services interface. The system interconnects devices through their adapters into the integration platform. Interconnection of distributed device services can here be considered as an abstract concept, as higher-level interoperability was the actual goal of the implementation. Thus, in addition to merely interconnecting device management into one central point, also device data was unified. Through this unification of the data representation, system provides syntactic interoperability. While OBIX data syntax for each device is actually constructed already by the device adapters, the integration platform functions as an OBIX server, providing access to the data and services of all connected devices.

In Publication IV, I presented a rule engine that operates in distributed device environments over Web services interface. The rule engine enables interoperations between different devices. With the rules, state change in one or more devices can cause actions to be executed, leading rule engine to command again other devices to change their states. This interoperation is implemented with Web technologies. Rules are described in OBIX XML format, they are created and managed through OBIX HTTP Web services interface, and both the rules and device services are accessed using URIs.

A smart device, that cooperates with a rule engine and other devices through Web services interface was presented in Publication II. Similarly to the rule engine, the smart device is categorized as a control system, as it can directly command actions in other OBIX solutions. It thus carries out interoperations between devices. Again, used Web technologies were OBIX HTTP interface and data format.

In Publication III, WebSocket communication for multi-agent systems was presented. It provides interconnection and syntactic interoperability between different MAS platforms and browser-based agents. WebSocket communication module functions as a bridge between different agent platforms, and agents can send messages in a same manner than inside one platform. It should be noted, that commonly MAS platforms already have HTTP communication built-in for this purpose. WebSocket communication provides a better fit to the agent communication paradigm, and enables communication with browser-based agents. Both interoperability solutions use Web technologies. Interconnection is enabled through WebSocket communication. Naturally, HTTP is also a viable option to be used in communication between MASs, as it is already available and highly supported by different platforms. However, WebSocket communication provides benefits when communicating with other solutions, such as browser-based agents. Syntactic interoperability is provided with specifying the data format sent over the WebSocket. This format is using XML for the message envelope, but ACL for the actual message. ACL was selected for better conformance with agent technologies and to follow the message structure used in the HTTP communication in MAS platforms.

In Publication V, I presented transformation of smart home data into semantic representation based on an ontology. This solution provides semantic interoperability and uses Web technologies, specifically, HTTP, XML, URI, XSLT, OWL and RDF. OBIX ontology was defined with OWL to align OBIX data model with the corresponding semantic RDF representation. Transformation was done in real-time using XSLT when oFMS received an HTTP request with a specific URI parameter to inform that the RDF format is preferred.

Integration of a smart home system with a multi-agent system was presented in Publication V. The solution provides basis for intelligence to be used from MASs for smart home control. While the integration enables the communication between MAS and smart home system, actual learning and adaptive control applications were left out of the scope here. However, there was an additional benefit gained from the integration. As agents can access and manage OBIX devices through the platform integration, and MAS was earlier extended with WebSocket communication, now also OBIX devices were made accessible through WebSocket interface for browser agents. In addition, while OBIX 1.0 does not offer any true real-time notification method to implement server-push communication, WebSocket communication provides a way to receive updates immediately when available. While the integration itself did not use any Web technologies, the resulting system enables the use of MASs, and other external AI systems for the smart home control through both HTTP and WebSocket interfaces.

I presented here, how using Web technologies the software implementations provide solutions for the different layers of the interoperability model. In addition to Web technologies, Semantic Web and MAS technologies were applied and combined in the development. As mentioned earlier, a number of other Web technology solutions exist that could be used for the same purposes than described here. However, examples given here demonstrate that Web technologies can easily be used to provide better interoperability on all of the interoperability layers.

6.2 Future Work

The presented work contributes to the smart home interoperability by introducing an interoperability model extended with functional properties that reflect operational capabilities of the systems. While available commercial products fail to provide high level of interoperability between different solutions, ecosystems represented by large international companies and organizations provide natural breeding grounds for realizing such interoperable device groups. Developments presented in this dissertation work for different types of interoperability can guide future development toward the right direction.

Future work on the smart home domain needs to consider all interoperability layers for the presented visions of future living environments to be reached. Standard network infrastructure for the communication is already commonly available in modern buildings, and hand-held mobile devices available for the system management are ubiquitous. In spite of that, however, smart homes have not yet become reality for the general public. Smart home interoperability has faced a chicken-and-egg problem. Availability of devices with high level of interoperability could accelerate the market demand. At the same time, the current low demand for smart devices fails to inspire companies to invest on interoperability. However, a tipping point will hopefully be reached in the near future, and interoperability investments become profitable for the companies. At that time, battle takes place between the device ecosystems, and finally, interoperability becomes an essential factor for smart devices.

Issues on technical interoperability have been agreed on, and commercial products are already on the market. These devices communicate over TCP/IP networks, but fail to exhibit syntactic interoperability. Interconnection can still, in some cases, allow semi-integrated user interface to gather independent device services into one widget based application. Usage of proprietary interfaces inhibits interoperation between individual devices. Furthermore, semantic interoperability using standard Semantic Web technologies appears only in research. Still, when commercial products with syntactic interoperability become common, they will probably use some kind of semantic notation. However, for exploiting common AI tools for the intelligence, standard semantic technologies should be used.

Practical next step for the consumer products is to concentrate on syntactic interoperability and interoperation. Common open interface for the device communication provided by a highly supported mobile platform could achieve significant market penetration if wide range of moderately priced devices would be made available. Then, simple tools for end-user system management and programming are needed for remote access and application composition.

Research on smart home interoperability should concentrate on further advancements. Regardless of which interface will be used in the future for syntactic interoperability, research can demonstrate possibilities it will enable. Comparison of different semantic notations and technologies can reveal pros and cons of each of them when applied to the context of smart homes. Semantics should also be considered together with the intelligence, as learning and adaptive behavior are the goals that are being pursued. Also, in addition to technological research, it is important to conduct studies from the human sciences and design research perspectives toward understanding social and cultural aspects of the introduction of smart home technologies (Mäyrä *et al.*, 2006).

Interdisciplinary research on smart homes and AI is necessary for developing suitable learning algorithms. These algorithms need to be adapted based on the requirements of the smart home. For example, while in some AI application contexts errors can be easily tolerated, in smart home context that is usually not the case. Also, right balance between automated intelligent system behavior and user manipulation need to be addressed (Kaila *et al.*, 2008). Related AI areas include at least MASs, machine learning, and neural networks.

References

Alam, M. R., Reaz, M. B. I. and Ali, M. A. M. A Review of Smart Homes - Past, Present, and Future. In *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. 42. Pages 1190-1203. IEEE. 2012.

Aldrich, F. K. Smart Homes: Past, Present and Future. Inside the smart home. Vol. 1. Pages 17-39. Springer. 2003.

Alonso, G., Casati, F., Kuno, H. and Machiraju, V. Web Services: Concepts, Architectures and Applications. Springer. 2004.

Arnold, K., Scheifler, R., Waldo, J., O'Sullivan, B. and Wollrath, A. Jini Specification. Addison-Wesley Longman. 1999.

Barnaghi, P., Sheth, A. and Henson, C. From Data to Actionable Knowledge: Big Data Challenges in the Web of Things. In *IEEE Intelligent Systems.* Vol. 28. Pages 6-11. IEEE. 2013.

Bellwood, T., Clément, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y. L., Januszewski, K., Lee, S., McKee, B., Munter, J. and Riegen, C. UDDI Version 3.0. OASIS Published specification. 2002.

Benslimane, D., Dustdar, S. and Sheth, A. Services Mashups: The New Generation of Web Applications. In *Internet Computing.* Vol. 12. Pages 13-15. IEEE. 2008.

Bergwik, E. Using the DIAL Protocol for Zero Configuration Connectivity in Cross-Platform Messaging. Master's Thesis. Linköpings Universitet. 2014.

Bernardet, U., Bermúdez i Badia, S., Duff, A., Inderbitzin, M., Le Groux, S., Manzolli, J., Mathews, Z., Mura, A., Väljamäe, A. and Verschure, P. F. M. J. The Experience Induction Machine: A New Paradigm for Mixed-Reality Interaction Design and Psychological Experimentation. In *The Engineering of Mixed Reality Systems.* Pages 357-379. Springer. 2010.

Berners-Lee, T., Fielding, R. and Frystyk, H. Hypertext Transfer Protocol -- HTTP/1.0. RFC 1945. Internet Engineering Task Force. 1996.

Berners-Lee, T., Hendler, J. and Lassila, O. The Semantic Web. In *Scientific American.* Vol 284. Pages 28-37. 2001.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. and Hellmann, S. Dbpedia - A crystallization point for the Web of Data. In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*. Vol. 7. Pages 154-165. Elsevier. 2009.

Booth, D., Champion, M., Ferris, C., McCabe, F., Newcomer, E. and Orchard, D. Web Services Architecture. Working Draft. W3C. 14 May 2003.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D. Web Services Architecture. Working Group Note. W3C. 11 February 2004.

Box, D., Cabrera, L. F., Critchley, C., Curbera, F., Ferguson, D., Graham, S., Hull, D., Kakivaya, G., Lewis, A. *et al.* Web Services Eventing (WS-Eventing). Member Submission. W3C. 2006.

Braden, R. (ed.) Requirements for Internet Hosts -- Communication Layers. RFC 1122. Internet Engineering Task Force. 1989.

Chan, M., Estève, D., Escriba, C. and Campo, E. A review of smart homes—Present state and future challenges. In *Computer Methods and Programs in Biomedicine.* 2008.

Chemishkian, S. and Lund, J. Experimental Bridge LonWorks / UPnP 1.0. In *Proceedings of the First Consumer Communications and Networking Conference.* Pages 400-405. IEEE. 2004.

Chen, H., Finin, T. and Joshi, A. An ontology for context-aware pervasive computing environments. In The Knowledge Engineering Review. Vol. 18. Pages 197-207. Cambridge University Press. 2003.

Chen, L., Nugent, C., Mulvenna, M., Finlay, D. and Hong, X. Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments. In *Intelligent Patient Management.* Pages 279–296. Springer. 2009.

Choi, J., Shin, D. and Shin, D. Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances. In *IEEE Transactions on Consumer Electronics.* Vol. 51. Pages 301-306. IEEE. 2005.

Cockburn, C. Domestic technologies: Cinderella and the engineers. In *Women's Studies International Forum.* Vol. 20. Pages 361-371. Elsevier. 1997.

Cook, D. J. and Das, S. K. How smart are our environments? An updated look at the state of the art. In *Pervasive and mobile computing*. Pages 53-73. Elsevier. 2007.

Crnkovic, G. D. Constructive Research and Info-computational Knowledge Generation. In *Model-Based Reasoning in Science and Technology.* Vol. 314. Pages 359-380. Springer. 2010.

De Souza, L. M. S., Spiess, P., Guinard, D., Köhler, M., Karnouskos, S. and Savio, D. SOCRADES: A Web Service Based Shop Floor Integration Infrastructure. In *Proceedings of the First International Conference on Internet of Things.* Pages 50-67. Springer. 2008.

DLNA. Part 1-1: Architectures and Protocols. An Industry Guide for Building Interoperable Platforms, Devices, and Applications. DLNA Guidelines March-2014. 2014.

Driscoll, D. and Mensch, A. (eds.) Devices Profile for Web Services (DPWS) Version 1.1. Standard. OASIS. 2009. http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.pdf

EIF. European Interoperability Framework for Pan-European eGovernment Services. Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens. European Communities. 2004.

Eyke, M. Building Automation Systems: A Practical Guide to Selection and Implementation. BSP Professional Books. 1988.

Felser, M. and Sauter, T. The fieldbus war: History or short break between battles? In *fourth international Workshop on Factory Communication Systems.* Pages 73–80. IEEE. 2002.

Feng, X., Shen, J. and Fan, Y. REST: An Alternative to RPC for Web Services Architecture. In *Proceedings of the first International Conference on Future Information Networks.* Pages 7-10. IEEE. 2009.

Fernbach, A., Granzer, W. and Kastner W. Interoperability at the Management Level of Building Automation Systems: A Case Study for BACnet and OPC UA. In *Proceedings of the Emerging Technologies & Factory Automation.* Pages 1-8. IEEE. 2011.

Fielding, R. Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation. University of California, Irvine. 2000.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616. Internet Engineering Task Force. 1999.

Firouzian, A., Asghar, Z., Tervonen, J., Pulli, P. and Yamamoto, G. Conceptual Design and Implementation of Indicator-based Smart Glasses: A navigational device for remote assistance of senior citizens suffering from memory loss. In *Proceedings of the ninth International Symposium on Medical Information and Communication Technology.* Pages 153-156. IEEE. 2015.

Frank, B. (ed.) Open Building Information eXchange (oBIX) Version 1.0. Committee Specification 01. OASIS. 2006. http://www.oasis-open.org/committees/download.php/21462/obix-1.0-cs-01.zip

García-Valls, M., Basanta-Val, P. and Estévez-Ayres, I. A component model for homogeneous implementation of reconfigurable service-based distributed real-time applications. In *Proceedings of the tenth Annual International Conference on New Technologies of Distributed Systems.* Pages 267-272. IEEE. 2010

Gemmill, C. (ed.). OASIS Open building information exchange (oBIX) Version 1.1 Committee Specification Draft 02 / Public Review Draft 02. December 2013. http://docs.oasis-open.org/obix/obix/v1.1/ csprd02/obix-v1.1-csprd02.pdf

Guarino, N., Oberle, D. and Staab, S. What Is an Ontology? In *Handbook on Ontologies.* Pages 1-17. Springer. 2009.

Guinard, D. A Web of Things Application Architecture - Integrating the Real-World into the Web. Doctoral Dissertation. Eidgenössische Technische Hochschule Zürich. 2011.

Guinard, D. and Trifa, V. Towards the web of things: Web mashups for embedded devices. In *Proceedings of the International World Wide Web Conference*. ACM. 2009.

Guinard, D., Trifa, V., Karnouskos, S., Spiess, P. and Savio, D. Interacting with the SOA-based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. In *IEEE Transactions on Services Computing.* Vol. 3. Pages 223-235. IEEE. 2010a.

Guinard, D., Trifa, V., Pham, T. and Liechti, O. Towards Physical Mashups in the Web of Things. In *Proceedings of the sixth International Conference on Networked Sensing Systems.* IEEE. 2009.

Guinard, D., Trifa, V. and Wilde, E. A Resource Oriented Architecture for the Web of Things. In *Proceedings of the second International Conference on Internet of Things.* Pages 1-8. IEEE. 2010b.

Hadlich, T. Providing device integration with OPC UA. In *Proceedings of the Industrial Informatics.* Pages 263-268. IEEE. 2006.

Han, D.-M. and Lim, J.-H. Design and Implementation of Smart Home Energy Management Systems based on ZigBee. In *IEEE Transactions on Consumer Electronics.* Vol. 56. Pages 1417-1425. IEEE. 2010.

Harper, R. Inside the Smart Home: Ideas, Possibilities and Methods. Inside the smart home. Vol. 1. Pages 1-13. Springer. 2003.

Heflin, J. and Hendler J. Semantic Interoperability on the Web. In *Proceedings of Extreme Markup Languages.* 2000.

Herman, I., Adida, B., Sporny, M. and Birbeck, M. RDFa 1.1 Primer - Third Edition - Rich Structured Data Markup for Web Documents. Working Group Note. W3C. 2015.

Hert, M., Reif, G. and Gall, H. C. Updating Relational Data via SPARQL/Update. In *Proceedings of the 13th International Conference on Database Theory and 13th International Conference on Extending Database Technology.* ACM. 2010.

Heuer, J. and Raggett, D. W3C Workshop on the Web of Things - Enablers and services for an open Web of Devices. Workshop Report. W3C. 2014.

Hevner, A. and Chatterjee, S. Design Research in Information Systems. In *Integrated Series in Information Systems.* Vol. 22. Springer. 2010.

Hevner, A., March, S. T., Park, J. and Ram, S. Design Science in Information Systems Research. In *MIS quarterly.* Vol. 28. Pages 75-105. University of Minnesota. 2004.

Hickson, I. (ed.) HTML Microdata. Working Group Note. W3C. 2013.

Horrocks, I., Ruttenberg, A., Hawke, S., Herman, I. *et al.* OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation. W3C. 2012.

Hui, J. W. and Culler, D. E. Extending IP to low-power, wireless personal area networks. In *IEEE Internet Computing*. Vol. 12. Pages 37-45. IEEE. 2008.

Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M. and Kettula, S. MuseumFinland—Finnish museums on the semantic web. In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*. Vol. 3. Pages 224-241. Elsevier. 2005.

Ihasalo, H. Transforming building automation data into building performance metrics-design, implementation and evaluation of use of a performance monitoring and management system. Doctoral Dissertation. Aalto University. 2012.

Ikeda, S., Asghar, Z., Hyry, J., Pulli, P., Pitkanen, A. and Kato, H. Remote Assistance Using Visual Prompts for Demented Elderly in Cooking. In *Proceedings of the fourth International Symposium on Applied Sciences in Biomedical and Communication Technologies.* ACM. 2011.

Jahn, M., Jentsch, M., Prause, C. R., Pramudianto, F., Al-Akkad, A. and Reiners, R. The Energy Aware Smart Home. In *Proceedings of the fifth International Conference on Future Information Technology.* Pages 1-8. IEEE. 2010.

Järvinen, H. An Interoperable Equipment Server for Building Automation Systems. Master's Thesis. Helsinki University of Technology. 2007.

Järvinen, H., Bernardet, U. and Verschure, P. F. M. J. Interaction Mapping affects Spatial Memory and the Sense of Presence when navigating in a Virtual Environment. In *Proceedings of the fifth International Conference on Tangible, Embedded, and Embodied Interaction.* Pages 321-324. ACM. 2011.

Järvinen, H. and Garcia-Gasulla. D. Position paper: An agent-oriented architecture for building automation systems applied to assistive technologies. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems.* Pages 71-80. IFAAMAS. 2012.

Kaila, L., Mikkonen, J., Vainio, A.-M. and Vanhala, J. The eHome - a Practical Smart Home Implementation. In *Proceedings of the first Workshop on Pervasive Computing @ Home.* Pages 9-15. 2008.

Kaldeli, E., Warriach, E. U., Bresser, J., Lazovik, A. and Aiello, M. Interoperation, Composition and Simulation of Services at Home. In *Proceedings of the eighth International Conference on Service-Oriented Computing.* Pages 167-181. Springer. 2010.

Kaldeli, E., Warriach, E. U., Lazovik, A. and Aiello, M. Coordinating the Web of Services for a Smart Home. In *ACM Transactions on the Web.* Vol. 7. ACM. 2013.

Kamilaris, A. Enabling Smart Homes using Web Technologies. Doctoral Dissertation. University of Cyprus. 2012.

Kamilaris, A., Pitsillides, A. and Trifa, V. The Smart Home meets the Web of Things. In *International Journal of Ad Hoc and Ubiquitous Computing.* Vol. 7. Pages 145-154. Inderscience. 2011.

Kastner, W., Neugschwandtner, G., Soucek, S. and Newmann, H. M. Communication systems for building automation and control. In *Proceedings of the IEEE*. Vol. 93. Pages 1178-1203. IEEE. 2005.

Khare, K. and Çelik, T. Microformats: A Pragmatic Path to the Semantic Web. In *Proceedings of the 15th international conference on World Wide Web.* Pages 865-866. ACM. 2006.

Kim, D., Lee, J. and Kwon, W. H. Design and Implementation of Home Network Systems using UPnP Middleware for Networked Appliances. In *IEEE Transactions on Consumer Electronics.* Vol. 48. Pages 963-972. IEEE. 2002.

Kim, E. and Choi, J. An Ontology based Context Model in a Smart Home. In *Proceedings of the international Conference on Computational Science and Its Applications.* Pages 11–20. Springer. 2006.

Kim, J.-T., Oh, Y.-J., Lee, H.-K., Paik, E.-H. and Park, K.-R. Implementation of the DLNA Proxy System for Sharing Home Media Contents. In *IEEE Transactions on Consumer Electronics.* Vol. 53. Pages 139-144. IEEE. 2007.

Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B. and Spasojevic, M. People, Places, Things: Web Presence for the Real World. Vol. 7. Pages 365-376. Springer. 2002.

Kivimäki, T., Kölndorfer, P., Vainio, A.-M., Pensas, H., Vuorela, T., Garschall, M. and Vanhala, J. User Interface for Social Networking Application for the Elderly. In *Proceedings of the sixth*

International Conference on Pervasive Technologies Related to Assistive Environments. ACM. 2013.

Kleimola, J. and Vuorimaa, P. RESTful Declarative Framework for Extended Home Entertainment Networks. In *Proceedings of the fifth Consumer Communications and Networking Conference.* Pages 311-315. IEEE. 2008.

Knauth, S., Kistler, R., Kaslin, D. and Klapproth, A. SARBAU towards highly self-configuring IP-fieldbus based building automation networks. In *Proceedings of the International Conference on Advanced Information Networking and Applications*. Pages 713-717. IEEE. 2008.

KNX. KNX: The worldwide STANDARD for home and building control. Marketing presentation. KNX Association International. 2015.

Koskela, T., Väänänen-Vainio-Mattila, K. Evolution towards smart home environments: empirical evaluation of three user interfaces. In *Personal and Ubiquitous Computing.* Vol. 8. Pages 234-240. Springer. 2004.

Kubicek, H., Cimander, R. and Scholl, H. J. Layers of Interoperability. In *Organizational Interoperability in E-Government.* Pages 85-96. Springer. 2011.

Lääkkö, M., Firouzian, A., Tervonen, J., Yamamoto, G. and Pulli, P. Attention Control and Eyesight Focus for Senior Citizens. In *Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality. Lecture Notes in Computer Science.* Vol. 8526. Pages 309-315. Springer. 2014.

Lai, C.-F., Huang, Y.-M. and Chao, H.-C. DLNA-Based Multimedia Sharing System for OSGI Framework With Extension to P2P Network. *In IEEE Systems Journal.* Vol. 4. Pages 262-270. IEEE. 2010.

Lazovik, A., Kaldeli, E., Lazovik, E. and Aiello M. Planning in a Smart Home: Visualization and Simulation. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling.* AAAI Press. 2009.

Lee, C., Nordstedt, D. and Helal, S. Enabling Smart Spaces with OSGi. In *IEEE Pervasive Computing.* Vol. 2. Pages 89-94. IEEE. 2003.

Lee, W. S. and Hong, S. H. Implementation of a KNX-ZigBee Gateway for Home Automation. In *Proceedings of the 13th International Symposium on Consumer Electronics.* Pages 545-549. IEEE. 2009.

Leiner, B. M., Cole, R., Postel, J. and Mills, D. The DARPA Internet Protocol Suite. In Communications Magazine. Vol. 23. Pages 29-34. IEEE. 1985.

Levermore, G. J., Building energy management systems: applications to low-energy HVAC and natural ventilation control. Taylor and Francis. 2000.

Linthicum, D. Next generation application integration: from simple information to Web services. Addison-Wesley Longman. 2003.

Liu, F., Wang, G. and Chou W. Web Service for Distributed Communication Systems. In *Proceedings of the International Conference on Service Operations and Logistics, and Informatics.* Pages 1030-1035. IEEE. 2006.

Maile, T., Fischer, M. and Huijbregts, R. The vision of integrated IP-based building systems. In *Journal of Corporate Real Estate*. Vol. 9. Pages 125-137. Emerald Group Publishing Limited. 2007.

Manola, F., Miller, E. and McBride, B. RDF Primer. W3C Recommendation. W3C. 2004.

Månsson, L.-G. and McIntyre, D. A Summary of IEA Annexes 16 and 17 - Building Energy Management Systems. Technical Synthesis Report. International Energy Agency. 1997.

Mäyrä, F., Soronen, A., Koskinen, I., Kuusela, K., Mikkonen, J., Vanhala, J. and Zakrzewski, M. Probing a Proactive Home: Challenges in Researching and Designing Everyday Smart Environments. In *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments.* Vol. 2. Pages 158-186. Agora Center. 2006.

Miori, V., Russo, D. and Aliberti, M. Domotic technologies incompatibility becomes user transparent. In *Communications of the ACM*. Vol. 53. Pages 153-157. ACM. 2010.

Miori, V., Tarrini, L., Manca, M. and Tolomei, G. An open standard solution for domotic interoperability. In *IEEE Transactions on Consumer Electronics*. Vol. 52. Pages 97-103. IEEE. 2006.

Mozer, M. C. The neural network house: An environment hat adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium of Intelligent Environments.* Pages 110-114. 1998.

Neilson, C., Isler, B., Osborne, M. *et al.* BACnet - A Data Communication Protocol for Building Automation and Control Networks. ANSI/ASHRAE Standard 135-2012. ASHRAE. 2012.

Neugschwandtner, M., Neugschwandtner, G. and Kastner, W. Web Services in Building Automation: Mapping KNX to OBIX. In *Proceedings of the Fifth International Conference on Industrial Informatics.* Vol. 1. Pages 87-92. IEEE. 2007.

Newcomer, E. and Lomow, G. Understanding SOA with Web Services (Independent Technology Guides). Addison-Wesley. 2004.

Nowack, B. CONFOTO: Browsing and annotating conference photos on the Semantic web. In *Web Semantics: Science, Services and Agents on the World Wide Web.* Vol. 4. Pages 263-266. Elsevier. 2006.

OPC Foundation. OPC Unified Architecture Specification Part 1: Overview and Concepts. Release 1.02. Industry Standard Specification. OPC Foundation. 2012.

OSGi Alliance. About the OSGi Service Platform, technical whitepaper, revision 4.1. 2007.

Oxford Dictionaries. Oxford University Press. Available: http://www.oxforddictionaries.com/definition/english/smart-home. Accessed Aug 2014.

Parks Associates. DLNA Market Overview Report. Parks Associates for DLNA. 2013.

Pautasso, C. and Wilde, E. Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. In *Proceedings of the 18th international conference on World Wide Web.* Pages 911-920. ACM. 2009.

Pautasso, C., Zimmermann, O. and Leymann, F. Restful web services vs. “big” web services: making the right architectural decision. In *Proceedings of the 17th international conference on World Wide Web.* Pages 805-814. ACM. 2008.

Pensas, H., Kivimäki, T., Vainio, A.-M., Konakas, S., Costicoglou, S., Kölndorfer, P., Summanen, K., Moisio, H. and Vanhala, J. Building a Client-Server Social Network Application for Elders and Safety Net. In *Proceedings of International Conference on Making Sense of Converging Media.* Pages 310-312. ACM. 2013.

Perumal, T., Ramli, A. R., Leong, C. Y. SOA-Based Framework for Home and Building Automation Systems (HBAS). In *International Journal of Smart Home*. Vol. 8. Pages 197-206. 2014

Perumal, T., Ramli, A. R., Leong, C. Y., Mansor, S. and Samsudin, K. Interoperability for smart home environment using web services. In *International Journal of Smart Home*. Pages 1-16. 2008.

Perumal, T., Ramli, A. R., Leong, C. Y., Samsudin, K. and Mansor, S. Middleware for heterogeneous subsystems interoperability in intelligent buildings. In *Automation in Construction*. Vol. 19. Pages 160-168. Elsevier. 2010.

Pulli, P., Hyry, J., Pouke, M. and Yamamoto, G. User interaction in smart ambient environment targeted for senior citizen. In *Medical & Biological Engineering & Computing.* Vol. 50. Pages 1119-1126. Springer. 2012.

Qiao, B., Liu, K. and Guy, C. A multi-agent system for building control. In *Proceedings of the international conference on Intelligent Agent Technology*. Pages 653-659. IEEE. 2006.

Ramos, C., Augusto, J. C. and Shapiro, D. Ambient Intelligence - the Next Step for Artificial Intelligence. In *IEEE Intelligent Systems.* Vol. 23. Pages 15-18. IEEE. 2008.

Rashidi, P. and Cook, D. J. Keeping the Resident in the Loop: Adapting the Smart Home to the User. In *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans.* Vol. 39. Pages 949-959. IEEE. 2009.

Schickhuber, G. and McCarthy, O. Distributed fieldbus and control network systems. In *Computing & Control Engineering Journal.* Vol. 8. Pages 21-32. IET. 1997.

Sánchez, D., Isern, D., Rodríguez-Rozas, Á. and Moreno, A. Agent-based platform to support the execution of parallel tasks. In *Expert Systems with Applications.* Vol. 38. Pages 6644-6656. Elsevier. 2011.

Santos, R., Carreira, P. Service Oriented Development of Building Energy Management Systems: An Architecture Blueprint. In *Proceedings of the third International Conference on Smart Grids and Green IT Systems.* SCITEPRESS. 2014.

Seaborne, A. and Manjunath, G. SPARQL/Update: A language for updating RDF graphs. Technical Report. Hewlett-Packard Development Company, L.P. 2007.

Shafiq, M. O., Ding, Y. and Fensel, D. Bridging Multi Agent Systems and Web Services: towards interoperability between Software Agents and Semantic Web Services. In *Proceedings of the tenth International Enterprise Distributed Object Computing Conference.* IEEE. 2006.

Shelby, Z. and Bormann, C. 6LoWPAN: The wireless embedded Internet. In *WILEY series in Communications Networking & Distributed Systems*. Vol. 43. Wiley Publishing. 2009.

Shelby, Z., Hartke, K. and Bormann, C. The Constrained Application Protocol (CoAP). RFC 7252. Internet Engineering Task Force. 2014.

Song, H., Kim, D., Lee, K. and Sung, J. UPnP-Based Sensor Network Management Architecture. In *Proceedings of the International Conference on Mobile Computing and Ubiquitous Networking.* IEEE. 2005.

Tervonen, J., Asghar, M. Z., Yamamoto, G. and Pulli, P. A Navigation Aid for People Suffering from Dementia Using a Body Worn Laser Device. In *Proceedings of the seventh International Symposium on Medical Information and Communication Technology.* Pages 178-182. IEEE. 2013.

Tokunaga, E., Ishikawa, H., Kurahashi, M., Morimoto, Y. and Nakajima, T. A framework for connecting home computing middleware. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*. Pages 765-770. IEEE. 2002.

Vainio, A.-M., Valtonen, M. and Vanhala, J. Proactive Fuzzy Control and Adaptation Methods for Smart Homes. In *Intelligent Systems.* Vol. 23. Pages 42-49. IEEE. 2008.

Valtchev, D. and Frankov, I. Service gateway architecture for a smart home. In *IEEE Communications Magazine.* Vol. 40. Pages 126-132. IEEE. 2002.

van der Veer, H. and Wiles, A. Achieving Technical Interoperability – the ETSI Approach. ETSI White Paper No.3. ETSI. 2008.

Vanhala, J., Mäyrä, F. and Koskinen, I. Living in Metamorphosis: Proactive Computing in the Home Environment. In *Interactions.* Vol. 12. Pages 28-31. ACM. 2005.

Venkitaraman, N. Wide-Area Media Sharing with UPnP/DLNA. In *Proceedings of the fifth IEEE Consumer Communications and Networking Conference.* Pages 294-298. IEEE. 2008.

Vuorimaa, P., Harmo, P., Hämäläinen, M., Itälä, T. and Miettinen, R. Active Life Home: a Portal-Based Home Care Platform. *In Proceedings of the fifth International Conference on PErvasive Technologies Related to Assistive Environments.* ACM. 2012.

Wagner, A., Anicic, D., Stühmer, R., Stojanovic, N. and Harth, A. Linked data and complex event processing for the smart energy grid. In *Proceedings of Linked Data in the Future Internet at the Future Internet Assembly*. 2010.

Wang, S., Xu, Z., Cao, J. and Zhang, J. A middleware for web service-enabled integration and interoperation of intelligent building systems. In *Automation in Construction*. Vol. 16. Pages 112-121. Elsevier. 2007.

Weerawarana, S., Curbera, F., Leymann, F., Storey, T. and Ferguson, D. F. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Prentice Hall PTR. 2005.

Wilde, E. Putting Things to REST. School of Information Report. University of California, Berkeley. 2007.

Witte, L. and Telkamp, G. Domotics and Infomobility in *Proceedings of the ASK-IT Project. In ASK-IT International Conference*. 2006.

Woods, W. A. What's in a Link: Foundations for Semantic Networks. In Representation and understanding: Studies in cognitive science. Pages 35-82. Academic Press. 1975.

Wooldridge, M. and Jennings, N. R. Intelligent agents: Theory and practice. In The Knowledge Engineering Review. Vol. 10. Pages 115-152. Cambridge University Press. 1995.

Wu, C., Liao, C. and Fu, L. Service-oriented smart-home architecture based on OSGi and mobile-agent technology. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. Vol. 37. Pages 193-205. IEEE. 2007.

Xu, J., Lee, Y.-H., Tsai, W.-T., Li, W., Son, Y.-S., Park, J.-H. and Moon, K.-D. Ontology-based Smart Home Solution and Service Composition. In *Proceedings of the International Conference on Embedded Software and Systems.* Pages 297-304. IEEE. 2009.

Yamazaki T. Beyond the smart home. In *Proceedings of the International Conference on Hybrid Information Technology.* Vol. 2. Pages 350-355. IEEE. 2006.

Youngblood, G. M., Heierman, E. O., Holder, L. B. and Cook, D. J. Automation intelligence for the smart environment. In *Proceedings of the International Joint Conference On Artificial Intelligence.* Pages 1513-1514. Morgan Kaufmann Publishers Inc. 2005.

Zamora-Izquierdo, M. A., Santa, J. and Gómez-Skarmeta, A.F. An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. In *Pervasive computing.* IEEE. 2010.

Zeeb, E., Bobek, A., Bohn, H. and Golatowski, F. Service-oriented architectures for embedded systems using devices profile for web services. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops.* Vol. 1. Page 956-963. IEEE. 2007.

Zhang, D., Gu, T. and Wang, X. Enabling Context-aware Smart Home with Semantic Web Technologies. In *Journal of Human-friendly Welfare Robotic Systems.* Pages 12-20. 2005.

Zhang, L. and Zhang J. SOA reference architecture. Web services research for emerging applications: Discoveries and trends. Information Science Reference. Pages 1-15. IGI Global. 2010.

Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y. and Pan, Y. Minerva: A Scalable OWL Ontology Storage and Inference System. In *Proceedings of the first Asian Semantic Web Conference.* Pages 429-443. Springer. 2006.

Zimmermann, H. OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. In IEEE Transactions on Communications. Vol. 28. Pages 425-432. IEEE. 1980.

Aalto-DD 183/2015

ISBN 978-952-60-6509-0 (printed)
ISBN 978-952-60-6510-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS