

Artificial intelligence and Internet of Things in a “smart home” context

A Distributed System Architecture

Lynggaard, Per

Publication date:
2014

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Lynggaard, P. (2014). Artificial intelligence and Internet of Things in a “smart home” context: A Distributed System Architecture (1 ed.). Department of Electronic Systems, Aalborg University.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

**Artificial intelligence and Internet of Things in a “smart home” context:
A Distributed System Architecture**

**A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of
Doctor of Philosophy**

Per B. Lynggaard

**Center for Communication, Media and Information Technologies (CMI)
Aalborg University Copenhagen, Denmark**



AALBORG UNIVERSITY
DENMARK

Mandatory page in PhD theses:

For any PhD thesis where one or more submitted or published papers are used in the thesis the PhD student should pay close attention to proper citation of the work. If the paper(s) are co-authored the student cannot normally use the papers as if the student was the sole author. To protect the student while still proving some freedom in using material developed during the PhD study a mandatory page including the following must be included:

1. Thesis title:

Artificial intelligence and Internet of Things in a “smart home” context: A Distributed System Architecture

2. Name of PhD student:

Per B. Lynggaard

3. Name and title of supervisor and any other supervisors.

Dr. Henning Olesen, Associate Professor, CMI, Aalborg University Copenhagen, Denmark

Dr. Birger Andersen, Professor, CWSA, Technical University of Denmark (co-supervisor)

4. List of published / submitted papers:

Conference Papers

Lynggaard, (2011), A Smart Home Development Tool combining Simulation, Emulation and Real-World IoT, Conference: AAU / CMI, Internet of Things –Our Environment Becomes Intelligent, 2011 (extended abstract)

Lynggaard, (2013), A Distributed Smart Home System Based on Smart Objects, Conference: AAU / CMI, Developing the Future ICT Infrastructure, 2013 (awaiting publishing: electronic proceedings / relevant journals)

Lynggaard, (2013), Smart Home Wireless Sensor Nodes, Addressing the Challenges using Smart Objects and Artificial Intelligence, WWRF Conference: Wireless Enabled Smart Societies in the 2020s, 2013 (awaiting publishing: meeting proceedings / WWRF Outlook / IEEE Vehicular Technology Magazine).

Journal Papers

Lynggaard, (2012), Distributed Smart Home Activity Recommender System Using Hidden Markov Model Principles, Scientific Journal of the University of Szczecin no. 762 TOM I p359-369. (Published)

Lynggaard, (2012), A Distributed Smart Home Artificial Intelligence System, Scientific Journal of the University of Szczecin no. 763 TOM II p521-531. (Published)

Lynggaard, (2013), Using ORB in a Smart Home Portable User Interface, IEEE Transactions on visualization and computer graphics (Submitted).

Lynggaard, (2013), Smart Home Learning and Activity Prediction using a Simplified Naïve Bayes Framework, IEEE Internet of Things Journal (Submitted).

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is partly based on the submitted or published scientific papers, which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Abstract

The history of smart homes has its source and roots in building and home automation. Today (2013), the home automation area offers remote and timer control of systems and embedded devices such as light, heating, ventilation, entertainment systems, appliances, etc., to improve comfort, convenience, energy efficiency, and security. However, the element of autonomous behavior is lacking, this is where the smart homes come into play.

The smart homes era builds on the progressing maturity of these areas and the Internet of Things evolution, adding artificial intelligence to the home automation field. Smart homes will be based on distributed multi-agent architectures to overcome technological challenges such as immature home intelligence, huge network and central server processing load; and embedded resource usage. At present, smart homes are still in their infancy, and they only exist in the form of laboratory experiments such as living labs.

To pave the way for future smart homes, more research on *distributed system architecture*, *smart objects*, and *distributed artificial intelligence frameworks* in a smart home context is needed. State-of-the-art research projects have so far only shown a centralized approach.

This work contributes to the field by developing a new *distributed framework* for smart homes, which comprises a real-time distributed system with autonomous behavior, parallel processing, context awareness, and node communication. The proposed solution takes into account significant technological challenges such as real-time learning, detection probability, battery lifetime, network communication, and embedded processing power. In particular, it introduces a novel approach to adapt and distribute the artificial intelligence to match the distributed system architecture in the smart home.

A multi-agent smart object model is provided to support the artificial intelligence framework with a new distributed architecture. This model focuses on the embedded resources, the sensor frameworks, and the employed algorithms and leads to considerable savings in battery power consumption, processing resources and network load. It is part of the developed hierarchical distributed smart home system architecture which consists of a Low-Level and a High-Level

Smart Home System (LL-SHS and HJ-SHS). The layout structure and architecture of this model are derived and part of it is simulated.

The methodology used in this work is based on a combination of established theories, i.e., the “hybrid imagination”, the “truthiness”, and the theory of induction and deduction. These theories are combined with an iterative process model, which is used to support the research process. This research process uses: technical and mathematical analysis based on library and online searches; objects oriented process modeling; model and code implementation; and test by using empiric dataset from other independent researchers. The research results are presented in a “research view” and a “presentation view” which covers the research process.

The proposed framework features a simplified implementation, high flexibility, learning and prediction on the fly, advanced temporal prediction, standalone capability, limited processing resources, and easy integration with the smart objects. Significant parts of the framework are simulated to validate their performance, and it is shown that the performance is comparable to state-of-the-art smart home technologies.

In a larger perspective, the proposed framework supports and facilitates the coming era of Internet of Things. The distributed approach and elements of the framework can be applied in many related areas, such as ambient-assisted living, e.g., assisting elderly people to stay longer at home, intelligent transportation systems, and many other sustainable solutions based on ICT.

Resume

Smart homes er en videreudvikling af hjemme- og bygningsautomationsområderne og tilbyder avancerede hjemmeautomations-funktionaliteter. Således tilbyder smart homes i dag (2013) fjernstyring og timerkontrol af indlejrede systemer såsom: lys, varme, ventilation, underholdningssystemer og apparater for at øge komfort, bekvemmelighed, energieffektivitet og sikkerhed. Imidlertid mangler dagens automationssystemer det ”intelligente” element, som smart homes tilbyder.

Smart homes udvikles i takt med, at området for hjemmeautomationssystemer modnes og gradvis udstyres med kunstig intelligens. Et væsentligt element i denne udvikling er ’Internet of Things’-evolutionen.

Fremtidens ’smart home’-teknologier vil blive baseret på en distribueret multiagentstruktur for at kunne håndtere de teknologiske udfordringer. Disse udfordringer omfatter: udvikling af intelligent adfærd, bearbejdning af store mængder data i netværk, håndtering af belastning og ressourceforbrug på centrale servere. Nutidens smart homes er ikke i stand til at håndtere disse teknologier, da smart homes kun eksisterer som laboratorieforsøg.

Mere forskning er nødvendig for at smart homes kan blive en realitet. Således behøves der forskning i distribueret systemarkitektur, intelligente objekter og i distribueret kunstig intelligens. Disse forskningsområder dækkes ikke af dagens state-of-the-art forskningsprojekter, som anvender centraliserede principper.

Dette forskningsprojekt bidrager til ’smart home’-forskningsområdet ved at udvikle en ny distribueret arkitektur, som tilbyder: et realtids-distribueret system med autonom adfærd, parallel databehandling, kontekstbevidsthed og distribueret netværkskommunikation. Den udviklede løsning håndterer de teknologiske udfordringer, som er: realtidslæring med øget sandsynlighed for detektion, forlængelse af batterilevetid, reduktion af behov for netværkskommunikation og regnekraft. Et særligt punkt er introduktion af unikke principper for at tilpasse og distribuere den kunstige intelligens, så den matcher den distribuerede systemarkitektur.

Endvidere udvikles en distribueret multi-agent 'smart objekt'-model, som understøtter den distribuerede kunstige intelligens med en ny arkitektur. Denne model fokuserer på indlejrede ressourcer, sensorarkitektur og anvendte algoritmer. Modellen medfører betydelige besparelser i strømforbrug (batterilevetid), processorressourcer og netværksbelastning. Endvidere er den en delmængde af den hierarkiske 'smart home'-arkitektur, som består af en lav- og højniveaudel. Arkitekturen og strukturen for denne model er udviklet og simuleret.

Dette forskningsprojekt anvender en metodisk fremgangsmåde, som er baseret på en kombination af etablerede teorier, såsom "hybrid imagination", "truthiness" og teoriene omkring induktion og deduktion. Disse teorier er kombineret med en iterativ procesmodel. Således anvender forskningsprocessen teknisk og matematisk analyse baseret på litteratursøgning, objektorienteret procesmodellering, model- og kodeimplementering og test ved hjælp af empiriske data fra uafhængige forskere. Forskningsresultaterne præsenteres i et forsknings- og et præsentationsperspektiv.

Resultaterne fra dette forskningsprojekt omfatter en forenklet fleksibel 'smart home'-arkitektur og implementering, som tilbyder: læring og forudsigelse i realtid, avancerede forudsigelser af komplekse tidsmæssige sekvenser, selvstændige objekter, en reduktion af processorressourcer og nem integration med smart-objekter. Væsentlige dele af arkitekturen er simuleret for at validere de opnåede resultater og sammenligne disse med state-of-the-art 'smart home'-teknologier frembragt af uafhængige forskere.

Set i et større perspektiv understøtter den udviklede arkitektur den kommende 'Internet of Things'-evolution. Endvidere er det muligt at anvende den distribuerede tilgang og de arkitektoniske strukturer i mange beslægtede områder, såsom: at hjælpe ældre til at blive længere hjemme, intelligente transportsystemer og bæredygtige løsninger baseret på IKT, m.v.

Acknowledgements

This thesis represents research that has been carried out through the last three years in the cross-disciplinary center for Communication Media Information technologies (CMI). CMI is organized under the Department of Electronic Systems at Aalborg University, Denmark. The work has been funded by CMI.

I would like to express my gratitude to my supervisor, Associate Professor Henning Olesen, Aalborg University and co-supervisor, Professor Birger Andersen, Technical University of Denmark, whose help, guidance and encouragement I will always remember and appreciate. I am also grateful to the reviewer of my thesis Associate Professor Reza Tadayoni, Aalborg University for valuable comments on this and to Trilingual Secretary Anette Bysøe for proofreading the manuscript. Thanks also go to my colleagues at CMI, who have supported me with advice and encouragement.

Finally, I would like to thank my family, friends and relatives for their enduring support, especially my wife Conny and my children Mia and Mikkel, to whom I dedicate this thesis, for being there and supporting me, even when I was working during vacations, weekends and nights.

Per B. Lynggaard

August 2013.

Table of contents

1	Introduction	1
1.1	Problem Motivation	1
1.2	Research Questions	6
1.3	Delimitations and Scope	7
1.4	Thesis Outline	9
2	Research Methodology, Approach and Design	12
2.1	The “Hybrid-Imagination” Perspective	12
2.2	The Scientific Work and “Truthiness” Perspective	14
2.3	Generalizing Theories by Using Induction and Deduction	16
2.4	Selected Methodical Approach	17
2.4.1	The Methodological Process Model	17
2.4.2	Research Methodology	20
2.4.3	Presentation Methodology	22
2.5	Summary	26
3	From Building Automation to Smart homes	27
3.1	Building Automation	28
3.2	Home Automation	32
3.3	Smart Homes Overview	34
3.3.1	Users Perspective	35
3.3.2	Discussion of Smart Home Barriers	36
3.3.3	Artificial Intelligence in Smart Homes	37
3.3.4	The Research Frontiers of Smart Homes	38
3.4	Internet of Things	40
3.5	Related Areas and Research Activities	42
3.5.1	Related Projects	42
3.5.2	Related Technologies and Research Areas	42
3.5.3	Related Industry Organizations	43
3.6	Summary	43
4	Distributed Systems and Smart Homes	45
4.1	Distributed Systems in General	46

4.2	Distributed Systems in a Smart Home Context	48
4.3	Centralized and Distributed Smart Homes	51
4.3.1	Centralized, Agent and Cloud Based Systems	51
4.3.2	Smart Home Device Communication	53
4.3.3	A Simple Scenario	56
4.4	Sensors and Actuators	57
4.4.1	Sensors	58
4.4.2	Device Power Consumption	59
4.4.3	Transmit Only Nodes	60
4.4.4	Sensor Clustering	62
4.4.5	Smart Objects / Agents	64
4.5	Network Topologies	67
4.5.1	Routing in Smart Homes	68
4.5.2	Factors Affecting the Network Choice	69
4.5.3	Wired Smart Home Networks	71
4.5.4	Wireless Smart Home Networks	72
4.5.5	Algorithms Used in Smart Home Networks	73
4.5.6	Wireless Network Types	76
4.6	Summary	79
5	Artificial Intelligence: Algorithms and Models	80
5.1	AI Definitions and Applications	80
5.1.1	Artificial Intelligence Definitions	80
5.1.2	Artificial Intelligence Application Area	82
5.2	AI and Pattern Recognition	83
5.3	Probabilistic Models in Smart Homes	85
5.4	Non Probabilistic Models in Smart Homes	86
5.5	Discussion of Selected Smart Home Models	87
5.5.1	Sensor Events and Actions, a Notation	87
5.5.2	Naïve Bayes Probabilistic Model	89
5.5.3	Hidden Markov Probabilistic Model	92
5.5.4	Conditional Random Field Probabilistic Model	95

5.6	Application of Probabilistic Models in Smart Homes	96
5.6.1	The Use of NB, HMM and CRF Classifiers.....	96
5.6.2	Comparison of NB, HMM and CRF Classifiers	97
5.7	Distributed Intelligence in Smart Homes	98
5.8	Summary.....	99
6	Centralized and Agent-Based Models	100
6.1	Analysis of System Architecture	100
6.1.1	Centralized Smart Home Model.....	100
6.1.2	Agent-Based Smart Home Model.....	107
6.1.3	Comparing the Centralized and the Agent-Based Systems.....	111
6.2	AI Frameworks	114
6.3	Summary.....	116
7	Distributed System Architecture for Smart Homes	117
7.1	Deriving the System Architecture	118
7.1.1	Architecture Development.....	118
7.1.2	Proposed Smart Home System Model.....	120
7.2	Smart Object Model.....	122
7.2.1	Smart Object Layered Model	123
7.2.2	Software Architecture.....	123
7.2.3	Smart Object Communication Architecture	128
7.2.4	Smart Object Implementation.....	133
7.3	Communication Between Smart Objects.....	136
7.3.1	A Simplified Scenario	136
7.3.2	Top Level Communication Model	138
7.3.3	The Smart Home Core.....	140
7.3.4	Sensor and Smart Object Clustering.....	142
7.3.5	The Transmit-Only Sensors.....	146
7.4	Problems Evaluated	148
7.5	Summary.....	153
8	Enhanced Framework Using Artificial Intelligence	155
8.1	Research Question and AI Framework.....	155

8.1.1	Relation to Research Question	155
8.1.2	The AI Framework; an Overview	156
8.2	The Smart Home Model; a Mathematical Derivation	159
8.3	Process model of High-Level Smart Home System	166
8.3.1	Object Process Model	167
8.3.2	Implementation Model	168
8.3.3	HL-SHS Learning	172
8.3.4	HL-SHS Prediction	173
8.4	Simulation and Performance of the HL-SHS	176
8.4.1	Simulation Model	176
8.4.2	Smart Home Simulation Example I	178
8.4.3	Smart Home Simulation Example II	185
8.4.4	HL-SHS Summary	187
8.5	Process Model of Low-Level Smart Home System	188
8.5.1	Object Process Model	188
8.5.2	LL-SHS Detailed Overview	189
8.5.3	Sensor Objects for LL-SHS	190
8.5.4	Event Buffer for LL-SHS	192
8.5.5	LL-SHS Activity Learning	193
8.5.6	LL-SHS Activity Prediction	194
8.6	Simulation and Performance of the LL-SHS	196
8.6.1	Simulation Model	196
8.6.2	Smart Home Simulation Example I	197
8.6.3	Smart Home Simulation Example II	200
8.6.4	LL-SHS Summary	202
8.7	Smart Object Model, revisited	203
8.8	Simulated Performance of the Smart Object Model	205
8.9	Summary	213
9	Overview of the Final Concept and Contributions	214
9.1	A High-Level Overview	214
9.1.1	The System	214

9.1.2	Application Examples	216
9.1.3	Other Application Examples	220
9.2	A Detailed Overview	221
9.2.1	User Interaction	222
9.2.2	Smart Objects and Sensors	223
9.2.3	LL-SHS.....	224
9.2.4	HL-SHS	224
9.3	Contributions	225
9.4	Interpretation of Contribution and Results	233
9.4.1	Overview of This Work and Selected Contributions.....	233
9.4.2	Question 1: Model selection	234
9.4.3	Question 2: Results of the simulations	239
9.4.4	Question 3: General system architecture	244
10	Conclusion and Outlook	247
10.1	Answering the Research Question.....	247
10.2	Reflections	249
10.3	Discussion and Future Challenges.....	249
10.3.1	Smart Homes and Users	249
10.3.2	Pushing the Smart Homes Forward.....	251
10.3.3	Future Smart Object Challenges.....	251
10.3.4	Future Artificial Intelligence Challenges	252
11	Bibliography	254

Abbreviations and Acronyms

AAL	Ambient Assisted Living
AI	Artificial Intelligence
ASBR	Adaptive Scenario Based Reasoning
ASH	Agent-based Smart Home
BAS	Building Automation System
CASAS	Center for Advanced Studies in Adaptive Systems, WSU
CBR	Case Based Reasoning
CEBUS	Consumer Electronics Bus
CORBA	Common Object Request Broker
CRF	Conditional Random Field
CSH	Centralized Smart Home
CSMA-CD	Carrier Sense Multiple Access with Collision Detection
DSSS	Direct Sequence Spread Spectrum
FFD	Full Function Device, ZigBee
FIFO	First-In-First-Out
GUI	Graphical User Interface
HL-SHS	High-Level Smart Home System
HMM	Hidden Markov Model
HVAC	Heating, Ventilation and Air Conditioning
i.i.d.	independent and identically distributed
ISM	Industrial Scientific Medical
LBT	Buffer - Look Back Time
LEACH	Low Energy Adaptive Clustering Hierarchy
LL-SHS	Low-Level Smart Home System
MPLAB	A simulator from Microchip
NB	Naïve Bayes
PAN	Personal Area Network
RAID	Redundant Array of Independent Disks
RFID	Radio-frequency identification
RFD	Reduced Function Device, ZigBee

SDR	Software Defined Radio
SHN	Smart Home Network
SHS	Smart Home System
SoC	System on a Chip
S.O.	Smart Object
SPIN	Sensor Protocols for Information via Negotiation
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

1 INTRODUCTION

This chapter provides the problem motivation (Section 1.1), which leads to formulation of the research questions (Section 1.2). By limiting these, the scope of this work is defined (Section 1.3). Finally, the thesis outline is presented (Section 1.4)

1.1 PROBLEM MOTIVATION

Building automation has been a research field for the last two decades and has contributed with many standards, theories and technologies, which have been published and commercialized (Wang, 2010). Throughout the years building automation has developed from performing simple controlled functions such as regulating the heating, ventilation, and air conditioning to handling the changing needs throughout its lifecycle (Sauter & Soucek, 2011). Today (2013), building automation covers an umbrella of network and computerized technologies that are integrated into commonly available building management systems (Wang, 2010), (Sagi & Mijic, 2012).

From this technological era the home automation systems have developed. Thus, from a pragmatic point of view the home automation is a residential expansion of the building automation area (Turner, 2011). The purpose of home automation is to ease life for its residents by controlling mundane functions such as light, ventilation, heat and appliances to improve comfort, convenience, and energy efficiency in the automated homes. This is performed in a non-autonomous way by adding simple remote controlled, timer based and pre-programmable functions. Examples of remotely controllable domestic activities could be heating, lighting, houseplants, entertainment systems, pet feeding, yard watering, and controlling different kinds of domestic robots such as vacuum cleaners. This research area is in focus today where researchers look into optimizing and maturing the technologies for bringing them into general use (Sagi & Mijic, 2012), (Liutkevičius et al., 2011).

The smart home area emerges from the technologies researched and developed in the home automation area and the building automation areas (Alam et al., 2012), (Sagi & Mijic, 2012). Thus, it is considered an extension of these areas where more advanced control features and autonomous behavior are added in the form of artificial intelligence (AI). In addition, it is stimulated by the Internet of Things (IoT) research area (Bandyopadhyay & Sen, 2011). The Internet of Things (IoT) research area provides context awareness, processing capabilities, and communication possibilities to physical things in general. Whereas the smart homes area is a subset of this dealing with homes only; however, this area can benefit from the research, technologies, and functionalities developed in the IoT area.

It is expected that smart homes will have a huge impact on our future lifestyle. They will be able to act “intelligently” and provide services to its user in almost the same way, as a good old-fashioned “butler” would do. Thus, smart homes may communicate about a lot of tasks such as adjusting lights and music level; controlling heating; and changing multimedia settings according to user preferences and mood (CERP-IoT et al., 2010). However, they will also provide services in the form of assisting and supervising areas such as: Ambient Assisted Living (AAL), i.e., elderly to stay longer at home; telemedicine; and energy and pollution savings (Basu et al., 2013), (Chan et al., 2008).

Today (2013), many drivers are fueling this area, especially consumer, AAL, entertainment industry, and green technologies (Alam et al., 2012), (Fedosseev et al., 2011). However, barriers exist in the form of technological limitations, high cost, inflexibility, and poor manageability (Brush et al., 2011). The technological limiting barrier comprises a system architecture challenge in form of the limitations found in the centralized models researched today (2013). They suffer from a huge amount of network traffic, huge processing requirements on a centralized server, network interferences, complications with expansion and setup functionalities, and lack of real-time learning. Additionally, the employed AI frameworks are mainly standard solutions borrowed from the pattern recognition research area without adaptation to the specific problem area.

Research has shown that use of a distributed concept in combination with today’s (2013) technology in the form of cheap embedded computers, fast sensors and advanced networks

can provide the resources needed to overcome these challenges (Silva et al., 2012), (Alam et al., 2012).

To move the smart home research frontiers a distributed smart home architecture must be derived. This architecture needs to deal with the problems and challenges in the centralized approach, the agent-based approach, and make it adaptable and compatible with the future of IoT (Alam et al., 2012), (Silva et al., 2012), (Uckelmann & Harrison, 2011). Additionally, the research carried out in these areas has not adequately addressed how a distributed architecture can be combined with AI elements. Hence, in the past researchers have mostly worked with centralized systems, why limited research has been done in the area of distributed agent-based systems (Alam et al., 2012), (Reinisch et al., 2010), (Section 3.3).

The agent-based *Smart Object*¹ (S.O.) and the IoT research areas contain important distributed technologies for deriving a distributed architecture. These technologies have been discussed repeatedly among researchers; however, only limited research has been done in the key areas such as network load, resource consumption, and ability to support a distributed AI framework (Silva et al., 2012), (Liang et al., 2002). Firstly, the battery power usage (i.e., battery lifetime) in the wireless smart home devices is a problem (Casilari et al., 2010), (Bleda et al., 2012), (Sundmaeker et al., 2010). Secondly, the smart home network interconnecting wireless devices is limited by effects such as transmit power, interferences, retransmission, and other channel phenomenon's (Bleda et al., 2012), (Rashid et al., 2012), (Casilari et al., 2010), (Yao et al., 2010). Finally, the needed processing power for supporting AI systems in smart homes is challenging (Rashid et al., 2012), (Wu & Shao, 2012), (Bandyopadhyay & Sen, 2011).

Supporting a distributed AI framework on an agent-based S.O. embedded platform requires heavy resources such as network bandwidth, processing power, and storage capacity. Thus, to

¹ The term Smart Object (S.O.) covers an object that is based on a small piece of software, which is installed on either an embedded hardware platform or build into an existing product such as a smart TV, a radio, or a toaster. This device is able to: communicate; be context aware; capture and process event from the sensors; and suggest actions based on AI.

overcome these problems research is needed to divide the algorithms into smaller hierarchical units, to simplify the employed algorithms, and to run the algorithms in parallel. Additionally, simplifications in sensor event information and data exchanged with systems out of context need considerations.

A distributed AI processing framework needs to support an embedded microcontroller concept, a mechanism which is able to handle temporal connected activities, and facilitate autonomous S.O.s that are spatially distributed and context aware. Nonetheless, distributed artificial systems in smart homes have not been an active research area (Alam et al., 2012), (Silva et al., 2012), but it is expected that the IoT paradigm provides an AI framework that is useable in smart homes (Chen et al., 2011), (Gershenfeld et al., 2004), (Chen et al., 2013). Additionally, limited research has been done in the area of real-time learning, prediction and adaptation on the fly in smart homes (Cheng et al., 2009).

Most AI research in smart homes deals with user *activity*² recognition and prediction in a centralized form (Kasteren et al., 2008b), (Cook, 2012), (Alam et al., 2012). However, this research provides general useable datasets, which are publicly available (Cook, 2012), (Kasteren et al., 2011). These datasets are usable in distributed AI smart home research and for making a fair comparison between different smart home approaches, architectures and models.

A distributed architecture that uses small IoT based S.O's spatially distributed in smart homes needs to be resource aware, needs to provide a framework for node cooperation, and it needs to cluster the S.O. nodes according to their available resources and their spatial position (Li et al., 2011b), (Khan & Aziz, 2009). However, such a concept provides challenges in form of organization algorithms and information exchange principles, which need to be researched.

The S.O. devices need a communication model that connects the devices and provides access to out of context devices such as user interface and cloud based services (Pensas & Vanhala, 2010). This model must be flexible, must support generic device types, and it must facilitate

² An activity is what the user do, e.g., turning on the TV.

devices that are powered by either a battery or the mains (Shah et al., 2009). A key device category is the transmit-only sensors which are a research area in the WSN context (Zhao et al., 2013), (Blaszczyszyn & Radunovic, 2008). Using this technology in the smart home area provides means for lowering the power consumption in the sensor nodes, which are mainly battery powered. Such a combination has not been explored and needs research in different areas such as modifying existing technologies (ZigBee, 6LoWPAN) and Software Defined Radio (SDR) (Lu & Wu, 2011), (Starsinic, 2010).

Deriving the content and architecture of a distributed S.O. provides challenges in form of software architecture, generic interfaces, and embedded hardware resources (Alam et al., 2012), (Silva et al., 2012). The software architecture can benefit from the research performed in the IoT and WSN areas regarding generic middleware interface frameworks (Arabnia et al., 2010), (Park et al., 2013), (Eisenhauer et al., 2009). Thus, combining different research areas and their technologies moves the limited research performed in the S.O. area forward (Alam et al., 2012). The embedded hardware architecture must offer the needed resources such as interfaces, control logics, event logics, communication logics, support for AI processing, support for small embedded microprocessors, and power resource management (López et al., 2012), (Trevenor, 2012). However, limited research has been performed to derive a S.O. embedded platform concept, whereas some research in the controlled home area has been performed (Alam et al., 2012), (Basu et al., 2013).

The methodology used in this work is based on a combination of established theories. Firstly, the modern “hybrid imagination” theory enables the use of an experimental approach and thereby guides this scientific work (Christensen et al., 2011), (Normann & Jamison, 2011). Secondly, a theory about “truthiness” in research provides the basis for the iterative model used in this work (Brodersen, 2008). Thirdly, a used process model arrives by combining an iterative process model with the theory of induction and deduction (Brodersen, 2008). Finally, this iterative process model is used to support the research phases. Thus, a technical and mathematical analysis based on library and online searches in combination with gained knowledge starts the process. Output from this analysis is captured in objects oriented process models. Vitale selected parts of these models are implemented and tested by using empiric

datasets from other independent researchers. A detailed description of each time-step in the model and a “presentation view” of the research process are provided in section 2.4.

1.2 RESEARCH QUESTIONS

In a nutshell, the challenges are how to derive and combine a distributed system architecture, including its subjects, discussed in section 1.1, with a distributed derived smart home AI framework. Focus is on the discussed technologies and models (Section 1.1) that deal with the main conceptual challenges, i.e., battery lifetime, bandwidth usage, and processing resources.

Additionally, as described in section 1.1, the challenges include:

- A dedicated model. This model needs to deal with the problems and challenges in the agent-based research areas and the centralized smart home models research areas to make the Smart Home Systems (SHS) models adaptable and compatible with the future of IoT. In addition it needs to facilitate and support a distributed AI framework.
- A SHS contained S.O. concept. This concept must be derived in the light of IoT and in terms of network load, resource consumption, and its ability to support a distributed AI framework. As noted focus is on battery power usage, bandwidth (i.e., channel capacity), and utilization of the available processing power.
- Content and architecture of a distributed S.O. including its challenges such as generic interfaces, structured software architecture, and the embedded hardware devices ability to offer the needed resources.
- An embedded hardware S.O. platform architecture which facilitates the structured software framework. It must offers the needed resources such as interfaces, control logics, event logics, communication logics, support for AI processing, support for small embedded microprocessors, and power resource management.
- An S.O. communication model which is based on known technologies such as ZigBee and 6LoWPAN. It includes transmit-only sensors, clustering, and an overview of future Software Defined Radio (SDR) technologies.
- Simplifications in S.O. and sensor event information and the output data exchanged with systems out of context.

- An AI framework that facilitates the S.O. concept and provides hierarchical layers which offer simple processing units and more advanced temporal processing.
- To overcome the battery power usage, the limited processing power, and the scarce bandwidth resources research is needed in dividing, simplifying, and parallel processing the AI algorithms.

A limitation is that it should be based on well proved and existing systems and technologies. This restriction ensures that it is possible to simulate models and concepts on existing devices and evaluate these against existing comparable research results. However, this limitation does not prevent future extensions and changes as discussed in section 10.2.

The overall research problem can be stated as follows:

Question 1: *How should the distributed system architecture for smart homes be designed in order to incorporate AI and the diversity of S.O's?*

Question 2: *How should the AI be distributed to comply with the smart home system architecture?*

Research question one is essential because it establishes research and provides knowledge about a distributed system architecture which is based on S.O.s (*agents*³). This moves the research frontier in these areas further on. Research question two is essential because it deals with the possibility of distributing smart home AI systems and add real-time learning so it fits with the derived smart home system architecture including its S.O.s. An answer to this question will provide beneficial knowledge which will move the smart home and the IoT research frontiers.

1.3 DELIMITATIONS AND SCOPE

This thesis focuses mainly on smart homes, which cover most of the functionality in the home automation area (Chapter 3) with the added capability to learn from the user behavior and based on this learning, predict services to the user. This matches the standpoint from which

³ An agent is defined as a single AI instance that is able to make a simple prediction based on learning.

the research has been conducted. Thus, home automation is not covered beyond what is needed for exploring and discussing the smart home research. The term *smart home* is widely used for building automation, home automation and in conjunction with IoT. In this thesis the term is limited to homes, i.e., flats, houses and places where people live. Nevertheless, the outcome from this work supports related areas such as AAL, green energy saving technologies and telemedicine. However, to narrow the subject of this work these areas are not discussed unless they support or exemplify the main topics.

In general, smart homes span multiple disciplines why the focus needs to be narrowed down to cover smart homes at a conceptual level, a technical level, and an artificial level. Firstly, the conceptual level is limited to the two basic principles used in smart homes, i.e., the centralized and the distributed one. A combination of these and S.O. technology provides an advantageous compromise. Secondly, at the technical level only available technology is explored in the context of the derived system architecture. This cover standard door switches; wireless sensor network topology; technology and protocols; transmit-only sensors; and routing devices and principles. However, these technologies are used in a new combination that is integrated into the distributed smart home system architecture. In addition, the derived systems must support distributed AI embedded on small controllers with limited resources and other existing devices. It must focus on technologies that deal with the battery, bandwidth and processing challenges. Thirdly, regarding the AI level the research is limited to probabilistic models and only includes other model types when needed to clarify and streamline the research. Nevertheless, these probabilistic algorithms are integrated, modified and used in new ways.

As stated, the primary outcome is limited to a distributed system architecture and an AI framework that supports it. These models offer a high-level relationship, i.e., analysis and design models are provided; however, manufacturers need to design implementation models for supporting their particular products, i.e., fill-in the implementation details such as communication protocol contents, interaction with user interfaces, write proprietary device dependent software, and design their own factory standards for communication, etc. In addition, areas such as user involvement; access rights; usability; market; privacy and trust; and security are excluded. Nonetheless, some of these areas are discussed in the thesis when

needed for clarification purposes of the main topics. In this context it is noted that this work only covers one-user scenarios in the smart home. However, this choice is justified by assuming that each user is identified by a human-recognition-based system such as the position of a smart phone, Radio-frequency identification (RFID) tags and simple visual based sensors.

The validation of the research outcome is based on simulation by using tools such as Matlab, derived Java programs, and microcontroller simulators. But, the research results are based on real life collected datasets available for public download from well recognized universities.

Finally, the derived models, concepts, and solutions are based on currently existing technologies and devices only. I.e., an extrapolation of technological subjects is not performed.

1.4 THESIS OUTLINE

This thesis research, discusses and derives a distributed system architecture for smart homes including its distributed AI framework as stated in the research questions. The dependency and structure of the chapters in this thesis are illustrated in Figure 1.1. As illustrated, two tracks exist. The leftmost track starts with chapter 4 that provides the background for chapters 6 and 7, which deal with centralized and agent-based models. The rightmost track starts with chapter 5 that provides the background for chapter 8, which deals with the enhanced framework that uses AI.

Chapter 1: Sets the scope of the thesis and defines the research questions.

Chapter 2: Explains and discusses the used research methodology.

Chapter 3: Discusses the development from building automation to smart homes. It includes the user perspective, the barriers decreasing the smart home deployment, the use of AI in smart homes, and it outlines the research frontiers.

Chapter 4: Provides a background of distributed systems in relation to smart homes and it provides an overview of selected research areas involved in this field.

Chapter 5: Gives the background of AI with focus on smart homes. It discusses the different types of AI algorithms and presents their application in the context of smart homes.

Chapter 6: Presents an analysis of the state-of-the-art in centralized and agent-based models including its AI framework.

Chapter 7: Covers the derivation of the distributed system architecture including its S.O. model. The S.O.s model is discussed in terms of its architecture with focus on software, middleware and implementation. In addition, a communication model for the S.O.s is derived which supports communication with its outer contexts and integrates the concept of transmit-only-sensor.

Chapter 8: Presents the derived distributed AI framework which is hierarchically based. The performance of the framework elements is researched, designed, simulated and verified against reference material.

Chapter 9: Presents a high-level overview of the final concept and states the contributions from this work.

Chapter 10: Contains the conclusion, outlook, contributions and the future challenges.

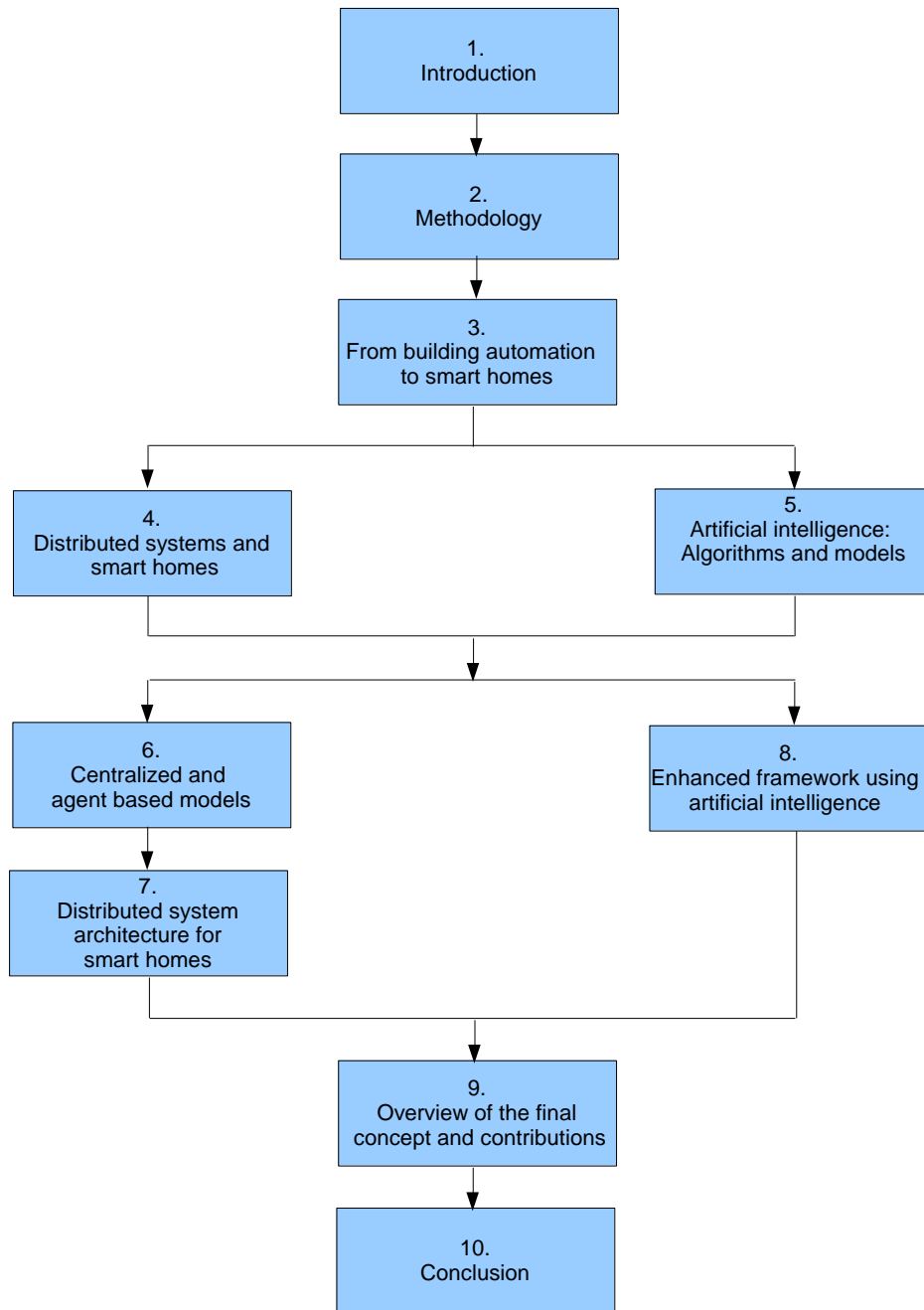


Figure 1.1. Provides a sequence and dependency overview of the chapters in this thesis.

2 RESEARCH METHODOLOGY, APPROACH AND DESIGN

An overview of the rationale behind the chosen methodology is provided in the following. It starts with a presentation of the “hybrid-Imagination” and “truthiness” perspectives (Sections 2.1 and 2.2). These perspectives are important because the up-to-date interpretation of the term “research” deals with the use of modern and future methods; similarly the problematic subjectivity of the term “truthiness” is relevant. Both perspectives integrate well with the inductive and deductive based iterative process model derived for this work.

Following these theoretical and general process considerations the used methodology is discussed and presented. Firstly, the methodological process model is derived and discussed (Subsection 2.4.1). Secondly, the methodology used in the research process is presented and explained (Subsection 2.4.2). Thirdly, the research methodology is viewed from a presentation perspective which follows the structure in this thesis.

2.1 THE “HYBRID-IMAGINATION” PERSPECTIVE

For today’s researcher it is important to understand and make some reflections about the meaning, context and implications of the terms science and research. This makes it possible to perform research in a proper manner. Defining “proper manner” is not a trivial task because it has changed through the influence of history and politics that together with funding and business potential have affected it in form of controlling values. Thus, this section provides an overview and discussion of the important parts filtered by the author’s subjective interpretation.

According to Jamison (2011) producing knowledge is increasingly important, but most of this ongoing knowledge making process cannot be referred to as science and the so-called “scientific community” has simply become a figment of imagination. This point of view is supported in Normann et al. (2011) by a reference to the work of Latour (1998):

Science is certainty; Research is uncertainty. Science is supposed to be cold, straight and detached; Research is warm, involving and risky. Science puts an end to the

vagaries of human disputes; Research fuels controversies by more controversies. Science produces objectivity by escaping as much as possible from the shackles of ideology, passions and emotions; Research feeds on all of those as so many handles to render familiar new objects of enquiry (Latour, 1998).

This deals with the paradigm shift we are seeing from “science” as a certainty provider to a more uncertain concept founded in “research”. This approach supports a more experimental methodology that fits well in the modern world. Thus, this concept offers the degree of freedom needed to deviate from the established process models and designing one that fits to this specific work.

This work uses small dynamic cycles that complete a specific research topic, i.e., the results grow out of a circular concept as discussed later. Sorting these topics and looking into the most difficult and critical ones (critical items) in the beginning of the research is advantageous. The rationale behind this is that the cost and risk of changes are much lower in the beginning of a project; actually, this cost relationship is exponential (Ambler, 2006). Thus, in this work the most difficult and critical parts are planned early in the research process. However, a risk is that the prioritization of the parts is based on an incomplete knowledge because it is decided early in the project phase. Adding an iterative element means that critical parts are revisited, but redoing them is resource consuming anyway.

The principles of “hybrid imagination” have been used as a guide for this work. As stated in (Normann & Jamison, 2011) the “hybrid imagination” approach uses the external changing conditions as an empowerment that supports the understanding of an experimenting approach. This means that the process of exploring, researching and experimenting should adapt to the changing external conditions in the form of adjusting processes and methods used, planning, and decisions. In this work these adjustments have been done based on the present knowledge level, i.e., as the research progresses more knowledge is provided, and this new knowledge then guides the adjustments for the future processes and methods.

It has been found that the solidly based concepts of “research versus science” and the “hybrid imagination” catalyze the used methodology.

2.2 THE SCIENTIFIC WORK AND “TRUTHINESS” PERSPECTIVE

Fundamentally, a methodology is based on a systematic collection of guidelines for solving a problem. Because this project is primarily based on research, the outcomes must be anchored in commonly accepted scientific research methods, i.e., it needs to be solidly founded in a scientific research methodology. As a consequence, the scientific outcome must be considered to be the “truth” about the subject in question. A process for approaching the “truth” is going through a sequence of research activities that supply statements or claims that converge into the “truth”. As will be seen in this section this is a non-trivial task.

Before going into the discussion of methodology it is important to define what scientific work actually is. Brodersen (2008) defines it this way:

The objective is a precise control of one's thinking, which is missing its goal if it is not systematic (Brodersen, 2008).

Also presented is a list dealing with all the necessary phases:

1. Description of the problem (unsatisfactory versus satisfactory state)
2. Theory about getting from unsatisfactory state to satisfactory state
3. Experiments that can falsify/verify each hypothesis
4. Expected outcome of the experiments
5. Observed outcome of the experiments
6. Conclusions to be drawn from the experiment outcome

A discussion of some of these points in the line of this work is performed in the following.

Combining the discussed approach with a cyclic process model provides some benefits. By embedding the six points list into each cycle in an iterative process it become self-sustaining, i.e., each round provides outcomes that are useable for planning the processes and methods used in the coming rounds. However, the iterative element also means that critical parts are revisited which provide the possibility to add further corrections.

Choosing the methodology based on a process of self-sustaining iterative steps is in good agreement with the basic idea of “hybrid imagination”. This is the case as long as the

timeframe used for gathering sufficient scientific knowledge and obtaining technical skills in each cycle is short compared to the contextual timeframe.

Normally, choosing a process model solves some problems, but also leaves the problems and questions about “truthiness” of the produced knowledge unsolved. In general knowledge and statements about reality based on solid theories are accepted if the supporting arguments are so convincing that people agree they are true. Thus, science does not provide the only objective and absolute truth (Brodersen, 2008):

Science supplies claims (statements) about reality which approach truth following long-term, systematic work. We will never achieve absolute truth – we can only get close (Brodersen, 2008).

From a more philosophical point of view Brodersen (2008) states that:

- Truth is what each of us perceives to be true based on our own experience; truth is private.
- A claim is true if it corresponds to external reality.
- The purpose of language is to provide descriptive, precise definitions and an objective presentation of reality.
- A statement is true if it is coherent with and can be united with a broader system of statements without giving rise to contradictions; theory construction and logically coherent argumentation.
- Things that everyone agrees about are true (consensus); the concept of truth is anchored in a social and historical context, which is subject to constant change

Relating the term “truth” to this research project is difficult because the expected outcome consists of this thesis describing the developed concepts along with simulated results. In addition it provides mathematical and some coded algorithms for future research projects. A proper way to deal with the “truth” in this context is that the “truth” is provided by comparing the developed results with a reference, i.e., reviewed published results from other independent researchers. Thus, a review process will ensure that the reference contents are coherent and

can be united with a broader system of statements without giving rise to contradictions and it will also provide proof that consensus is found to some extent.

2.3 GENERALIZING THEORIES BY USING INDUCTION AND DEDUCTION

The principles of *induction* and *deduction* help to generalize a theory by removing the special attributes, i.e., transforming a subjective theory into a more objective one. Most people find this approach increasing the reliability and truth level.

Using the principle of induction means drawing conclusions based on specific examples and hypotheses, i.e., a more general abstraction level is sought. However, induction can lead to a wrong conclusion. This is exemplified by the commonly known fact that thousands of lemmings jump off a cliff and thereby kill themselves every year (Brodersen, 2008). A way to deal with this risk is combining it with common sense. Common sense helps us choose and find the right path in the possible choices.

The induction principle is used in the methodology of this work where it provides a theory based on data from simulated smart home components mathematical derivations, library researches and Internet explorations.

Regarding deduction it involves drawing a conclusion about individual cases based on general rules. This is kind of a transferring method because it is possible to progress from a general level of abstraction to a more specialized one. Reflecting on this in connection with the cyclic development process used in this work, it provides the background for succeeding rounds. Thus, when a round is finished a new layer of features is added to the unfinished product, but this often leads to new ideas and improvements of existing product parts –actually this is what “prototyping” is about.

The figure below provides a graphical overview of the just discussed theoretical items (Brodersen, 2008):

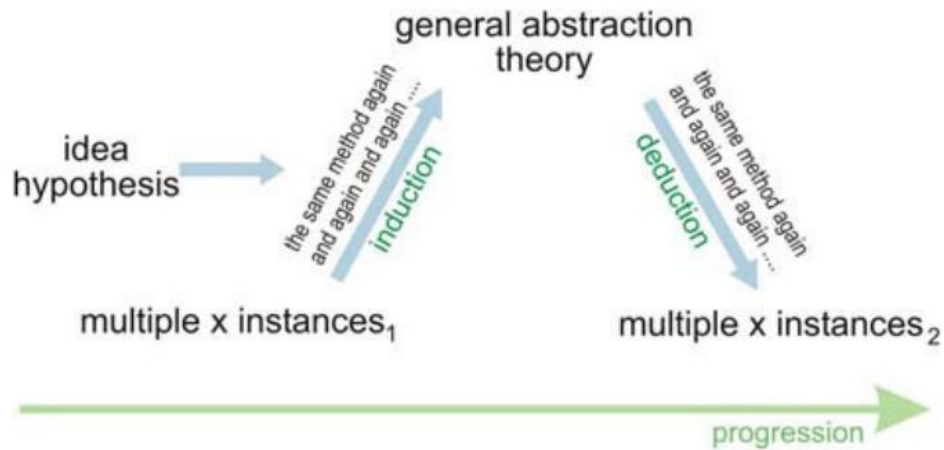


Figure 2.1. Using induction and deduction to reach a logical conclusion (Brodersen, 2008).

As illustrated in Figure 2.1 this process involves a hypothesis, which in this work is substituted with a draft attempted and simulated solution to the research questions. By combining this in a synthetic way with induction, i.e., generalization based on collected information, a more general abstraction level can be reached. Filtering this level creates a base for using deductive methods to infer new knowledge into the project by generating examples and doing experiments with the simulated results.

2.4 SELECTED METHODOLOGICAL APPROACH

A model for the selected methodological approach is presented. Thus, based on the discussed methodical elements in the previous sections focus in this section is set to combine these into the used methodology.

2.4.1 THE METHODOLOGICAL PROCESS MODEL

A part of the methodology governing this work is based on existing datasets achieved from state-of-the-art research in the smart home area. Thus, Cook (2012) and Kasteren et al. (2011) have produced several datasets which are based on real-world experiments where people lived in smart homes for some period of time (Section 4.3). These datasets are usable in this work because they contain raw sensor data arriving from sensors triggered by the residents performing their daily tasks. All datasets are annotated by the residents on the fly, by using a Bluetooth headset. The other source of data was papers and online sources which deal with

smart home experiments and architectures, but also related sources such as IoT, distributed technologies, and pattern recognition have been included.

All datasets points to the use of qualitative methods because the available data are not quantitative measurable, they need to be explained, and they need new theories and knowledge to be researched. Qualitative research methods are emergent, interpretative and require complex reasoning that is multifaceted, iterative, and simultaneous; additionally they include inductive and deductive processes. The qualitative method research methodology offers iteratively based experimental research strategies. Thus, this work selects a combined inductive/deductive cyclic approach that analyzes the datasets in an iterative manner and thereby derives algorithms, models and simulations (see the following paragraph). Finally, dataset based simulations are used to explore the derived results, which are then validated and verified by comparing it with other related research works. Because the validation datasets are annotated by the smart home resident they can be characterized as quantitative data, which is considered accurate and reliable through validity and reliability.

The cyclic process used in this work divides a task into a collection of small packets. This is beneficial because it lowers the project risk, it enables early presentation, it offers isolated review of one research topic at a time, and it revisits packets when more knowledge is available (Cockburn, 2008). A downside is that requirements are not steady because they also evolve with the produced packages. Nonetheless, in research context processes are guided iteratively from the research questions, so these deviations are limited and manageable.

A risk prioritization approach was used to control the used cyclic process. Thus, by selecting high risk items and focusing on these first in each research cycle the risks of being trapped late in the process by unsolvable items were reduced. Additionally, this cyclic process was combined with a sequence of induction and deduction steps (Section 2.3) as explained later and illustrated in Figure 2.2.

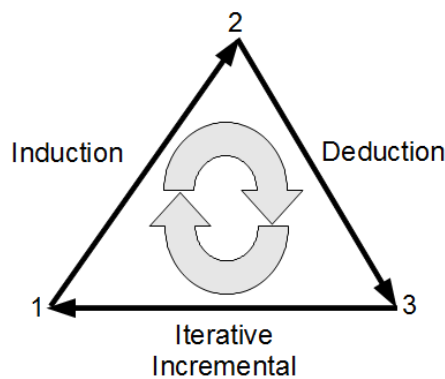


Figure 2.2. Methodological process model used in this work.

The process starts at point 1 in Figure 2.2 by collecting specific system knowledge. Using the induction step brings the model into point 2 where the established abstract knowledge is generalized into a theory. Using this theory to “clean” the results derived from simulations and mathematical derivations produces outputs that have more certainty and contents with a higher level of truth. This generalized knowledge is then transferred through the deduction step (point 3) which produces and generates examples that are useable for simulations. These examples are then combined with theoretical knowledge to refine and filter them into the focus area where they are compared with reference models. By comparing the achieved outcome with similar works, performed by others, an updated general theory is produced by using the induction step, i.e., moving from 3 through point 1 to point 2, etc. Thus, moving around this circular model produces general outcomes such as a distributed conceptual smart home model and it produces more specific outcomes such as the detailed AI framework that has been derived in this work.

The time-structure of the used research methodology is presented in subsection 2.4.2. It follows the strategy discussed earlier in this section. However, this research methodology did not provide the best way to present, illustrate and explain the results and outcomes from this research work. Thus, a presentation methodology view has been derived which organize the topics in a logically structured way that is more readable (Subsection 2.4.3). Nonetheless, it is noted that the presentation methodology is an alternative view of the research methodology which covers the performed research work.

The presentation methodology structure is discussed in the following and illustrated in Figure 2.3. As explained in subsection 2.4.2 the two models have been derived in a separate manner. Thus, the distributed system architecture and the AI framework have been derived individually based on the background chapters 4 and 5, i.e., they do not depend on each other. The reason for decoupling these two main subjects from each other is to provide a loose coupling. A loose coupling means that each of them can be changed or substituted in the future without affecting the other one. It also means that they can be treated individually throughout their lifecycle, i.e., development, test, updates, maintenance, and end-of-life. From a presentation point of view (Subsection 2.4.3) the background research has been presented in chapters 4 and 5 as illustrated in Figure 2.3. These chapters provide an overview of the used and related state-of-the-art research areas. The main contribution of this work is allocated to chapters 6, 7, and 8. Where chapters 6 and 7 present and discuss the derived distributed system architecture, chapter 8 presents the derived distributed AI framework.

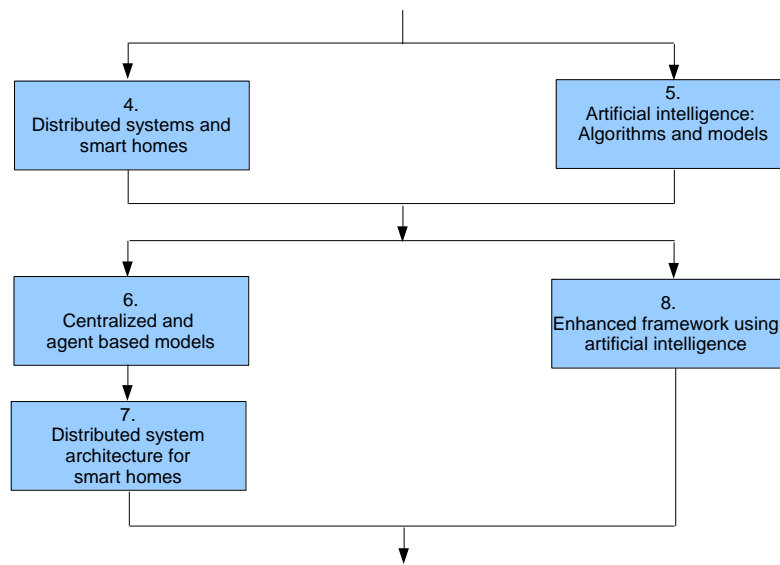


Figure 2.3. Presentation methodology for this work. Please note, the full figure is available in Figure 1.1.

2.4.2 RESEARCH METHODOLOGY

A more detailed overview of the research performed to deal with the research questions (Section 1.2) is provided and discussed. Thus, by using the methodology discussed in section 2.1 to section 2.3 some phases have been derived and used in this work:

Phase one: The critical element was the holistic distributed system architecture including its AI. The term “holistic” underlines that this model contains hierarchically based smart devices which connect to high-level subjects such as user-interface and more complex systems. In this process many small cycles were used to verify and update the model. Thus, a bottom-up approach was used to deal with the distributed smart devices and a top-down approach was used to interface the high-level elements to these low-level elements.

Phase two: The low-level model part was iterated by using a bottom-up approach. Focus was set on the low-level critical elements such as power savings, network load savings, and embedded processor resource savings. Using this focus the distributed system architecture was iterated to include an AI framework and combine this with the smart devices (S.O.s). Hence, the model was re-designed, the simulation framework was modified, and the simulated results were verified – in a repeating bottom-up process.

Phase three: The critical item in focus was the low-level communication between the smart devices. By adding knowledge from related areas such as Internet of Things and wireless sensor networks into an iterative process the embedded S.O. concept was derived.

Phase four: The critical item was integrating advanced processing, data storage and a user-interface. The research process was based on iterating through different phases such as update model, modify simulation model, simulate, verify in a top-down manner.

Phase five: By using the distributed system architecture as input together with the mathematical description of the probabilistic AI models potential mapping structures were investigated. This mapping process was considered a critical item because it defined the overall layout of the AI framework. Further iterations over the mapping structure in combination with derived mathematical probabilistic AI models converged into two implementation equations.

Phase six: The implementation equations and the distributed system architecture were modeled, through series of iterations, by two object process models (Dori, 2002). These models provide the basis for implementing and integrating the AI framework into the derived

distributed system architectures. Finally, some simulations with derived tools were performed to validate and compare the simulated results with other researchers work.

2.4.3 PRESENTATION METHODOLOGY

The presentation methodology contains the information presented in the research methodology viewed from a different perspective, as discussed earlier in section 2.4. It follows the thesis structure, i.e., chapters 6, 7 and 8 are presented in the following three tables. Please note that the term *phase* in these tables is different from the *phases* used in the previous section.

Phase	Method	Tools
Initial theory (inductive)	Literature review and investigation of the research question one.	Abstract model construction and theory development (library and online searches).
Initial model (deductive)	Development of heuristic model of SHS types, based on initial theory	Designing and developing two generalisable models.
Model mapping (inductive)	Analyze if researched SHS structures (qualitative data) fit into the models.	Library and internet SHS model search. The two developed models.
Evaluation	Investigate the degree of model matches achieved (qualitative data).	The two developed models. Mapping results from the models.
Modify models (deductive)	Modifying heuristic SHS models, based on the evaluation.	Re-designing and developing two generalisable models.
Derive models pros and cons (deductive)	Analyze the models in a distributed and a centralized SHS problem context.	Library and internet SHS model search. The two developed models.
Deriving SHS holistic model (inductive)	Interpretation of derived models pros and cons. Development of final holistic model.	Model construction and theory development (library and online searches).

Table 2.1. The presentation methodology flow covering the used process in chapter 6.

The development process used in chapter 6 follows the discussed cyclic development approach discussed earlier. As noted (Table 2.1) it bounces between inductive and deductive steps (Section 2.3) in an iterative fashion to converge into a final holistic model.

Phase	Method	Tools
Initial SHS (inductive)	Literature review and development of SHS model structure and architecture based on qualitative data.	Theory development (library and online searches). SHS holistic model.
SHS model architecture (deductive)	Development of SHS model architecture.	Final holistic model, research question, and initial theory.
SHS model investigation (inductive)	Literature review, SHS architectural analysis, research of SHS model structure and elements.	SHS model architecture. Theory development (library and online searches).
SHS model design (deductive)	Design of SHS model elements.	SHS model architecture. Theory development (library and online searches).
Initial S.O. (inductive)	Literature review of agent-based systems and IoT. In addition, development of S.O. model structure and architecture based on qualitative data.	Theory development (library and online searches). SHS model elements.
S.O. model architecture (deductive)	Development of S.O. model architecture.	Research question and initial S.O. model structure. Smart home architecture from Cook (2012) and Kasteren et al. (2008a).
S.O. model implementation (inductive)	Design and development of S.O. implementation model.	S.O. model architecture and S.O. model structure.
S.O. model evaluation (inductive)	S.O. model evaluation based on an empirical scenario.	S.O. implementation model. Empirically derived scenario.
Final SHS model reflection (inductive)	Analysis of model using deductively derived research sub-questions and the questions derived for the holistic SHS model.	Final SHS model. Research sub-questions. SHS holistic model pros and cons.

Table 2.2. The presentation methodology flow covering the used process in chapter 7.

The flow in chapter 7 is presented in Table 2.2. It follows the cyclic development process, discussed earlier, and iterate to the final SHS model by using inductive and deductive steps (Section 2.3). Some notes to the table are:

- The phase *S.O. model architecture* uses smart homes architectures from different researchers, and these are detailed in chapter 7 and later in this section (note 1).
- The phase *S.O. model evaluation* uses an empirically derived scenario. This scenario has been derived from the system architecture problems revealed and discussed in chapter 6. Its primary purpose is to provide a platform for comparing the existing models with the derived SHS model. Thus, it has been designed from this perspective.
- The derived sub-questions used in phase *Final SHS model reflection* are described in the chapter 6.

Similarly, the flow in chapter 8 is presented in Table 2.3. It iterates to the final SHS model by using inductive and deductive steps (Section 2.3) and it follows the cyclic development process, discussed earlier. Additionally, some of the items have the following notes:

- Phase *Verification of AI framework* uses two dataset from WSU CASAS as detailed in subsections 8.4.2, 8.4.3 and later in this section (note 1).
- Phase *Interpretation of AI framework analysis* is detailed in subsections 8.4.2, 8.4.3, 8.6.2, 8.6.3 and later in this section (note 2).
- Phase *Verification of AI framework on S.O. model*, together with its used dataset, is detailed in section 8.8 and later in this section (note 1).

Note 1: These datasets are sampled and annotated from experiments performed by Cook et al and Kasteren et al. who set the research frontiers in the smart home research area today (2013). Moreover, as stated in section 1.3 this work is restricted to derived models, concepts, and solutions that are based on currently existing technologies and devices only. I.e., an extrapolation of technological subjects is not performed.

Note 2: The choice of candidates for the comparison with others works are the ASBR and CBR systems presented by Chen et al (2009), which are also considered state-of-the-art in the smart homes research area.

Additionally, two papers have been derived from this work, which have been published in an international journal (Scientific Journal of the University of Szczecin); however, they have not been included in this thesis in order to keep it a pure monograph.

Phase	Method	Tools
Initial AI framework (inductive)	Literature review and analysis of AI framework model structures and architecture based on qualitative data.	Theory development (library and online searches). Final SHS model. Research-question two.
Derive AI framework (deductive)	Derive AI framework architecture using mathematical modeling, cross analysis with the final SHS model.	AI framework architecture. Mathematical tools (MATLAB, etc.)
Investigation (inductive)	Mathematical interpretation of derived equations.	Derive AI framework architecture. Mathematical tools (MATLAB, etc.)
Derive AI framework components (deductive)	Modeling derived equations into implementation object process models.	Derived equations investigation results. Modeling tools.
AI framework components (deductive)	Design of implementation models based on mathematical analysis and object process models.	Derived equations investigation results. Implementation object process models. Mathematical tools (MATLAB, etc.)
Simulation framework (inductive)	Analysis of Simulation tools and models.	Theory development (library and online searches). Object process models.
Simulation framework (deductive)	Design of Simulation tools and models. Design and implementation of simulation programs.	Java and MATLAB tools.
Verification of AI framework (deductive)	Analysis of AI framework using simulation programs and empiric qualitative datasets.	Empiric dataset from WSU CASAS (Cook, 2012).
Interpretation of AI framework analysis (inductive)	Interpretation of verification results by comparing with other researchers results.	Output from the verification of AI framework phase. Other research results (library and online searches).
Verification of AI framework on S.O. model (inductive)	Simulation of implemented AI framework on the S.O. model. The simulation uses qualitative empiric datasets.	Programming and simulation tools. Qualitative empiric datasets from Cook (2012).

Table 2.3. The presentation methodology flow covering the used process in chapter 8.

2.5 SUMMARY

The first part in this chapter deals with the modern term “hybrid imagination”. It was concluded that the solidly based concept of hybrid imagination supports the used circular based methods, it aligns the research methodology with the future, and it provides a guide for using the scientific methodology in this work. For the challenge of providing a “true” research (i.e., “truthiness”) the important rules and guidelines provided by Brodersen (2008) were incorporated into the methodology of this research work. Another important item was the discussion of the used iterative process model in the light of a deductive and inductive step concept. Finally, the selected methodology was presented and discussed. This methodology combines all previously discussed elements into the used strategy, which is presented by using a “presentation view” making the employed research methodology more readable (Section 2.4).

3 FROM BUILDING AUTOMATION TO SMART HOMES

This chapter is divided into a presentation of the historical development of smart homes and a detailed explanation of its predecessors the building automation area (Section 3.1) and home automation (Section 3.2) area. The smart homes are presented in section 3.3. Its relation to the IoT research area is discussed in section 3.4 and other related research areas and activities are discussed in section 3.5.

The associations and relations between the intelligent buildings, the home automation and the smart home areas need some elaboration. Thus, these areas and their relationship are the focus of this chapter.

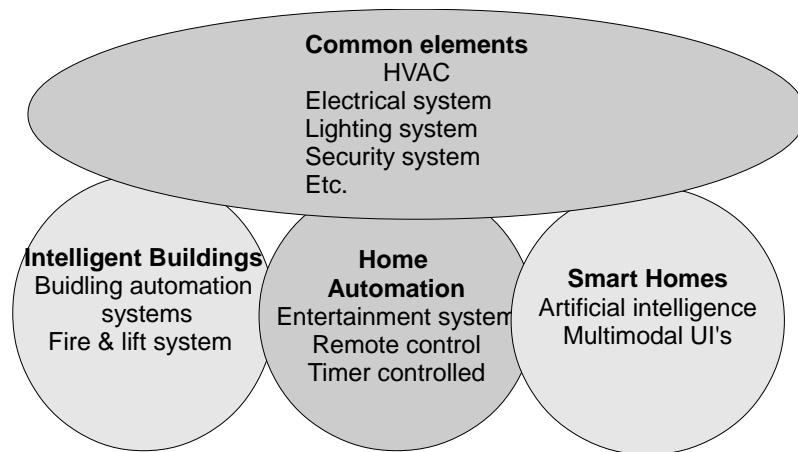


Figure 3.1. Relations between Intelligent Buildings, Home Automation and Smart Homes.

As illustrated in Figure 3.1 *Intelligent Buildings*, *Home Automation* and *Smart Homes* are not simply synonymous of each other even though they have *common elements*. *Intelligent Buildings* have a Building Automation Systems (BAS) that handles its resources in a planned fashion, thus it needs some fixed in-advance programming to be able to behave intelligently and it is supervised by some administrative staff. *Building automation* is discussed in section 3.1. *Home Automation* systems contain many resources and technologies that are similar to the ones offered by the BAS system. However, they are dedicated to offer systems that support functionality such as entertainment, timer and remote control, i.e., they focus on “normal homes” where people live, i.e., houses, detached-houses and flats, etc. From a

pragmatic point of view *Home Automation* is a residential extension of the *Intelligent Buildings* concept. *Home Automation* is discussed in section 3.2. *Smart homes* extend the *Home automation* area by adding intelligent behavior as discussed in section 3.3. Finally section 3.5 offers overview of the related research areas and activities.

3.1 BUILDING AUTOMATION

A higher level understanding of the combined buildings and AI research areas is difficult to obtain. People who talk and write about it are often using different terms and from time to time they mix it up in an incorrect way, thus a lot of confusion exists. This is probably a consequence of the missing standardization and lack of a common technical language that normally is provided and developed over time by professionals. So, the field of building automation in combination with AI is not different in that perspective. Even though this research field has not converged into few mainstreams that are well established in the academic world, it has been around for the last two decades. During this period many different standards have been suggested and numerous intelligent building theories and technologies have been published and developed.

One of the reasons why this field has not converged into a common standardization is that the definition of intelligent buildings has changed over the years. Thus, the concept of intelligent buildings started in the early 1980's is mainly driven by the development of relevant technologies and changing environmental needs. In the period from 1980 to 1985 intelligent buildings are controlled to do some well-defined functions. After this period until approximately 1991 this definition was redefined to deal with buildings that are capable of responding to changing needs. Finally, from approximately 1992 to present this definition shifts to deal with building features that effectively satisfy the changing needs (Tanasiev & Badea, 2012). Historically, the focus has changed to a more operative form, i.e., from *respond* to *satisfy* based on changing needs.

Despite the huge amount of work, no general worldwide single definition of the term *building automation* exists today. However, the approaches to define it can be divided into three categories (Wang, 2010). First, the term “performance based definition” is used to express

different kind of performances in a building. Second group is a “services based definition” and the third is a “system based definition”. These are stated in the following.

- A performance based definition:
 - European Intelligent Building Group: A building created to give its users the most efficient environment; at the same time, the building utilizes and manages resources efficiently and minimizes the life costs of hardware and facilities.
 - Intelligent Building Institute (IBI): An intelligent building provides a highly efficient, comfortable and convenient environment by satisfying four fundamental demands: Structure, system, service and management, and optimizing their interrelationship.
- A service based definition:
 - Japanese Intelligent Building Institute (JIBI): An intelligent building is a building with the service functions of communication, office automation and building automation, and is convenient for intelligent activities.
- A system based definition:
 - Chinese IB Design Standard (GB/T50314–2000): Intelligent building provide building automation, office automation and communication network systems, and an optimal composition integrates the structure, system, service and management, providing the building with high efficiency, comfort, convenience and safety to user.

These three categories express the definition of intelligent buildings from different viewpoints, i.e., performance, service, and systems viewpoints (Wang, 2010). But, simplified it can be boiled down to the following assumption:

Intelligent buildings provide the user with efficient and well integrated services, comfort and security by providing effectively managed resources that adapt in an intelligent way.

Based on the interpretation of the intelligent building definition a question raises about what an intelligent building provides? To answer this it is beneficial to divide “intelligent buildings” into some groups. Thus, from the “smart home” scope of view intelligent buildings

can be divided into one group that contains “smart homes” and another group that deals with the remaining buildings, i.e., factories, hospitals, schools etc. The last group is not well defined and consists of a residual mix, i.e., different kind of buildings beyond group one and group two.

Intelligent buildings can be considered as a collection of concatenated structures, systems and technologies that have growth over time (Johnson, 2012). This statement is supported by Wang (2010) who states that a building service provides an umbrella of technologies. These use a wide range of computerized building control systems such as BAS or Building Management Systems (BMS). The BAS comprises several subsystems connected in various ways to form a complete intelligent building system. Often these systems are designed and engineered to fit the individual requirements for the service that are needed in a particular building. However, some general service characteristics can be found within the variety of different intelligent buildings. Common services for most of these are discussed by Granzer et al. (2010) and they are illustrated in Figure 3.2.

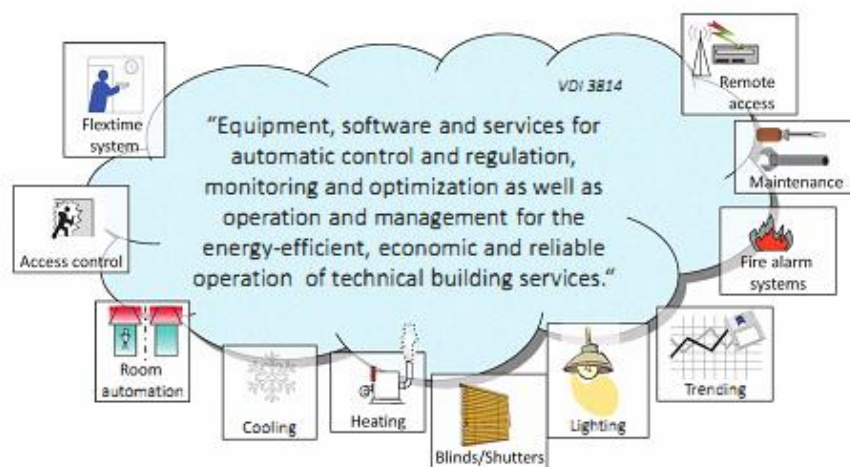


Figure 3.2. Common services offered by most building management systems (Granzer et al., 2010).

Figure 3.2 shows the services offered by a state-of-the-art BAS which includes functionality that keeps the building climate within a specific range by regulating the Heating, Ventilation and Air Conditioning (HVAC) systems. In general, it also takes care of other structures such as electrical, lighting, security, fire and lift systems. In hospitals and industrial buildings BAS also manages steam, compressed air and water systems. Moreover, BAS is often used to

monitor, control and manage these services and thereby able to provide corrections to the systems; they emit notifications (e.g., emails) to building staff in case of device malfunction or failure.

Looking into the benefits of using a BAS they comprise: Save time / cost, provide comfort and sustainability. In addition, it reduces the operation cost and increases the staff efficiency because it is possible to control multiple systems from one centralized computer system. In addition, it provides cost savings because only one interface is needed as an alternative to multiple front-end interfaces. Moreover, it optimizes energy efficiency because it provides interoperability between energy storing, load shaping and load shifting technologies. This interoperability enables more precise energy management often named micro-management (Johnson, 2012). Regarding troubleshooting and maintenance BAS provides many real time data from its monitoring behavior which means that more accurate forecasts of failure and wear down are provided. This reduces the amount of high cost unscheduled repairs. In some building systems BAS also takes care of security and safety, which makes the often complex process of building and remote access manageable and easy to use. Thus, it encapsulates the complexity and provides simple operations for users and administrators.

Because many BAS systems are designed in an open and flexible way integration with other access systems or human resource database systems can easily be done. However, BAS also has downsides in form of increased building wiring and network systems complexity and it has relatively high initial installation costs. Another disadvantage is that the building normally last longer than the BAS system, meaning that further investment is needed on the fly to handle upgrades and maintenance.

From a sustainability point of view reducing the environmental pollution is one of the top priority items of the European Union that adopted the 20-20-20 “renewable energy directive” with the target of reducing the greenhouse gas emissions, cut the energy consumption and increase the renewable energy use – all by 20% before 2020 (Tanasiev & Badea, 2012). Tanasiev et al. (2012) highlights that the highest energy consumption in Romania is consumer heating and following that, is water heating. This consumption can be reduced by approximately 30% by using a BAS system according to Tanasiev et al. (2012).

Looking into the future for BAS systems reveals that new BAS applications have emerged. They are optimized to fulfill the owners demand for a more fine-grained information level about building performance so operating costs can be reduced, corporate sustainability goals can be met and the occupants can be safe, comfortable and productive. These new systems are based on open platforms and they use standardized communication standards like BACnet, LonWorks, Modbus, PROFIBUS and EIB (Wang, 2010). They also provide a more intelligent behavior, because they can do much more than a simply connect and exchange information over a communication network. These modern devices are now able to communicate in a way that truly traps into their embedded intelligence and they are able to work together in an ad hoc manner, predict trouble and take preventive actions to avoid an upcoming problem without human assistance. In the essence, these new BAS systems help owners attain their facility related goals by optimizing the capability of all equipment and systems across their entire enterprise (JohnsonControl, 2012). In general, the tendency is that these systems add AI to obtain a higher degree of behavior that approaches the skills of a human operator and they communicate together to optimize the overall system performance.

3.2 HOME AUTOMATION

The purpose of *Home Automation* is to make life easier for its residents by controlling mundane functions such as light, heat, ventilation, entertainment systems, appliances etc. to improve comfort, convenience, energy efficiency, and security. But it requires the parameters to be pre-programmed by the users, i.e., it cannot be done automatically.

The *Home Automation* concept is not autonomous in any way. It is simply remote controlled by the users to do timer based or pre-programmed functions that can be either simple or more complex. Often this requires trained personnel to interact with it as illustrated in Figure 3.3.

Modern *Home Automation* homes of today (2013) are wired by power lines, TV outlets and they are equipped with Internet that is delivered wireless or by wire. So, these connections make it possible to remote control domestic activities and devices such as houseplants, entertainment systems, pet feeding, yard watering, and control different kinds of domestic robots like vacuum cleaners. Thus, it is possible to remote control these from either a near or a more distant place using a personal computer or a modern smart phone (Granzer et al., 2010).

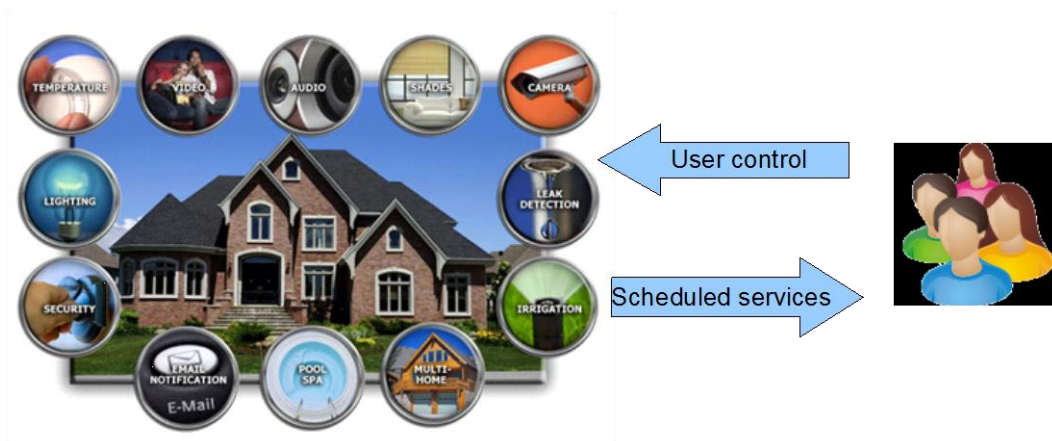


Figure 3.3. Services offered by Home Automation are scheduled ahead and user controlled.

There are many research projects exist in the home automation field. Some researchers use the Internet browser in the smart phones to connect to the automated home systems. An example is a Bluetooth based home automation gateway (Mehairi et al., 2007). Mingyi et al. (2010) introduce a home automation management system that combines devices and the Internet by using a microcontroller. The same approach has been used by Xiaojuan et al. (2010) who present a home automation gateway that controls home automation devices and take care of illegal access or intrusion from the outside world. Other related projects are the “Keep in Touch” project that deals with smart devices in combination with near field / radio frequency technologies to facilitate tele-monitoring processes for elderly people (Dohr et al., 2010). In addition, remote control of automated homes by using smart phones has been presented (Mao et al., 2010). Similarly, a slightly different approach using mobile RFID tags has also been presented (Darianian & Michael, 2008).

Looking into the state-of-the-art of available systems (2013) different choices exist. One of the most popular ones is the *mControl* (Top-ten-reviews, 2013). It provides a real-time automation and control engine; it supports Windows; it has remote access by both mobile and personal computer; and it has multiple user interfaces. In addition, it supports most standards and offers energy management, security and surveillance, media management and data logging. Other similar systems are *HomeSeer*, *Control4*, *PowerHome*, *Vivint* and *ActiveHome Pro*. These Home Automation systems offer different variants of the functionality offered by *mControl*. The company *HomeSeer* base their home automation devices and user interface on

the popular Android operating system from Google named Android@Home (Helmke, 2013). An illustration of *HomeSeer*'s Android based user interface is given in Figure 3.4.

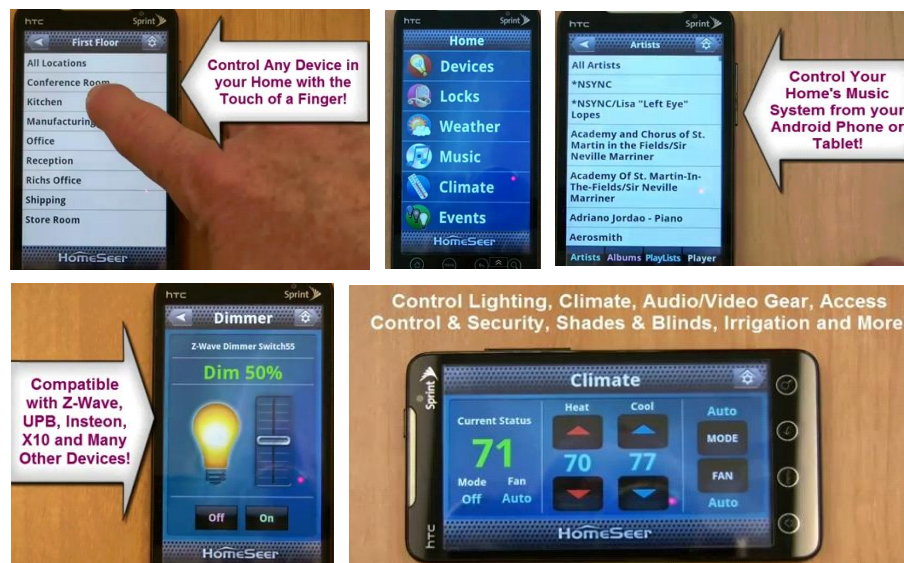


Figure 3.4. HomeSeer Android@Home Smart Phone user interface.

Google announced at their I/O 2011 conference that they are looking into the Home Automation market with a concept named Android@Home (Google-I/O, 2011).

Focus on Home Automation technologies has also been a subject for the European Union, especially in their FP7 programs. Looking into their work program for 2013 in the ICT area many of the main subjects are related to the Home Automation field. Thus, areas as service infrastructures, cognitive systems and robotics, ICT for health, ageing well, low carbon emission and future emerging technologies are listed (EU_FP7, 2013). One particular co-funded program named *HYDRA* provided a middleware that enables home automation developers to incorporate heterogeneous physical devices into their applications together with an easy to use web interface. So, the hydra framework enables secure and trustworthy devices and services through distributed security and trust middleware components.

3.3 SMART HOMES OVERVIEW

Shifting the focus to Smart Homes (sometimes referred to as a smart houses or eHomes) they contain common elements that are shared with the intelligent buildings. Some of these elements are HVAC, lightning, electrical systems as illustrated in Figure 3.1. They are

considered an extension of the home automation field because they add more advanced control features and AI.

It is predicted that our future life will be influenced by the smart homes because they will provide services according to our preferences and they will act “intelligently”. The concept of smart homes has become increasingly popular throughout the last couple of years. I.e., new buildings are planned and built according to new sustainability guidelines. Domestic homes are becoming more energy efficient throughout the last periods, especially for two reasons. The home owner would like to reduce operational costs over the home lifetime and contribute to the environmental saving.

Reinisch et al. (2010) reviewed smart homes under a digital ecosystem perspective. Their main goals were to minimize energy consumption and at the same time guarantee user comfort by applying AI to the smart homes. A similar rationale has been used by the European Union in the broader scope of Internet of Things (IoT).

Thus, they consider IoT as an umbrella for a new paradigm where smart homes are one of the carrying elements (EU_Commission, 2009). Furthermore, the European commission is funding the Ambient Assisted Living Joint Program that works with more than 100 projects that have resulted from the first four calls (AAL_Association, 2012). Its aim is to create better conditions for elderly and strengthen industrial opportunities in Europe by the use of ICT.

The EU Commission also looks into the area of green technology where they have set a roadmap for moving to a low-carbon economy in 2050. One important element is low-emission buildings with intelligent heating and cooling systems (European_Commission, 2013).

3.3.1 USERS PERSPECTIVE

One of the main barriers for adopting and disseminating smart home technologies is to design a user-friendly AI system that controls the smart home and its objects (Brush et al., 2011). This may be realized in the form of an intelligent manager, which encapsulates the complexity and presents the user for a simple interface. Thus, one of the most important features is the ability to learn and predict user activities in an autonomous way to ease manageability. One

approach for managing smart homes is by using an intuitive Graphical User Interface (GUI) based 3D virtual reality world (IBM Sales & Distribution, 2010) similar to the one used in “Second Life”. The disadvantage is that it requires a huge amount of processing power and large amount of data to be exchanged or stored.

A more useful concept is to use today’s smart phone camera to detect and identify the S.O.s (Lynggaard, 2013d). Such a concept could be based on high-level software objects handled by cloud-based services. It offers the benefits of high usability, limiting the amount of stored and exchanged data, and it moves the S.O. configuration to the service providers and device manufacturers.

Alternative ways for users to interact could be to use finger gesture in front of the camera or still pictures for remote detection, e.g., like the content-based image retrieval concept used by Google. These approaches are, however, quite complicated with low usability and poor manageability (Silva et al., 2012).

3.3.2 DISCUSSION OF SMART HOME BARRIERS

Different barriers exist that reduces the smart home technology acceptance ratio. Brush et al. (2011) point out that especially four barriers should be considered. They conducted 14 semi-structured interviews and household tours for exploring why the household had installed home automation; the reason for installing home automation; their long term experience of living with it; and how they handled guests and security considerations. The four barriers they found are high cost of ownership, inflexibility, poor manageability and difficulty achieving security.

Looking more specifically into the high cost of ownership it should be seen in the light of the market potential for smart home / Home automation devices. So, the Broadband Forum (Subsection 3.5.3) outlined a value proposition MR-239 (Fedosseev et al., 2011). It brings the next generation of value added service like AI controlled devices to connected homes. In the area of home monitoring and home control the North American market is estimated to reach 2.4 billion dollars in 2012. The U.S market for home healthcare technology will reach 5.7 billion dollars by 2015 and effect 26 million users. Supplementing this are the world markets

for energy and media management that are expected to reach 45 and 22.3 billion dollar respectively.

The market is there, but it is a matter of being able to market smart home products, which can be produced by different manufacturers, are plug-and-play enabled, offer a level of autonomous behavior based on AI, and are cheap.

Regarding the inflexibility and poor manageability is discussed in subsection 3.3.1 and revisited in subsection 7.2.4. The difficulties in achieving security is outside the scope of this work (Section 1.3), but the topic is discussed ad hoc when needed for clarify other topics.

3.3.3 ARTIFICIAL INTELLIGENCE IN SMART HOMES

Commonly the term smart home creates associations to a home that are able to think on its own and act intelligently by using some kind of AI. However, as stated earlier this kind of smart homes is not what is available on the market. The services and devices offered today (2013) by smart homes are much closer to what are offered by the home automation area. So, today real smart homes are not available at a commercial level, they primarily exist at a laboratory level e.g., as living labs. One example of such a smart home living lab is the *iSpace* that is resident at the University of Essex. It is fitted with intelligent gadgets that can detect and learn from the occupant's behavior and thereby suggest services that could improve the quality of life or suits user's needs. These gadgets can communicate with each other, coordinate their actions and they allow remote access via Internet and GSM (UniversityOfEssex, 2013). From a more theoretical point of view the research area dealing with adaptive AI in smart homes is in its infancy. Some examples are (Cheng et al., 2009), (Medjahed et al., 2009). Cheng et al. (2009) discuss an adaptive scenario-based reasoning system based on simple user descriptions and a lightweight learning methodology. This system is only partly adaptive, and it uses a non-portable simplified user profile management system. Other systems employ "Case-Based Reasoning", but the case-to-case based learning approach puts a huge demand on computational resources. Arabnia et al. (2010) predict that the partly intelligent smart home area will play an important role in the IoT technologies to come. Another approach is an adaptive calendar concept updating users' calendars based on their recognized activity (Yu et al., 2010).

3.3.4 THE RESEARCH FRONTIERS OF SMART HOMES

Even though a lot of methods and knowledge on human reasoning, semantics, rule-based system and AI are available, very little has been done on how this could be used to manage distributed smart homes and their contained S.O.s.

Silva et al. (2012) review the smart home state-of-the-art from two viewpoints, where they cover the categories technology and service. They conclude that the increasing amount of embedded computer power together with the fast sensors development rate will promote the use of AI systems based on distributed sensor network. These intelligent network nodes will then be able to learn and predict events⁴ in real time or do some off-line processing. They also forecast that the architecture probably will be multi agent-based and that these systems will be in form of an add-on to existing technologies.

Another paper that reviews the past, present and future in the light of previous smart home research and technologies forecasts a similar prophecy (Alam et al., 2012). Their paper sets several directions on the smart home future research. One direction is the use of a middleware layer to integrate heterogeneous devices so that multivendor devices can coexists in the future. They also predict that the future smart homes will apply distributed devices and intelligence in form of smart devices. Regarding the user interface they conclude that it probably will be auditorily or visually based. Seen from a service point of view they predict that future healthcare service providers will use smart home technologies to provide remote healthcare services, especially to elderly who do not require intensive healthcare support. From a more global point of view they also predict that the concept of the smart grid coordinating the global electricity distribution and load.

However, these papers only cover some theoretical considerations and prophecies that have not been manifested into systems. This is mainly because the technology is not available in a mature functional form yet; only technology fractions exist in form of home automation

⁴ An event is a message from a sensor that is sent to S.O.s or devices.

devices and some theoretically based laboratory models. Thus, this area still needs to be researched and clarified in the future – the research frontiers are not at that level yet.

Some examples are provided in the following to illustrate the smart home state-of-the-art frontiers.

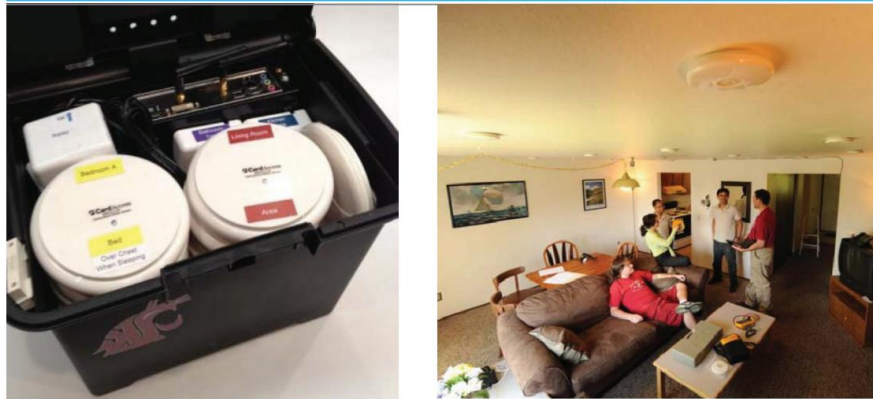


Figure 3.5. CASAS smart home components (Cook et al., 2012).

The state-of-the-art (2013) CASAS project “A Smart Home in a Box” is illustrated in Figure 3.5 (Cook et al., 2012). It is a lightweight design that is easy to install and it provides smart home capabilities out of the box. This approach implements AI in a centralized form which uses sensor data stored in a central database. While they have achieved to reducing the infrastructure barrier, their approach still has the centralized system drawbacks. In addition, the AI system still needs training in advance, i.e., no real-time learning.

The “ThinkHome” project has been carried out by a research group at Vienna University of Technology in 2009. Even though it is some years old it is still state-of-the-art in distributed agent-based smart home systems (Alam et al., 2012). The “ThinkHome” project aims to design and implement an adaptive control mechanism for maximizing the energy efficiency in building. This mechanism uses a concept where an ontology knowledge database processes and stores the learning, which has been supplied from a set of highly specialized agents in a multi-agent system (Reinisch et al., 2010). Each agent has its own scope such as the AI based agent, the user preference agent, and the context inference agent. A user interface interacts with the multi-agent system to control its behavior and preferences.

3.4 INTERNET OF THINGS

The Internet of Things (IoT) research area is important in the smart home context because the contained devices (S.O.s) share technology and functionality with it. The IoT is considered as the next step in the evolution of the Internet. The EU Commission has written an IoT action plan for Europe (EC, 2009), stating that IoT will drastically modify the way our societies function in the coming 5 to 15 years. A combination of Internet and emerging technologies like wireless communications, context awareness and embedded wireless sensor networks transforms everyday objects into intelligent and context-aware S.O.s. The technology resources are already available, and the development of IoT is more a matter of technology structures, market shares, values and earnings.

A variety of application and technology drivers exists for the IoT area. Some of these are presented in the following.

Valhouli (2010) argues that the intelligent green building and the AAL (Dohr et al., 2010) areas are important drivers. In addition, they find that proliferation of intelligent and green buildings has the potential to drive adoption of sensors and devices with embedded processing power.

According to Bandyopadhyay et al. (2011) the unquestionable main strength of IoT is its impact on every-days life for a potential user. They highlight areas such as AAL, smart homes, smart offices, e-health and enhanced learning. Similar they point out that business users will benefit from industrial manufacturing, logistics, business process managements, intelligent transportation systems of people and goods.

Jain et al. (2011) highlight a group of application areas for IoT which they predict as the main drivers. Thus, they forecast automated warehouses; smart medicine which can monitor the use and abuse; origin of food to avoid infections; monitor health by sampling sufficiently often to allow early detection and recovery improvement; smart buildings which control HVAC and lightning; intelligent transportation which is safe and energy efficient; avoidances of pollution and disaster by allowing early warnings and prevention of catastrophes.

As noted, the application area of IoT is widespread and includes many subareas. Some of these could be beneficial in the smart home area. Today, most of the commonly used objects (things) in the home only allow simple user interactions and they do not possess any processing capability. Typical examples could be a lamp equipped with a motion sensor or a battery indicator in a mobile phone. Most of these things belong to the home automation area.

In the future it is expected that the Internet as we know it today (2013) will be integrated into a multitude of things (EC, 2009), (Liu & Tong, 2010), hence commonly known objects such as clothes, food packing, toothbrushes, etc. will be equipped with some level of Internet-addressable AI. Thus, these S.O.s will offer context awareness and communication features, and they will share some level of pseudo-intelligence depending on their processing capability and consumed power limitation (Liu & Tong, 2010), (Mao et al., 2010), (Grønbaek, 2008), (Castellani et al., 2010), (Darianian & Michael, 2008).

This development will lead to new forms of communication between people and things and between things themselves. So, the challenge is to go beyond today's state-of-the-art, making these S.O.s context-aware, intelligent and able to communicate via IP, and combining them into a distributed system for the future smart home. They should be able to not only react to changes in the environment, but also perform AI-based reasoning to take into account the preferences of the user inhabiting the smart home. A lot of research is needed in this area (EC, 2009).

In an S.O. focus, the IoT area shares many common elements with it. Both areas need to be context aware, provide AI, have a low form factor, use limited power and memory resources, have the ability to communicate, provide a general (standardized) interface for independent manufacturing. Thus, the S.O.s are related to and based on the concept provided by the IoT research area. This strategy ensures that the S.O. concept developed in this work provides the basic knowledge and technologies to facilitate the future change from S.O.s to the coming world of IoT, without the need of a smart home revolution.

3.5 RELATED AREAS AND RESEARCH ACTIVITIES

A short overview of: related projects; related technologies and research areas; and related industry organizations are presented in this section.

3.5.1 RELATED PROJECTS

One of the most important related projects is the EU co-funded HYDRA project. It developed middleware for embedded systems including their networks. The primary aim was to offer a middleware concept to researchers that facilitate ambient intelligence, wireless devices and sensors (Eisenhauer et al., 2009). The project ended in 2010.

3.5.2 RELATED TECHNOLOGIES AND RESEARCH AREAS

The Machine-to-machine communication (M2M) is a related area which contributes with technology to the smart home research area. It covers communication between embedded processors, microcontrollers, smart sensors, actuators, and mobile devices with minimum or no human intervention. Researchers look into this area in combination with IoT and smart homes (Chen et al., 2011), (Chen, 2013), (Bandyopadhyay & Sen, 2011), (Starsinic, 2010). It is assumed that M2M and sensor related communications will be a large part of the communication in 2020 (Sørensen & Skouby, 2009). In addition, ETSI has formed a technical committee to conduct standardization in the M2M area (Bandyopadhyay & Sen, 2011). However, the M2M area has some challenges that need to be dealt with. One of these is the privacy issues, i.e., the authorization of access if no human is involved (Alam & Noll, 2010).

Embedded microprocessor and microcontroller technologies are an important subject for researching implementation of small low-power embedded autonomous systems. Thus they are in focus for the future IoT and smart home areas. Today (2013) they offer high-level processing capabilities for a low resource usage. Many manufacturers exist; however, in this work only the PIC18xx devices from Microchip are used and discussed. It is a state-of-the-art technological device that in average represents the achievements in the area. Thus some researchers use these devices from Microchip in their research projects (Yao et al., 2010), (Boegel, 2012), (Casilari et al., 2010). Similarly, the ZigBee SoC device CC24xx from Texas Instruments is a popular choice in smart home and IoT research (Yan & Dan, 2010), (Khan & Aziz, 2009), (Chen et al., 2013).

A SDR has the flexibility and characteristics needed for smart homes and M2M architectures and networks. Basically, a SDR is a collection of software and hardware that handle all radio functions. This approach facilitates updates in form of adding new wireless features and capabilities to existing radio systems without changing the hardware (Starsinic, 2010). Additionally, it enables construction of multiband and multicarrier gateways which are able to communicate simultaneously using different protocols and different frequency bands. In the IoT research SDR is believed to provide one of the needed means to integrate and reduce the extensive design space required (Sundmaeker et al., 2010). The future trend for the SDR technology is reflected in the progressing standardization of the next generation wireless systems, which involves SDR technology. Thus, standardization of systems such as Long-Term Evolution (LTE) advanced, higher data rate WiFi (IEEE802.11ac/ad) and cognitive radio are in progress to meet requirements for higher data rates and cheaper services (Araki & Morishita, 2012).

3.5.3 RELATED INDUSTRY ORGANIZATIONS

One of the important industrial organizations is the broadband forum. It is the leading industry organization with more than 180 active members (2011). It manages broadband devices and associated services in form of standards. Thus, in the connected home area the organization manages services in form of security, control, monitoring, health, and energy management (Fedosseev et al., 2011).

3.6 SUMMARY

This chapter provided an overview of the development throughout the last decade in the three areas *intelligent buildings*, *home automation*, and *smart homes*. It provided a definition of *intelligent buildings* and discussed what building management systems have provided in relation to AI now and in the future. The *home automation* area has provided remote control to the homes for controlling mundane functions such as light, heat, ventilation, entertainment systems, appliances etc. to improve comfort, convenience, energy efficiency, and security. However, this area did not provide any antonymous behavior.

The *smart homes* area is an extension of the *home automation* area. It was discussed in the light of autonomous behavior, drivers and barriers for the research area, and its possible user

interfaces. It was stated that AI in smart homes was not commercially available; however, they exist in living labs.

Finally, the lack of research in using AI in distributed smart homes was discussed. However, some researchers have forecasted that smart devices (agents) will be used in combination with AI and some of the driving mechanisms were discussed (Silva et al., 2012), (Alam et al., 2012). The usage of the IoT area and other related areas was presented and discussed.

4 DISTRIBUTED SYSTEMS AND SMART HOMES

This chapter introduces distributed systems at a general level (Section 4.1) and uses knowledge from this context to discuss and compare distributed and centralized smart home concepts (Sections 4.2 and 4.3). Following this the focus is set to discuss the challenges and use of sensors and actuators in the smart home context (Section 4.4). Finally, the smart home network topologies are discussed in the light of state-of-the-art research projects and technologies (Section 4.5).

Considering the first research question in the view of smart home distributed systems, it can be divided into some sub-problem areas. These are discussed in the following. For convenience the first research question is restated:

Question 1: *How should the distributed system architecture for smart homes be designed in order to incorporate AI and the diversity of S.O's?*

To clarify the *design* of a *distributed system architecture* that *incorporates AI* the term *distributed system architecture* needs to be explored. Viewed from a high abstraction level the smart homes are spatially distributed containers where the *diversity of S.O.s* are included in a distributed manner. These S.O.s or devices have processing power and are context aware (Shah et al., 2009), (Trevennor, 2012), (López et al., 2012). Thus, smart homes are related to the areas of distributed processing, ubiquitous computing and cloud processing, by nature. However, research in the distributed smart home area is limited because most research performed in the last decade has been done with centralized models (Section 4.3).

Smart home distributed and centralized models are discussed in section 4.3. The integrated S.O.s are small autonomous objects which behave similar to the IoT devices discussed in section 3.4. S.O.s sense their context and use actuators to affect it. To obtain this ability they either integrate with sensors and actuators or they interface with them. Challenges in this approach are discussed in section 4.4. As discussed, S.O.s are spatially distributed why they need a network in order to communicate. This challenge is discussed in section 4.5.

4.1 DISTRIBUTED SYSTEMS IN GENERAL

Today, computer networks are widespread in our society and they are important parts in our daily lives. A typical example is our mobile phones which use networks in form of Bluetooth, WiFi and 3G or 4G. Most of these phones provide connections to the Internet. On a smaller scale companies have factory and corporate networks; and universities have Campus-nets. Even our cars have a lot of computers and networks such as controller area network and vehicle area network bus systems. These enable the computers to communicate, exchange information and synchronize actions. All these systems can be included in the term *distributed systems*.

A distributed system is a group of networked computers that are physically distributed within a network area. These computers communicate and coordinate their actions through message passing. However, today (2013) this term is overloaded because it is also used for a collection of tasks that run on a single computer, where they interact with each other through message passing. While there is no single definition of a distributed system some characteristics are local synchronization, component concurrency and independent failures of components. Moreover, many researchers have tried to define a distributed system. Burns et al. (2001) define a distributed computer system this way:

A distributed computer system is defined to be a system of multiple autonomous processing elements cooperating in a common purpose or to achieve a common goal (Burns & Wellings, 2001).

Another more general definition of distributed systems is provided by Coulouris et al. (2011):

A distributed system is one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages (Coulouris et al., 2011).

From these definitions it is noted that the range of definitions is quite broad. However, similarities are also present because both definitions focus on independent processes or components that carry out tasks in common. By coordinating their efforts they are able to communicate, even if they are spatially distributed.

From the last definition and the distributed system characteristics discussed earlier in this chapter the main implications are *concurrency*, *no global clock* and *independent failures* (Coulouris et al., 2011). *Concurrency* means that in theory it is possible to expand a distributed system by adding more resources, but the disadvantage is that more resources also need to be handled. *No global clock* means that it is not possible to derive a single global correct time, because of the limited accuracy in real implementations. *Independent failures* add new ways a distributed system can fail. Thus, if a connected computer fails it is disconnected from the network. However, it could be able to run in a reduced way and offer some degraded services. Additionally, other components in the network do not necessarily fail as a consequence of this singular failure, i.e., the non-failing part in a distributed system is often able to continue its service delivery.

Examples of distributed services in our daily lives are the Google web search engine that uses distributed infrastructure to do web searches. Another driver of distributed services is online gaming where many gamers join virtual environments for playing multiplayer games (Ericsson, 2011). Moreover, the area of financial trading requires that stock price information is processed and provided at different places at exactly the same time to prevent fraud.

Related areas that are partly adapted in the distributed system contexts are:

- pervasive networking. It describes the multitude of ways connections can be made from computers and smart phones in an online fashion.
- ubiquitous computing which covers many small computers that are present in the users home, often in an invisible way.
- cloud computing, which is a set of Internet based applications that offer storage capacity and computing service.

The area of distributed systems has a number of challenges. The most important of these are *heterogeneity*, *openness*, *security*, *scalability* and *handling of failures* (Coulouris et al., 2011). Network *heterogeneity* and transparency deal with the challenges in combining many heterogeneous networks and devices into one common system. Different methods are available, but often a decoupling middleware layer is used for interfacing S.O.s with each other. *Openness* and *scalability* express how easy it is to expand a system with new devices.

In this work it is assumed that devices can be added seamlessly, nevertheless, some initial setup is needed. *Security* covers the topics availability, integrity and confidentiality, whereas *handling of failures* deals with how to recover after failures. This work does not deal with these topics, nonetheless, much of the S.O. technology is highly correlated with the field of wireless sensor networks, so security and failure handling are to some extent inherited from this field (Section 4.5).

Distributed system models can be arranged into a few categories (Coulouris et al., 2011). These are *physical models*, *architectural models* and *functional models*. The *physical model* captures the hardware layout of a system in terms of computers and their interconnecting networks. The *architectural model* describes the system in terms of aggregated computational elements that are interconnected by a network. Finally, the *functional model* is an abstraction that characterizes the functionality of each individual element in a distributed system. This work uses a mix of these approaches, because the smart devices and their embedded AI systems need to look into the spatial and computational network elements.

4.2 DISTRIBUTED SYSTEMS IN A SMART HOME CONTEXT

Narrowing the concept of distributed systems to a smart home context brings forward many of the problems found in this area. Most SHS today (2013), which actually only exist in the laboratories (Cook et al., 2012), consist of a collection of heterogeneous systems that need to cooperate and communicate to perform their task effectively.

Basically, these systems can be divided into three categories. One category is the entertainment network that often is based on WiFi. This network supports devices such as computers, Internet enabled TV, and Internet enabled radios. Second category is the smart home network (SHN) that can be based on low bit-rate networks such as ZigBee, Bluetooth low energy, X10, INSTEON, and Z-Wave. This network supports sensors and actuators that learn from the user's behavior and offer services accordingly based on AI. These two categories are able to communicate by using gateway devices. However, the SHN does not provide the needed resources to carry network traffic related to the home network. In contrast, the home network devices can be part of both categories, i.e., they can contain separate elements from both categories. An example is a smart TV that can stream video from a Wi-Fi

based Internet connection and at the same time send simple event based information about the TV is turned on or off to the SHN devices. The third category is the body area network that relates to devices carried by a person such as a smart phone. It behaves similar to the home networking category. The discussed categories are illustrated in Figure 4.1 and discussed by Chen et al. (2011) in their work.

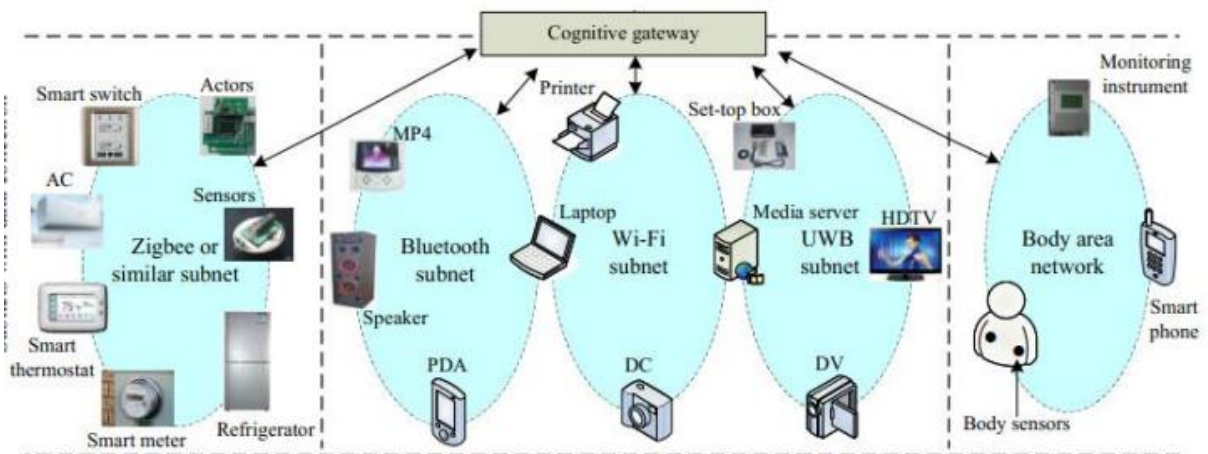


Figure 4.1. The different networks in a smart home: Smart home network (SHN), home networking, and body area networking (Chen et al., 2011).

By nature the smart home devices are physically distributed, i.e., they are placed at different spatial locations. Most of these systems are developed by different manufacturers who in lack of a single standard use known technology and common knowledge. This means that today's available smart home (i.e., home automation) devices use different operating systems, offer different services and use different programming platforms (Turner, 2011).

Researchers have suggested different solutions to these problems. One solution preferred by researchers is using a middleware layer. Examples of this approach are Jini (Gupta et al., 2002) and UPnP (Lee & Kim, 2007). Another solution is the Common Object Request Broker (CORBA) architecture (Artemio & Leonardo, 2012) that provides an object oriented way to interface devices and thereby obtain interoperability. Nevertheless, CORBA is suboptimal because the client needs knowledge about server method names, and the specifications do not consider interfaces with systems that are not CORBA enabled. Additionally Microsoft offers their variants, which are the COM and .NET frameworks. These solutions are for Microsoft Windows and provide almost the same functionality as CORBA. Nonetheless, they only

support Windows, which is not suitable for small low-power embedded systems and they suffer from the same drawbacks as CORBA.

Sun Microsystems has developed middleware systems that are based on java remote invocation, i.e., objects are activated by calls from the remote end. One example is the Remote Method Invocation that allows objects to be called remotely from other applications in a heterogeneous network. The Open Service Gateway Initiative (OSGi) framework is another example that defines a standardized interface for heterogeneous devices. This has been used by researchers in smart home contexts (Cheng et al., 2008), (Wu et al., 2008). Nonetheless, the disadvantages are that OSGi requires a java virtual machine to run and it needs a few hundreds of kilobytes of memory, which is an unrealistic amount on small low-power embedded devices (Jain et al., 2011).

Some researchers use web services that are a standardized collection of methods for interaction between applications, devices and services. Some of the popular ones are Simple Object Access Protocol and Web Service Description Language. They are based on XML and provide generic interfaces that are web based. Using a web based interface in a smart home is disadvantageous because it adds overhead in the used “mark up” language and it uses processor resources to run the HTTP protocol. This is evident by comparing it to an approach that uses a simple token based system such as X10 (Subsection 4.5.3). In addition, these web-based services need the Internet protocol as a backbone. In addition, it is disadvantageous because embedded Internet variants consume a considerably amount of memory for the IP stack, etc. and the protocol handling is costly in terms of processing power (Kim et al., 2008). Nonetheless, the 6LoWPAN standard offers low resource IP routing in smart homes (Subsection 4.5.4) based on the well known Internet architecture.

As stated in the research question this work offers a distributed system architecture that includes a generic framework, which contains S.O.s implemented on small embedded processing platforms. These S.O.s provide AI (Chapter 8) in a distributed form. Such a generic concept is new and useable in many of the contexts discussed in this section. Additionally, it could be used in cars to learn the user habits such as adjusting the seats, turn the radio on and control the air condition. In an AAL scenario S.O.s could supervise important behavior parameters and issue an alert if a person has fallen, etc. Furthermore, in

the area of green technologies S.O.s can control HVAC and electric installations such as light and electric appliances.

4.3 CENTRALIZED AND DISTRIBUTED SMART HOMES

As discussed, the smart homes area extends the more simple home automation area that is based on technology developed for building automation. This means that the home automation service and functionality area is dedicated to simple remote control and supervisory functions. These systems were mainly implemented on centralized servers because the technology and smart home knowhow was limited. Thus, the battery and embedded processing technology were not able to build small size communicating, context aware nodes that offer AI (López et al., 2012). So, the challenge for the next evolutionary step from home automation to real smart homes is to make use of today's (2013) technology and evaluate to what extent it is possible to fulfill the challenges. These are: intelligent behavior, context awareness, smart devices and a rich level of connectivity.

4.3.1 CENTRALIZED, AGENT AND CLOUD BASED SYSTEMS

Centralized smart homes use architectures with a single home automation server connected to the Internet. This server receives all sensor events and it runs the AI algorithms. Based on sensor inputs and predictions it schedules services to the user. In this context the main difference between the centralized and the distributed concept is in the underlying network that supports the devices with communication capabilities.

The example illustrated in Figure 4.2 covers the wireless sensor and actuator network SHS architecture proposed by Lameski et al. (2011).

As noted, the devices are connected to an automated home network with a server in the centre. Many of these devices use a proprietary defined network technology because there is no single home automation standard available. Thus, home automation networks often make use of a mix of protocols like ZigBee, Z-Wave and WiFi, (Zhao, 2010), (Rathnayaka et al., 2012), (Mao et al., 2010).

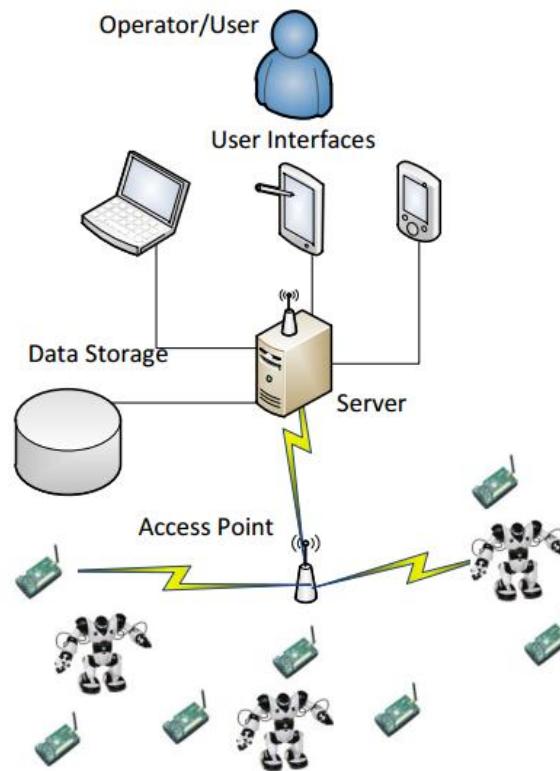


Figure 4.2. The wireless sensor and actuator network centralized smart home architecture (Lameski et al., 2011).

As stated, most SHS research is based on a centralized approach. This is also the case for most multi-agent-based systems. An example is the “ThinkHome” smart home (Subsection 3.3.4) that contains few high-level specialized agents. Only one agent handles all computations, so it is a centralized system (Reinisch et al., 2010). A similar approach has been presented by Hanzi et al. (2013) who use a central *control agent* in a star topology. Hannon et al. (2005) present an agent-based system where the agents are specialized to handle a high functionality level, such as: kitchen agent, entertainment agent, control system agent, and HVAC agent. However, their topology is centralized around a manager placed in the network center.

In the process of moving from automated homes to real smart homes the availability of cheaper faster Internet connections and cloud services causes a gradual replacement of centralized automated home servers with cloud based solutions. An example of such a

concept, provided by the Broadband forum (Fedosseev et al., 2011), (Subsection 3.5.3) is shown in Figure 4.3.

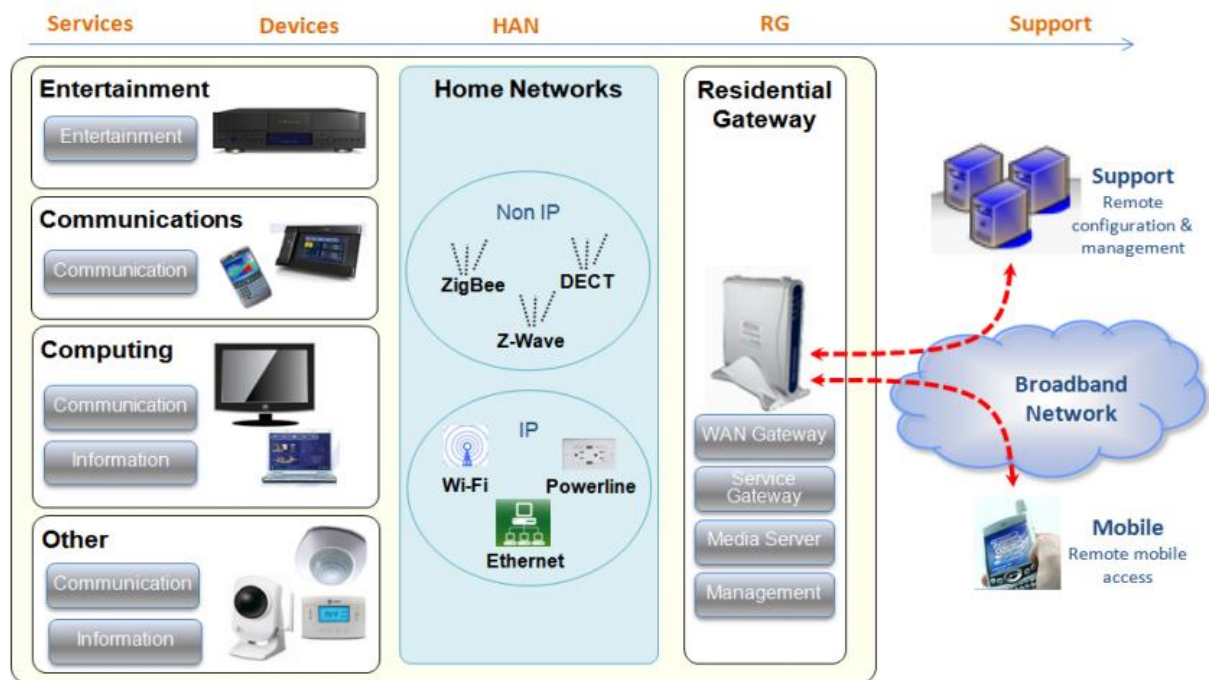


Figure 4.3. A general architecture of a home automation concept (Fedosseev et al., 2011).

A cloud solution provides benefits in the form of increased computer processing power and data storage capacity. In addition, it releases the user from the task of installing, updating and backing up software on the centralized automated home server. In the near future it is also likely that device information can be retrieved online for easy device setup of similar devices. However, a downside is that devices may stop working or downgrade their service level depending on the different system errors. Furthermore, the security and the network load also need to be addressed. Examples of architectures with cloud-based automated home devices and network elements are the Honeywell Smart Home and the Control4 systems (Ye & Huang, 2011).

4.3.2 SMART HOME DEVICE COMMUNICATION

The SHNs need to handle data exchange between its connected devices. It also needs to communicate with systems outside its own context such as cloud services. Thus, an important backbone in this concept is the underlying network that supports the devices with these

communication capabilities. These network systems can be divided into different logically parts and they can be based on both centralized and distributed topologies.

In general, the networks can be divided into a topology part and a functionality part as illustrated in Figure 4.4. The topology part relates to the physical network, i.e., the way nodes are connected and organized. The functionality part refers to the way functions are divided among the network devices and it describes how resources are shared. So, functionality looks into what function the nodes provide, but not how it is implemented – this is a job for the topology part.

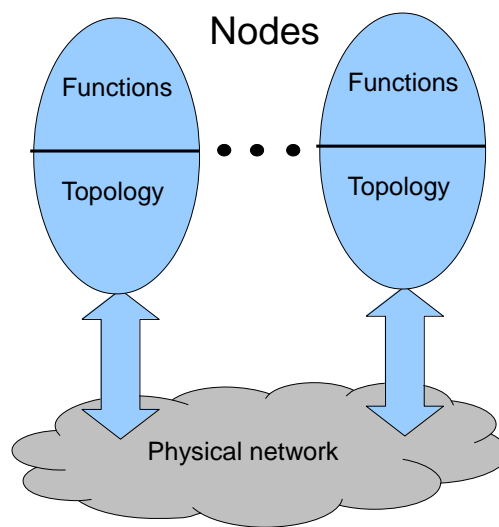


Figure 4.4. The smart home node functionality and its physical network topology.

Elaborating over the node functionality part illustrated in Figure 4.4 some common network node functions are given in Table 4.1. A more detailed coverage of this subject can be found in (Starsinic, 2010), (Perumal et al., 2011).

Type	Description
Sensor node	A simple sensor node is able to receive information from a sensor and transmit it to a central point. More advanced sensor nodes are able to do a small amount of processing on sensor data before passing it on.
Actuator node	An actuator node receives information from a central point and controls an actuator based on this information. Some examples are controlling a kitchen lamp, TV, etc.
Gateway node (sometimes named bridging node)	Interface the heterogeneous protocols used by the low-level sensor nodes. An example is bridging the two technologies Blue tooth and ZigBee, so they can co-exist in one SHN. It also provides access from the outer Internet-based world to the SHN.
Routing node (sometimes named forwarding node)	This node only provides routing capability. So it simply receives a message and passes it on to the next node(s).

Table 4.1. The most common node functions found in smart homes.

These common node functions need to support the SHN devices with communication capabilities so they are able to communicate. One of the most important elements that need to be addressed, from a smart device point of view, is the low data rate device-to-device communication (often named machine-to-machine (M2M) communication), (Subsection 3.5.2). They are typically battery powered devices and they have very low power consumption. In many cases, they autonomously communicate with each other or with the central controller. Creating M2M networks in smart homes requires that the different protocols interface and elements communicate with each other in an effective and efficient manner (Starsinic, 2010). Different principles for this communication exist, but one of the most efficient ones is the use of multi-agent-based systems (Reinisch et al., 2010).

In general, at a high abstraction level, distribution of information in SHNs is needed to support the services that are offered to the user. When the user performs scenarios sensors are triggered and events are emitted. These events are collected and processed by some system placed in the SHN context. Based on its decisions a command can be emitted to the actuators, which in turn offer a service to the user. Performing this scenario requires that information such as sensor events, actions and actuator commands can be distributed through the SHN.

4.3.3 A SIMPLE SCENARIO

A simple scenario is presented to illustrate the amount of information that needs to be exchanged. It contains a living room with four sensors, i.e., ceiling lamp, table lamp, room sensor and an armchair sensor. In addition it contains ceiling and table lamp actuators Figure 4.5.

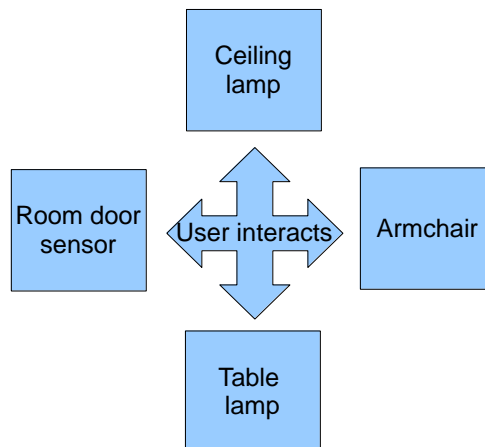


Figure 4.5. Living room - a simple smart home scenario model.

Many scenarios can be derived from the smart home scenario model Figure 4.5. However, an illustrative one would be to consider a scenario where the user enters the room at a particular time window T , turn off the ceiling lamp, sits down in the armchair, turn on the table lamp and open a book for reading. By repeating this *action*⁵ scenario over time the table lamps could learn this particular behavioral. That is, the table lamp turns on automatically when the user enters the room in the trained time window T , turnoff the ceiling lamp and sits down in the armchair. It is noted that the sequences are not important, i.e., the order of the sequences can be changed.

⁵ An action is a single atomic state in an activity. I.e., an activity consists of many actions.

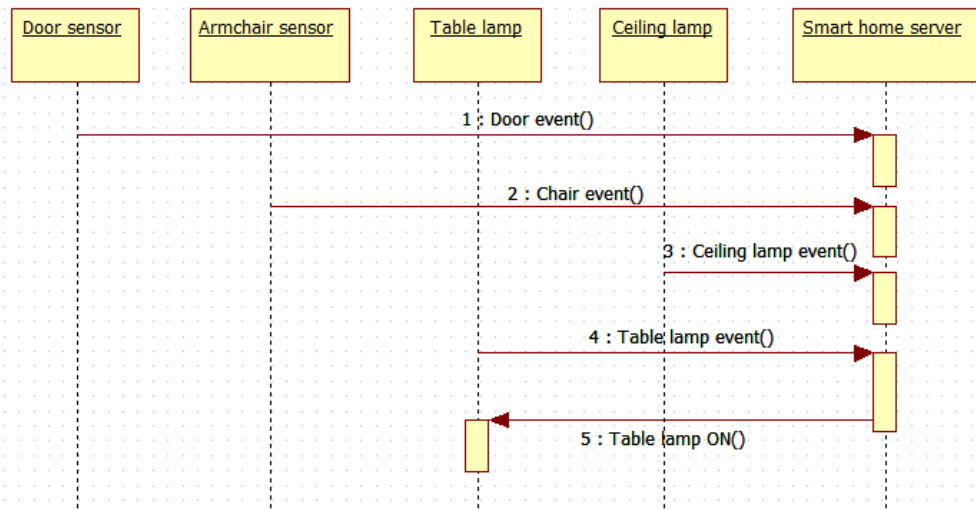


Figure 4.6. Sequence diagram for the living room scenario, centralized server.

Looking into the event distribution in Figure 4.6 the complexity of such a simple scenario becomes visible. Firstly, all sensor events need to be sent to one centralized server. This requires network bandwidth and it consumes sensor power. Network bandwidth is used because all sensors are allocated in the same network and therefore share its bandwidth resources. Power is used by the wireless sensors to transmit and route all the messages, this drains the batteries. Some network nodes need to use high transmits power to communicate because its adjacent nodes are in a shadow of walls, furniture's, etc. (Shah et al., 2009). But, high transmit power increases the interference level, that in turn requires even higher transmit power from the neighbor transceivers. Secondly, the server needs to send commands to the actuators. These commands can be complex and be time consuming, because some actuators require processing power to e.g., find a certain position or a certain light level. In addition, centralized server based SHSs have some challenges as discussed in section 4.3.

4.4 SENSORS AND ACTUATORS

In smart homes sensors are the primary source for information of user activity. This information is used by the artificial system to get a fragmented picture of the user current activity. Sensor technology has evolved through the last decades and today (2013) small integrated low-power sensors exist. So, the challenges to find which sensors are the most suitable for smart homes, how will the measured data be used, and where should they be installed.

A lot of different sensor types exist ranging from environmental sensors to biologic and medical sensors. Nevertheless, this section only deals with location sensors because they are the relevant group for smart homes in this work.

4.4.1 SENSORS

Location sensors are normally used to detect a change in context state caused by the user. For example, if the user takes a cup from the cupboard a sensor registers this. In general, location sensors are members of the groups: Simple switches, pressure mats, passive infrared detectors, radar, sound, light, and camera based sensors. Camera based sensors are not used much in smart homes, because users consider them to be too invasive (Cook, 2012). In addition they produce a huge amount of data that needs to be processed and transferred to the artificial system. However, they offer the possibility to build gesture based user interfaces (Kim & Kim, 2006). Sound (including ultrasound) and light sensors are used too, but they suffer from considerable costs, high current consumption, using comprehensive processing resources, and detection uncertainty compared to simple switches. However, they are able to provide high resolution contextual measurements. In general it is possible to transform multi-valued sensor outputs to binary ones by using programmable thresholds in the sensor. An example could be a temperature sensor that is programmed to send events when the temperature drops below 0 or rises over 30 degree centigrade.

Most sensors in smart homes are binary of nature, i.e., they only behave in an on-off manner. They offer simplicity and low network loads because only binary values need to be transferred. Tapia et al. (2004) argue that the cost of putting a lot of advanced sensors in a smart home often restrict its deployment. As a solution they developed a cheap and small simple on-off sensor. Brush et al. (2011) supports the view that cost is one of the largest barriers that prevents smart homes from being adopted. So, focusing on simple switch sensors is beneficial from a cost perspective. A simple sensor framework has been used in a work by Orr et al. (2000) they build a pressure sensitive floor to detect the user location. MIT researchers use simple switch sensors to detect user activity in form of state changes (Tapia et al., 2004). An example of using another approach, which can be quantized into binary data, is presented by Cho et al. (2010) it extracts user activities from the power consumptions in the smart home.

In general, sensors change state when they are triggered. If a sensor has a transmitter, but no receiver communication link handshake is not possible. Thus, the communication link needs to handle this. One approach that deals with this is the event driven, where the sensor transmits an event whenever it is activated. Such an approach uses power only when some change in the sensor state is detected. However, if the event is lost, the network consumer is not notified again. Another approach, which uses more power, is transmitting sensor status periodically e.g., every second. Such an approach increases the likelihood that a sensor state change will be detected, for the cost of consumed power and increased network load. A third approach is using a polling approach where the consumer asks the sensor if some changes have been detected. In this approach events are only transmitted when the consumer needs it. But, this requires that the sensor node has a receiver and not only a transmitter. A more comprehensive discussion of transmit-only sensors is provided in subsection 4.4.3.

When the sensors are triggered by some user activity there is a risk that this causes multi events. For example, if the user does not close a closet properly the switch could detect vibrations from the environments and emit a cascade of on-off events. Obviously, some algorithm or filter to prevent this from happening is needed. The question is where to place this filter – in the sensor or in the event consumer. This choice is a balance between bandwidth and power. Thus, if the filter is placed in the sensor, bandwidth is saved for a cost of doing more processing in the sensor node. Placing the filter in the consumer node moves the power consumption from the sensor node to the consumer node, but more bandwidth is used for the repeating events. Nonetheless, if the consumer node is powered by, e.g., the mains and the bandwidth load can be accepted this approach could be beneficial.

4.4.2 DEVICE POWER CONSUMPTION

Power consumption in SHNs is a problem if the devices are powered by batteries. Changing batteries on sensors and devices is a non trivial task. Often these sensors are hidden in a door frame or build into a closet, i.e., they are hard to reach. This problem increases with the amount of available sensors that can be more than one hundred in a smart home (Zhao et al., 2013). Another issue is the physical size of the used batteries because it needs to be small enough to fit into the low form-factor sensors and devices. But the dilemma is that, a small size battery also has a small battery capacity. It is expected that sensor devices will become

smaller in the future so small batteries are needed. Reducing the sensor sizes also provide another problem in form of heat. Small integrated circuits have a problem with cooling, because they have a small surface area.

To reduce power consumption in SHNs the amount of information transmitted need to be limited. Firstly, it can be limited by sending short atomic messages over a short distance. Secondly, organize the communication so it is possible for the sensor node to power sleep. Thirdly, use a mix of mains and battery powered devices (Jin & Kunz, 2011).

So, saving power in smart home devices are of high importance for low bit rate networks and battery powered devices (Yao et al., 2010). Further information can be gained by looking into a mix of power supply types and processing capabilities for the sensor nodes (Shah et al., 2009). In their work they mix both processing and power supply types to achieve better performance. However, the cost is a complicated concept that uses a broadcast approach.

Regarding the mains powered nodes they do not have the power limitations found in the battery driven ones so they offer a possibility to route messages from battery driven sensor nodes (Xu et al., 2013). This principle has been researched by Starsinic (2010).

A future approach is the concept of energy harvesting. It could be used for powering low-power sensors and their build in electronic circuits (Drew, 2012), (Boegel, 2012). Energy harvesting is a process where energy is harvested from kinetic energy. An example could be opening or closing a door. However, this technology is still in its infancy why it is not sufficiently mature for use in smart homes.

With respect to the smart home sensors and devices the main challenge is increasing the battery lifetime. Different possible focus points have been discussed earlier in this section. They are: Small messages, node power sleep, using a mix of battery and mains powered devices.

4.4.3 TRANSMIT ONLY NODES

In the discussion of saving battery power a major challenge is the possibility for sensor nodes to have a transmitter only, i.e., no receiver. The advantage is that it supports dense and long lasting applications deployed at a very low cost and power consumption (Blaszczyszyn &

Radunovic, 2008). However, the lack of receivers in the transmit-only nodes renders most existing MAC protocols invalid.

Zhao et al. (2013) address these challenges by developing a new architecture that supports transmit only nodes as well as standard nodes with a complete transceiver. The problem with having no receiver is that these nodes cannot synchronize with other nodes, e.g., by using a carrier sense multiple access scheme. This means that when a large number of nodes use such a greedy scheme the network performance will deteriorate fast. To overcome this they presented a scheme based on a time division multiplexing scheme without control, where all nodes transmit the same message (frame) randomly a number of times. The more number of times one node transmits the frame the more likely it will be received without collisions from the other frames. This enables the use of prioritization, which is why they defined two groups covering low priority (LP) and high priority (HP) frames. Yoon et al. (2007) showed that this scheme can achieve 97 % delivery probability in a 100 node Wireless Sensor Network (WSN).

Zhao et al. (2013) also showed that it is able to improve the delivery probability by requiring that the HP nodes to have a receiver. The HP node receiver was used to stop further transmit trials if an acknowledgement was received from the sink node. This lowers the collision probability and thereby increases the data delivering performance. Finally they presented a scheme where the pseudo-random generator seeds in the LP nodes were mirrored into the sink node. This means that the sink node is able to inform the HP nodes about what time interval to avoid because it is used by one of the LP nodes. One of the disadvantages in this scheme is the need for clock synchronization in all the nodes without having a receiver in the LP nodes.

To support layer 3 in the OSI model a link layer (layer 2) and a physical layer (layer 1) are needed. In general, the link layer takes care of contention, channel sharing i.e., MAC functionality, timing and signal processing. In general, factors as node power consumption, transmission delay, throughput, robustness, scalability, stability, and fairness have influenced the design of MAC protocols (Sohrabi & Pottie, 1999). Nonetheless, in transmit-only sensors this functionality is reduced and thereby indirectly improves the power savings.

In smart home context the ZigBee protocol based on the IEEE 802.15.4 standard is very popular. Another protocol that is also based on this standard is the 6LoWPAN. It could become popular too in the future because it is IP based and supports direct Internet connections. However, today (2013) it is too complex and resource consuming for S.O.s based on small, cheap, low-powers microcontrollers. The benefits that make ZigBee popular are: the build in encryption in form of Direct Sequence Spread Spectrum (DSSS), it is standardized, it enables asynchronous modes, and it offer development support for smart homes developers. None of the other standards offer all these benefits. Nevertheless, as descried earlier it is expected that a mix of all these protocols will be seen in the future.

4.4.4 SENSOR CLUSTERING

Combining smart home sensors and devices in clusters provides benefits as allowing most of the nodes to enter power sleep mode and thereby save battery lifetime. Furthermore, the interference level is lowered for the wireless clustered nodes. This is achieved by reducing the needed transmit power for reaching the nearby cluster head (Lynggaard, 2013c).

From a communication point of view many devices are often handled by organizing them into a cell like structure. This has been used for many mobile phone systems like the GSM and LTE systems, where the primary reason is lowering the interference level and mobile terminal power consumption. In WSN a similar structure named cluster-tree topology is used (Radi et al., 2011), (Sohraby et al., 2007). This WSN technology is often adapted to the SHNs.

From an algorithm point of view the Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm uses clustering techniques to enable node power sleeping. But, it does not optimize the choice of the cluster head in relation to save power. However, this problem has been solved in the Hybrid-LEACH algorithm (Azim & Islam, 2009).

The challenge in SHN clustering is founded in the immaturity level these possess. Thus, most SHNs are constructed by simply using a non optimized technology such as ZigBee or ZWave. They use their own default clustering methods, which do not take smart home conditions into consideration and they are designed for general purpose WSN's. Some researchers look into combining different networks into a single one, e.g., by using middleware concepts (Pensas & Vanhala, 2010). Most of these systems do not adapt to the special circumstances and

conditions that are given in a smart home context. However, ZigBee offers support to other cluster structures than the default ones (Khan & Aziz, 2009).

4.4.4.1 Clustering and Interferences

Viewing the defined cluster based S.O. and sensor topology in smart homes from an interference and power consumption point of view, underlines its usability. In a radio based system many types of interferences are in play. One of the most severe interference types is the co-channel interference that expresses the wanted signal power divided by the disturbing signal power (both at the same frequency). If this co-channel interference ratio is high, the wanted signal can be received with an acceptable bit error level. Other interference types are fading and adjacent channel power, but they are less severe than co-channel interference in smart homes (Lynggaard, 2013a).

Another major source for interference signals in smart homes is a WiFi router that often works at 2.4 GHz. Its frequency is in the Industrial Scientific Medical (ISM) band where most of the WSNs are allocated. These WSNs provide the lower layer interface for the S.O.s. Thus, WiFi routers will impose interference on the WSN's and thereby on the S.O.s (Lynggaard, 2013a). Yao et al. (2010) performed a series of experiments where they have found that the common 802.15.4 based WSN network capacity will drop to below 26 percent in some WiFi scenarios. A similar work by Hou et al. (2009) confirms this problem. In addition they have studied the radiation from a microwave oven and concluded that it would cause packet losses of approximately eight percent in a radius of 1.5 meter.

The consequences of these interferences are that the retransmission rate in the SHN will increase and the battery powered nodes are forced to increase their transmit power and thereby their power consumption (Subsection 4.5.2). Another factor that increases the node transmit power is shading, where the signal is lowered by disrupting furniture and walls (Shah et al., 2009).

Summing up, the challenge is to reduce the interference level in smart homes for saving power in the battery driven sensors and S.O.s. Selecting the sensor and S.O. spatial locations

is of high importance. This selecting process must take into consideration the intercommunication and out-of-context communication needed by the S.O.s.

4.4.5 SMART OBJECTS / AGENTS

S.O. is not a standardized exact definition, but merely a term that has different meanings in different contexts. Similar often used terms are smart devices, intelligent products and smart parts (López et al., 2012). For example the S.O. alliance (2013) has one description that associates the S.O. term with 6LoWPAN enabled IoT's. In another description they state that S.O.s are:

S.O.s are small computers with a sensor or actuator and a communication device, embedded in objects such as thermometers, car engines, light switches, and industry machinery. S.O.s enable a wide range of applications in areas such as home automation, building automation, factory monitoring, smart cities, structural health management systems, smart grid and energy management, and transportation (Alliance-IPSO, 2013).

López et al. (2012) define IoT based S.O.s as objects that: Have a unique identity and are able to sense and store measurements; transfer data and identification to other objects; and make decisions about themselves. Satoh (2009) states that some S.O. types are unable to communicate because they are missing the necessary resources. Therefore, S.O.s with limited computing and communication capabilities should delegate these with other S.O.s that have these resources available; this includes smart home servers (Satoh, 2009). Trying to combine the definitions from the different areas into one that can be used for smart homes could be beneficial. By looking into the discussed different definitions it is possible to find common elements that provide consensus of what an S.O. is. Thus, S.O.s:

- must be distributed in the smart home
- must contain computer power to some extent or be able to hand over tasks to other systems that have this resource
- must be able to communicate with each other and the external context
- must be able to handle sensors and actuators.

The state-of-the-art research in the S.O. research field uses a concept based on S.O.s that are named *agents* (Del-Hyo et al., 2012),(Arabnia et al., 2010),(Cook, 2012). An agent covers an isolated functionality or a domain in a smart home, e.g., one agent controls the heating and another controls the light.

A collection of agents are named a multi-agent system. Some illustrative examples of these systems are presented in the following. In the smart homes area Hannon et al. (2005) present an agent-based model, which is based on the biologic area. They suggest an agent-based distributed system as described in section 4.3. The agent-based smart home approach is also supported by Cook et al. (2012). They have built an experimental smart home in their laboratory named “MavHome”. This model is based on multi task agents where each agent addresses a specialized part and then coordinates with the others. Thus, one agent could handle HVAC and another one the power lines. Ling et al. (2002) present a smart agent framework for smart homes. They allocate five agents that divide the smart home functions into categories such as: Functions, interfaces, preferences, resources and controls.

Looking into the future the development in processing power and power consumption for small embedded processor devices, controllers and wireless transceiver technologies will most likely improve considerably. These technologies are the backbone for developing a distributed S.O. device that is approaching an IoT device and is able to offer distributed intelligence (Chapter 5). Alam et al. (2012) supports this prediction:

Although smart home research was initiated several decades ago, it still faces problems of immature home intelligence because of inadequate algorithms, improper activity recognition methods, and low rates of prediction accuracy. Providing distributed intelligence to all appliances may be an effective solution because it removes the burden of processing huge amounts of information from a central intelligent system. Each device will be responsible for its own domain and share only important information with the central intelligent system. The system will eventually transform into a multi-agent system with distributed intelligence by integrating smart appliances (Alam et al., 2012).

S.O.s need processing power to be able to offer their services. This processing power needs to be provided by small cheap embedded processors that fulfill the S.O. concept. Actually, many different processors and wireless platforms have been used by the researchers to perform experiments. Trevennor (2012) discusses smart home enablers and demonstrate some wireless sensor technology in the form of a homemade processing module based on a low-cost microcontroller and a ZigBee module from Sparkfun. The purpose of this work is to demonstrate that wireless nodes can be small and cheap. Chong et al. (2011) look into the IoT area to implement an SHS. They use a 8051 processor and the CC2530 wireless ZigBee module to build a simple system that can be remote controlled from a home page. Their conclusions were that the gateways provide some challenges and that the system did not provide sufficient processing power to run AI algorithms. Mao et al. (2010) present a controlled centralized home system that uses an ARM9 processor. Their works were targeted a centralized server concept with an Internet connection. However, none of these works deal with the issues of saving power, lower component form factor and provide sufficient processing power to support and run AI algorithms. Power consumption for S.O.s can be divided into three parts: Sensors, micro processor or controller, network interface. Power saving in the network interface is discussed in subsection 4.4.2.

Sensors as such do normally not use much power unless they need some driving electronics and wireless transceivers. However, even then these electronics can be duty-cycled, i.e., using an appropriate power sequence alternating between sleep and run. Using this method Lutz et al. (2010) have shown that the energy consumption can be lowered with a factor up to 10^6 . Regarding the microprocessor power consumption it can be exemplified by using a modern (2013) existing micro processor, which is the PIC18F46J50 from the company Microchip. It is able to save power by using different modes that offer different performances depending on its programming. In deep sleep mode it offers a current consumption of 13 nA. Wakeup from this mode can be done from an external event such as a ZigBee transceiver. In full run mode it uses 10 mA and offers approximately 10^6 instructions per second, which is more than sufficient to run the artificial network as explained in section 8.7. The form factor of components has decreased through the years, thus the ZigBee module NY2400SC (from the company A.N. Solutions) only consumes 3 cm^2 and the PIC18F46J50 consumes 1 cm^2 .

However, these numbers need some additional amount of area for the battery and other needed components.

As exemplified, today's electronic components are sufficiently mature to support an S.O. framework that offers: Transceiver functionality, low power consumption and high processing power. The challenge is to go beyond today's state-of-the-art, making these S.O.s context-aware, intelligent and able to communicate, and combining them into a distributed system for the future smart homes. They should be able to not only react to changes in the environment, but also perform AI-based reasoning to take into account the preferences of the users inhabiting the smart home. Actually, a lot of research is needed in this area.

4.5 NETWORK TOPOLOGIES

The smart home topology, i.e., the organization and layout of nodes can be done in many ways and are individual for smart homes. However, as discussed earlier at a heuristic level two basic types exist. One is this centralized and the other is the distributed approach (Sohraby et al., 2007).

The centralized network (e.g., the star topology) has a central node that receives all information from the sensor nodes by using the intermediate routing nodes to pass it on (Wang, 2010). The central node does all the processing and based on the results it controls the actuators, e.g., turning on the light. This topology reduces the complexity required by the nodes together with the network complexity. The central node must have a powerful processor to be able to do all the calculations in real time. A drawback is that the system is totally dependent on the central node and fails totally if that fails.

The distributed network (or decentralized network) is a type where each node can communicate with its neighbors (Li & Yu, 2011). However, in wireless networks some of the nodes may be too far from each other to communicate directly. This means that the distributed topology must provide multi-hop capabilities and the nodes must know the node address of their neighbors. Some systems add more complexity by providing ad-hoc networking capability where, e.g., new nodes can arrive and automatically be integrated into the existing network topology. Thus in the distributed topology, sensor nodes use routing nodes to forward the

messages until they reach their final destination, i.e., the final node. This means that a message may be routed by using different protocols, which is time consuming and resource demanding (Starsinic, 2010).

The challenges are to find the best combination of the basic topology types seen in the light of their ability to make inter-device communication and to communicate with systems outside its own context such as cloud services. Additionally, the topology of supporting devices with different communication capabilities is challenging.

4.5.1 ROUTING IN SMART HOMES

Routing in smart homes is a complex matter. It needs to combine many different device types into one common network. Additionally, its intelligent gateways must be able to make decisions based on device energy resources and processing power. Furthermore, routers are needed to connect this common network to the world outside its own context such as cloud based services and a portable user interface.

Many people associate SHNs with broadband streaming of multimedia services, but this not the purpose with these networks as discussed in subsection 4.2. However, multimedia and entertainment devices are connected to the Internet by cable or wireless technologies by using a parallel IP based network. Similarly, the SHN has Internet access in its gateway, so it is possible to route between these two networks. But, it should be noted that this connection can only be used for smart home events and actions, because the bandwidth in the SHNs are very limited (Subsection 4.2).

In the future SHNs will be an important new class of low bit rate communication (Li et al., 2011a), (Starsinic, 2010). As discussed, such a concept would require that the smart home owner is capable and willing to manage several local networks where each has its own gateway. An alternative approach is presented by Starsinic et al. (2010) in Figure 4.7 where they have integrated the different network protocol gateways into a centralized one. By using this approach the user only needs to interface with one central gateway to manage the smart home. The always powered gateway is able to scan for new devices and subnets autonomously and thereby enable plug and play. However, the disadvantage in this approach is complexity and the implicitly used centralized approach.

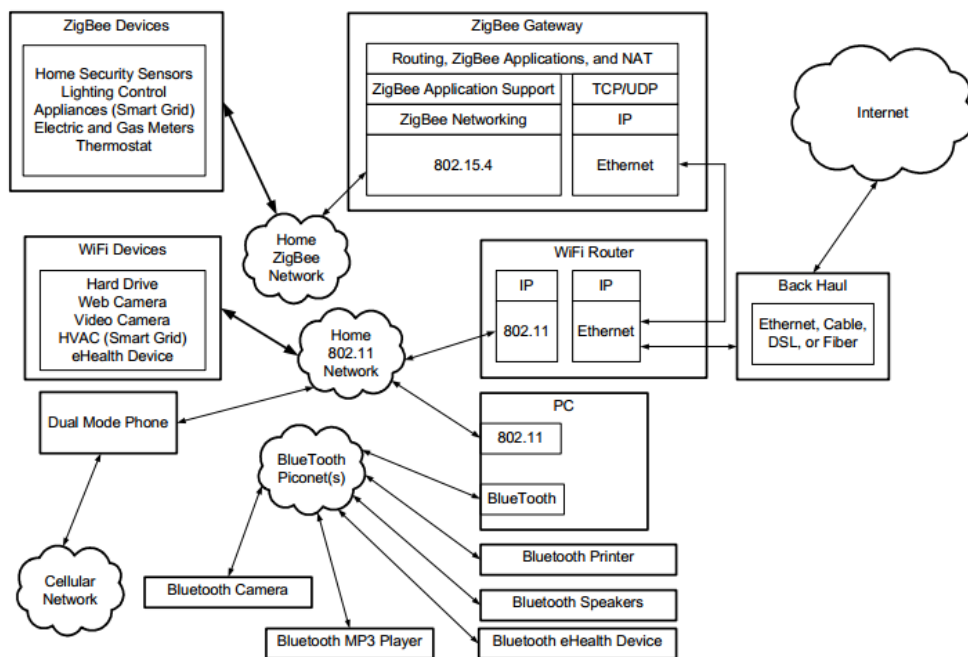


Figure 4.7. A converged smart home network (Starsinic, 2010).

A new (2013) useful concept for building distributed gateways is the Software Defined Radio (SDR) technology. It is still in its infancy, but it has a huge potential in gateway systems where it offers multi-protocol radios by using a fixed and reduced amount of hardware (Pucker & Kaiser, 2013). SDRs are able to bridge between two subnets running different standards and using different frequency bands. Their architecture handles and simplifies network upgrades and protocol updates could be supported via simple software downloads (Tribble, 2008), (Subsection 3.5.2).

The use of combined gateways versus isolated network approaches offers advantages and disadvantages that need clarification. An additional challenge in smart home routing is to find an advantageous level of integration between an SHN, an entertainment network and gateways. In the future, SDR's are expected to be a useful part in both smart home routing and networking.

4.5.2 FACTORS AFFECTING THE NETWORK CHOICE

In general, a lot of considerations are needed to guide network choices for smart homes. Many factors are important; however, for this work at least three factors need attention, they are:

Range, bit-rate and energy consumption (Sundmaecker et al., 2010), (Casilari et al., 2010). Actually, these are interdependent and all end up in battery power consumption and network load.

Range in wired networks is limited by the connecting cables, whereas in wireless networks the range is limited by interferences and power consumption. This connection can be seen by looking into the loss factor in a wireless system, which is the free space loss⁶. This loss is the most dominant factor in a link budget. To overcome this loss the wireless network devices need transmit power. Similarly, the bit-rate in the channel between the transmitter and the receiver is limited by the Shannon channel capacity formula⁷. Moreover, this means that to increase the reliability of the receive information more transmit power is needed, which in turn requires more (battery) power.

The energy consumption in the network transceivers is related to the individual transmitter and receiver contributions. The transmitter consumes a lot of power when it transmits but almost nothing else; whereas the receiver uses power most of the time because it listens for incoming messages. Other factors that influence the network choices are (Gershenfeld et al., 2004), (Starsinic, 2010):

- cost of components
- component size, it matters because they often need to be hidden
- scalability, i.e., how many devices can coexist, this also influences component power consumption and delay
- standardization, which reduces the amount of proprietary networks
- reliability in wired and wireless networks have different levels; and finally
- privacy together with security.

⁶ It is expressed as: $L = \log_{10}\left(\frac{\lambda}{4\pi D}\right)^2$ where λ is wavelength and D is distance between the receiver and the transmitter.

⁷ $\frac{C}{B} = \log_2\left(1 + \frac{S}{N}\right)$. Channel capacity (C), received signal power (S), noise power (N), bandwidth (B).

Thus, when an SHN topology, components and concepts need to be chosen many factors need to be considered for obtaining the best compromises. In addition, these factors need attention when the next generation SHNs are considered.

4.5.3 WIRED SMART HOME NETWORKS

Wired SHNs are useable at places where it is possible to hide the wired installation (Kaila & Mikkonen, 2008), e.g., in closets and behind shelves, etc. Because they are powered by the mains they are able to handle more complex and intensive processing tasks. However, most of these wired systems suffer from poor bit-rates that limit their usability (Chen et al., 2011).

In general, many embedded electronic systems use and provide network communication capabilities. Examples of such commonly seen networks are IIC, USB and RS-485. They all offer multi-drop connectivity, i.e., point-to-point or point-to-multi-point connectivity. Nowadays USB is by far the most used protocol. It is the standard interface for PC's and many small-size and low-cost embedded microcontroller devices also offer USB connectivity. However, USB is designed for high speed and not for power saving. In the context of smart homes this standard should only be used in connection with devices that are powered by the mains grid.

Existing cabling in smart homes can also be used for communication. An example is power-line communication that uses the mains cables to communicate without the need of any other infrastructure element. Different protocols are available e.g., INSTEON, EHS/KNX, UPB, X10 and CEBUS. INSTEON is used for connecting light switches, thermostats, motion sensors, etc. X10 is much alike, but it has become very popular with millions of users worldwide due to its simplicity. Other protocols like KNX is a standardized OSI-based network communication protocol (EN 50090, ISO/IEC 14543, CEN EN 13321-1). It is able to use different physical media like twisted pair, power-line, radio, infrared and Ethernet connections. Common for all these wired systems is that their communication takes place over the mains grid. They all offer a low data rate channel (normally a few hundreds of bits per second) over long distances. However, faster bitrates can be achieved by limiting the distance that the signal has to travel.

Name	Wired/ Wireless	Data rate	Standard
INSTEON	Both	180 bit/s	Smartlab Inc. Patented
KNX	Both	1200 bit/s	EN 50090, ISO/IEC 14543, CEN EN 13321-1
UPB	Wired	240 bit/s	PCS Powerline Systems of Northridge, California
X10	Both	20 bit/s.	Pico Electronics (Industry standard)
CEBUS	Both	8 kbit/s	EIA-600

Table 4.2. Overview of wired communication standards.

Overview of some of the most popular wired protocols are given in Table 4.2. An example of how these wired protocols can be combined with wireless protocols to form an SHN is provided by the work of Jin et al. (2011). They discuss how to combine wired and wireless networks by the use of mainstream technologies.

Using wired network devices as framework for weak resource devices is beneficial because it is able to provide a mains-powered core, which can supply the weak resource devices with power. However, the low bitrates provided by these wired networks require some focus, i.e., it is a challenging research area.

4.5.4 WIRELESS SMART HOME NETWORKS

Most SHNs are founded in the world of WSN's (Pensas & Vanhala, 2010), (Sohraby et al., 2007). That is, the smart home topology is strongly related to the WSN that often consist of a collection of inexpensive and compact-size computational nodes. These nodes are able to sense and report their contextual conditions, i.e., they measure parameters that can be forwarded to a central point for appropriate processing.

Mainly two WSN categories are used (Sohraby et al., 2007):

- Category 1 WSNs: Almost invariably mesh-based systems with multi-hop radio connectivity among or between WNs, utilizing dynamic routing in both the wireless and wire line portions of the network.
- Category 2 WSNs: Point-to-point or multipoint-to-point (star-based) systems generally with single-hop radio connectivity to WNs, utilizing static routing over the wireless

network; typically, there will be only one route from the WNs to the companion terrestrial / wire line forwarding node (WNs are pendent nodes).

Coupling the WSN topology to the SHN topology it provides a more complete picture of how SHNs work. Starting with the Category 2 WSN its most commonly used topology is the star network structure. As discussed earlier, this structure has been used to support early building information and home automation systems. However, it still has a potential in the smart home area, especially because it provides a simple network structure and an uncomplicated behavior. Looking into the Category 1 WSN it contains the structure and dynamic needed by future smart homes. Thus, it offers a structure with small islands (clusters) that contain chains of end-nodes (devices) linked to a dynamic route wireless router. This cluster based structure fits well with a hierarchical based smart home structure and a distributed agent approach.

The challenge is to use a combination of these established WSN technologies to achieve a flexible wireless network that is able to support nodes with different processing capabilities, different power supply types and be able to support advantageous clustering topologies.

4.5.5 ALGORITHMS USED IN SMART HOME NETWORKS

Node communication provides a mechanism for transporting and exchanging smart home sensor and device information. It is carried out by the backbone network and it supports different modes. One mode supports direct node access for transporting events, actions and identity information between the sensor nodes and to the outer world. Another mode supports broadcasts when multi devices need the same information.

Different established WSN principles are available for node communication (Sohraby et al., 2007). These will be discussed, in the light of the smart home application area, for clarifying their potential use.

4.5.5.1 Flooding Algorithm

In this algorithm each node simply transmits copies of a received message to its neighbors. Such a concept is simple, scalable and it requires low maintenance. However, if there are ring structures in the network, messages can circulate forever. To avoid this each message is equipped with a time to live counter. Another disadvantage is its susceptibility to traffic

implosion, i.e., one message replicates into many messages and thereby raises the network load. In addition, the same message can be routed twice by a given node. Finally, this strategy also is resource blind, i.e., battery powered nodes are flooded by the same amount of packets as the mains powered nodes.

The flooding algorithm is not commonly used in SHNs, but it is used in machine-to-machine (M2M) communication because of its simplicity (Subsection 3.5.2). Furthermore, it is expected to be one of the communication schemes used in the world of Internet-of-things (IoT) (Chen et al., 2011). From a smart home context it is reasonable to expect that this will be used when the M2M and IoT areas integrate into the smart home context in the future. Jin et al. (2011) have found that the flooding algorithm is superior to dedicated routing algorithms in the case of broadcasting. Their investigations have focused on power consumption, network load and average latency.

4.5.5.2 Gossiping Algorithm

This is a variant of the flooding algorithm. Each node forwards the message randomly to each of its neighbors until it arrives at its destination or the hop counter is exceeded. The disadvantage is that message latency (delay) can be long (Hedetniemi & Liestman, 1988). This algorithm is useable for single node broadcasting to many smart home nodes. However, if the information only needs to be broadcasted to a subset of these nodes common flooding is more useable.

4.5.5.3 SPIN Algorithm – Sensor Protocols for Information via Negotiation (SPIN)

The main objective is to disseminate a message from one sensor node to all the other nodes (Kulik et al., 2002). By using descriptive metadata the SPIN algorithm notifies all other nodes about is available data type and content. Nodes interested in this type of information can then subscribe to it (Park & Corson, 1997). The disadvantages are node complexity and that data are not always delivered because intermediate nodes may not be interested and therefore drop the advertising message. In smart home context this algorithm is useable to distribute information to nodes that have a particular interest to it. However, it has not been used much

in the smart home research area. Xu et al. (2013) have used it in a modified form to reduce the message flooding in the flooding algorithm.

4.5.5.4 LEACH Algorithm – Low Energy Adaptive Clustering Hierarchy (LEACH)

This algorithm divides the sensor nodes into clusters where each cluster has a cluster head that handles the nodes communication needs. Thus, the cluster head uses a Time Division Multiple Access (TDMA) approach to poll (using a round-robin fashion) each node for messages and collect these into a packet that is transmits to the data sink. Despite these benefits LEACH suffers from a high complexity level (Heinzelman et al., 1999). In the context of SHNs Azim et al. (2009) have developed a different version named Hybrid-LEACH. It has considerably lower power consumption compared to the original LEACH algorithm.

Some smart homes state-of-the-art network algorithms use a subset or a mix of the discussed category 1 and category 2 network models in combination with variants of the presented WSN routing strategy and algorithms. Xu et al. (2013) present their state-of-the-art work, which covers a smart home wireless network. Their work is based on the fact that some nodes use batteries whereas others have a fixed power supply from the mains grid. The mains grid nodes are then assigned the task of being backbone nodes for the data routing whereas the battery-powered nodes only transmit data relevant to themselves. Such a concept ensures that the battery powered nodes only use battery power for transmitting communication relevant to them, and only use power for receiving most of the time. It should be noted that transmitting messages uses much more power than receiving them. However, as already discussed, the receiver is always listening and thereby using battery power over a much longer time than transmitting does.

Xu et al. (2013) modified the presented LEACH routing algorithm. Thus, this algorithm was modified into an algorithm named LEACH-Pi, which restricts the choice of the cluster head to be one of the mains powered nodes. Similarly, the SPIN algorithm was modified into an algorithm named SPIN-Pi, which pairs the battery powered nodes with a mains powered counterpart that takes care of the data subscription. The latter node exchanges data with the paired battery nodes. The SPIN-Pi algorithm is mainly for distributing data between nodes, whereas the LEACH-Pi algorithm is useable for point-to-point communication. The

disadvantage of these schemes is their spatial cluster head placement selection and blind spots (Azim & Islam, 2009). A similar work has been done by Takizawa et al. (2012) that combines mains powered nodes with battery powered ones in a 6LoWPAN scheme.

Summing up, all the discussed node communication algorithms offer different services, capabilities and performances. Smart home state-of-the-art technologies that extend the existing node communication technologies are discussed. It enables battery and mains powered nodes to be part of an SHN in an advantageous way. However, using this technology still leaves problems such as node clustering, allocation of cluster masters and node power consumption unanswered.

4.5.6 WIRELESS NETWORK TYPES

As previously discussed, many well-defined and standardized protocols exist in the WSN area. Most of these are used in smart homes often in a mixed fashion because no single standard exists, manufacturers use their own favorite, and a combination of the different technologies is needed.

An overview of the wireless network types that are commonly used in smart homes is provided in Table 4.3 (Gomez & Paradells, 2010), (Starsinic, 2010). For being able to compare these some relevant areas have been selected, they are: Frequency, data rate, device range, device security, and the used modulation spreading techniques (Gomez & Paradells, 2010). The frequency parameter is important because, if the wireless network type supports the ISM band, users can deploy these devices without type approval or the need of some special license, in most countries. Parameters such as high data rate and device range are normally desired, however, increasing these parameters normally also means that more power is needed (Subsection 4.5.2). This is a paradigm for battery power devices, i.e., a compromise must be found. Modulation and spreading techniques are important for data rate, bit error probability, channel robustness, noise resilience, and are vital parameters in wireless multi-node systems based on multiplexing technologies.

Protocol	Frequency (MHz)	Data-rate (kb/s)	Security	Range (m)	Modulation /Spreading technique	Specification publicly available
ZigBee	868/915/2400	20/40/250	Yes	10-100	QAM/DSSS	Yes
6LoWPAN	868/915/2400	20/40/250	Yes	30	QAM/DSSS	Yes
Z-Wave	868/915/2400	10/40/200	Yes	30	BFSK/No	No
INSTEON	904	38.4	Yes	45	FSK/No	No
Wavenis	433/868/915	5/19/100	Yes	200	GFSK/FHSS	No
BT-low energy	2400	270	Yes	50	GFSK/FHSS	Yes

Table 4.3. The most commonly used wireless network types in smart homes (QAM is quadrature amplitude modulation, FHSS is frequency hopping spread spectrum, FSK is frequency shift keying, BFSK is binary FSK, GFSK means Gaussian FSK).

One of the most usable network technologies for smart homes is ZigBee (Yan & Dan, 2010), (Ma et al., 2012), (Trevenor, 2012). It uses the ISM band, offers scalable bitrates, employs encryption, has low power consumption, it provides scalable range, and it offers a flexible network topology. The ZigBee technology is used later in this work, why it is presented at a detailed technical level in the following. In addition, the 6LoWPAN technology is presented because it shares many similarities with ZigBee (Lu & Wu, 2011). Alternative technologies such as Z-Wave, INSTEON, and Wavenis have been used too, but they have not achieved the same popularity as ZigBee in the smart home context (Gomez & Paradells, 2010).

ZigBee is a WSN technology that has been developed by the ZigBee Alliance for low data rate and short range applications. This means it is well suited for the smart home area. The ZigBee protocol stack is composed of four main layers. Layer 1 and 2 are defined by the IEEE 802.15.4 standard whereas the other layers are defined by the ZigBee specification. Actually, the IEEE 802.15.4 standard provides a very efficient communication channel in small WSN and smart homes that are small battery powered (Ma et al., 2012). This standard is able to form the different network topologies needed in smart homes. ZigBee is capable of working in the ISM band that is regulated by the ITU-R in 5.138, 5.150, and 5.280 radio regulations. This worldwide regulation means that each WSN node device in the smart home does not need to be type approved.

ZigBee uses DSSS based modulation scheme, which is noise resistant and implicitly adds encryption. This is useful in smart homes where the noise level from WiFi and other wireless systems can be high. ZigBee offers data rates from 20 kbit/s to 250 kbit/s which is sufficient for most smart home applications, except for entertainment systems. However, most of these systems use WiFi today. Basically ZigBee offers two channel access methods one with beacon and one without. This is very useful in smart homes where both TDMA synchronous and asynchronous devices could be used.

To support application development ZigBee defines two profiles. The first one is the automated public application profile, which defines descriptions, commands, attributes, and other practices for applications in the application area of lighting, HVAC, window shades, and security. The second profile is the Smart Energy Profile, which focuses on energy demand in response and load management applications.

The 6LoWPAN is an IP based technology (Dunkels, 2008) which is defined by the 6LoWPAN working group under IETF. It uses a layered model where layer 1 and 2 are defined by IEEE 802.15.4. On top of these layers are a 6LoWPAN adaptation layer, followed by the common Internet layers such as IPv6, TCP/UDP, and HTTP/TFTP. Offering Internet comparability on the top layers means that format conversion is not needed in the routers and gateways, which saves resources in these. However, the IPv6 datagram does not fit 6LoWPANs buffer size, meaning processing power is needed to do real time data manipulation (Hui, 2008).

The standard IEEE 802.15.4 used in the 6LoWPAN two lower layers is also used in ZigBee. This provides the possibility to use these standards interchangeably seen from a network topology point of view. A detailed overview is presented in Table 4.3. Thus, some of the large manufacturers such as NXP (Philips), Atmel, and Texas Instruments provide integrated circuits that support both ZigBee and 6LoWPAN in a single integrated circuit. In addition, some researchers claim this standard will be the future “backbone” in the IoT (Lu & Wu, 2011).

4.6 SUMMARY

This chapter explored the research frontiers in the relevant research areas. Firstly, the distributed system challenges were discussed. Secondly, the research was focused to the challenge of handling distributed devices from different manufacturers in the light of IoT without any available standard. Thirdly, the challenges of wired and wireless data exchange mechanisms and functions needed in an SHN were discussed. Included were network topologies and technologies which were researched in the context of power consumption, power sleeping, interference, and protocols. Finally, the sensor types and their challenges such as network load, cost, and power consumption were explored. In addition, the research areas device-power-consumption and transmit-only sensors were explored to find their key challenges. In this context the challenges of S.O.s were explored in terms of context-awareness, intelligent behavior, communication capabilities, and their ability to be combined into a distributed system usable for future smart homes.

5 ARTIFICIAL INTELLIGENCE: ALGORITHMS AND MODELS

In chapter 4 a distributed conceptual framework and its included problem areas were discussed in a smart home context. Furthermore, it has been discussed that a smart home implicitly requires Artificial Intelligence (AI) to be context aware and be able to provide intelligent services to its user. The background for the AI part is covered in this chapter.

Looking into the second research question in the view of smart home AI systems, it can be divided into some sub-problem areas that are discussed in the following. For convenience the second research question is restated:

Question 2: *How should the AI be distributed to comply with the smart home system architecture?*

This research question relates to the research area of distributing *AI* in a *smart home system architecture*. This is discussed in terms of AI definitions, AI applications and the AI algorithms in sections 5.1 and 5.2. Additionally, different AI models need to be explored and this is found in sections 5.3 and 5.4. Moreover, focusing on AI models that are based on currently existing technologies and devices limits the candidate field (Section 1.3). Thus, researchers have pointed out that the probabilistic based family provides some good candidates for smart home usage (Section 5.5). These selected models are presented and discussed in sections 5.5, 5.6 and 5.7.

5.1 AI DEFINITIONS AND APPLICATIONS

A working definition and clearness of the AI term are provided in subsection 5.1.1. Some examples of the AI application areas are provided in subsection 5.1.2.

5.1.1 ARTIFICIAL INTELLIGENCE DEFINITIONS

Initially defining the term *artificial intelligence* is beneficial, because no precise unique definition exists today. Hence, by using a simple description it can be stated that artificial

Intelligence (AI) is the intelligence of machines or software. This description covers elements such as knowledge management as well as reasoning and learning models; and it includes methods such as mathematical optimization, logic and probability. Elaborating over the missing definition Luger et al. (2008) attempted to define it this way:

AI may be defined as the branch of computer science that is concerned with the automation of intelligent behaviour (Luger, 2008).

From this definition, AI is a discipline of computer science and it is based on a theoretical framework that covers this field. Such a definition highlights and expresses the lack of knowledge that characterises this field. Thus, AI researchers often build particular intelligent artefacts that are able to solve one isolated problem. For example, the Bayesian networks are often used to visualize patterns and the Markov models are often used to recognize speech.

Another attempt to defined AI is made by Pole et al. (2010) who claim that:

AI is the field that studies the synthesis and analysis of computational agents that act intelligently (Pole & Mackworth, 2010).

Their definition covers the term *agents*, which they define as something that acts intelligently in an environment. A refined definition of this term is a *computational agent* that makes decisions which can be expressed in computational terms. These decisions can be broken into small primitive operations that are implementable on physical devices. Describing an agent from a different angle provides clearness about its concept. An agent acts in environments where it is possible to use history and knowledge from prior observations. This means that an agent is able to learn from the past and make use of this learning to predict similar learned situations in the future.

Analysing agents from a more general perspective reveals that agent designers concerns about: modularity, modelling the world, uncertainty in actions, goals of preference, how to learn, and how to limit computational resources. In addition, this discussed agent-based concept is in close agreement with the definition attempt provided by Russell et al. (2009).

These AI definitions are useful in smart homes where elements to be predicted are of singular as well as complex nature. Singular systems rely on local simple event based intelligent artefacts to make decisions such as *turning on the coffee-brewer*. Complex systems needs to use the past event based history together with the present events to predict complex actions such as *the user goes to sleep*.

However, the challenge is to map these general definitions into usable ones that are optimal for smart home context, that are standardized and that are implementable with the available technology (Alam et al., 2012).

5.1.2 ARTIFICIAL INTELLIGENCE APPLICATION AREA

The application area for AI covers a wide range of applications such as toys, scientific research tools, medical diagnosis and robot control. In addition, many of today's services are based on embedded AI, examples are self navigating vacuum cleaners, recommender engines, gaming engines, cars gearboxes, speech recognition, and industrial robots. Liu et al. (2010) use AI to build a mobile learning system which is able to vary the problem and exercise difficulty level, so it matches the level of the student. This system varies the explanation and learning content based on the student's weak knowledge areas and change its teaching strategies. Chen & Liu et al. (2010) used an AI system to optimize the limited water resources from a river basin. They solved a water resource conflict and improved its economic efficiency greatly. Kondo et al. (2011) employed AI to improve the diagnostics of lung cancer. They used AI in the medical image reorganisation process with a high success rate. Thus, the application area of AI is widespread and often it solves the problems in a beneficial manner.

The future challenges and perspective of AI are difficult to predict, however, Turner (2013) has suggested some possible candidate areas:

- Robotics will be able to think for themselves and react to changing conditions by using AI.
- Weather forecasts will be based on AI and thereby save lives in case of big storms.
- Dangerous and monotonous tasks will be performed by AI systems. Today, robotics already does this to some extent.

- Swarm robotics that will be able to save the planet in form of cleaning pollution.
- Intelligent transportation systems and cars will be able to run by themselves i.e., the human factor will be eliminated. This will increase safety, because AI reacts much more precise and faster than humans.
- Space exploration with AI robots. NASA already works with this.
- Protect your finances. AI will be able to spot fraud in the change of credit card use.
- Staying safe. AI based security are able to distinguish unknown persons, etc.
- Virtual assistants will help with traditional secretary's tasks and as a medical assistant.

However, this is only a limited list of possibilities today's AI research is covering a huge area of applications and technologies. In the area of smart homes the future challenge is how to integrate AI based technologies that affect people's living and housing with smart home technologies.

5.2 AI AND PATTERN RECOGNITION

To concretize the broad term AI in the light of computational data learning the term Machine learning is used. The part of machine learning that assigns categories to input values is named pattern recognition. Thus, pattern recognition algorithms aims to classify input values into the most likely output values depending on statistical variation. As such, this is the backbone algorithms used in machine learning and AI. Thus, the term AI will be used synonymously with the term pattern recognition in this work.

From a high abstraction level pattern recognition methods can be divided into three model groups. These groups have been used in the smart home context, why it is informative to discuss their usability and usage areas.

Group 1: The temporal data mining model finds templates in the data arriving from the sensors. These templates can be used to detect recurring actions. One approach based on this method is presented by Jakkula et al. (2008) who use it for detecting abnormal behaviour patterns in smart homes for people with cognitive disabilities. A similar method is the temporal-pattern that automatically identifies repeating patterns in a sensor data stream. It looks for sensor events that are interdependent, so if event E1 occur event E2 can be expected

with some probability within time window T . This method has been used by Jakkula et al. (2007) to detect anomaly frequently-occurring events. The advantage is that these methods find recurrent patterns automatically without user feedback. However, this also means that it is unknown whether the detected patterns are of any use. These techniques are closer to action discovery rather than action detection.

Group 2: This group consists of the logic based methods. They build behaviour rules as a function of time that is hardcoded into the system. Using these rules it is possible to check if some specific behaviour takes place in a defined time window. This method has been used by Chen et al. (2009) in their adaptive scenario based reasoning (ASBR) system, (Section 5.2), (Subsection 6.1.1). The disadvantage in this method is the ambiguity that exists in which action that has been performed. I.e., it is not possible to get information on which action is the most likely one.

Group 3: This group is the temporal probabilistic models. They model sequential data, i.e., a sequence of sensor event observations. Based on these, the goal is to infer a matching sequence of hidden states that correspond to the performed actions. This is performed by searching the sequence of states that maximizes the probability of the actions given the sensor readings. Two model types exist. They are the generative and the discriminative probabilistic models. Generative models use the joint probability between sensor events and the action predicted why they can be used to generate (sample) data. Discriminative models are used for inference only and therefore define the conditional probability directly. Examples of these different models are the naïve Bayes model (NB), Hidden Markov Model (HMM) and the Conditional Random Field (CRF) (Cook, 2012), (Kasteren et al., 2008b), (Fang et al., 2012), (Kasteren et al., 2011). An advantage of these models is that efficient algorithms exist that perform inference in an optimal manner. The disadvantage is that annotated training data are needed to learn the model parameters.

A smart home challenge is to choose the right combination of models that offers the needed properties in terms of learn-ability, inference ability, and its use of implementation resources.

5.3 PROBABILISTIC MODELS IN SMART HOMES

Using AI in smart homes is challenging. Especially, because the recognition model that handles the activity monitoring includes some difficulties. It interprets the received sensor events and recognizes the actions performed by the user. This is a challenging task because sensor data are noisy and the related activity data start and end points are unknown (Cook, 2012).

A variety of smart home models exists. One popular group is the category of non-parametric supervised classification models that contain decision trees, neural networks, naïve Bayes classifier and the supporting vector machine. All these models estimate category labels based on supervised training. Another popular group is the supervised categorical sequence labelling models that comprises conditional random field, hidden Markov models and variants of the hidden Markov models. They are useful to predict a sequence of categorical labels. I.e., they are able to use time correlated information (Chua et al., 2011), (Cook, 2012).

To handle the smart home challenges probabilistic models are beneficial because they perform well in this context and they are the first choice for most researchers (Kasteren et al., 2010), (Alam et al., 2012). Hence, probabilistic models offer robustness in the presence of noise, they are able to handle sequential data, and they generate probability distribution over the class label (Cook, 2012), (Kasteren et al., 2008a). In addition, efficient algorithms have been developed over the years by independent researchers.

In the group of probabilistic models the most commonly-used ones are the naïve Bayes model, the hidden Markov model, and the conditional random field model. The naïve Bayes model handles the sensors independently without temporal information (Rashidi et al., 2011). This means that it is simple to implement, but lack the time dimension and the interdependencies in a sequence of actions. In contrast, the hidden Markov model adds temporal information by including a probability element that describes transitions between the actions (Kasteren et al., 2010). Thus, the hidden Markov model has temporal interdependencies, but it suffers from the complexity of the inference methods.

These models are generative probabilistic, i.e., they require a model describing the dependencies between the observed sensor data and the actions. However, they have problems

with modelling long-term dependencies. Another classifier named Conditional Random Fields (CRF) can handle this. It is based on a discriminative probabilistic model (Kasteren et al., 2008b). But it suffers from high computational costs in training.

A challenge in smart home context is that these models require a large amount of processing, memory and power resources. These resources are not available on the distributed small cheap S.O.s derived in this work (Chapter 4). So at this level the challenge is to transform these AI algorithms into a form that enables distribution to different processing platforms. Depending on its available resources it offers a different level of prediction. Another challenge is that these algorithms have not been used very much in real-time learning.

5.4 NON PROBABILISTIC MODELS IN SMART HOMES

A variety of non probabilistic smart homes models exists. Nevertheless they have not been used much in a distributed context for reasons such as: they are resource demanding, they use large databases, they do not scale well, and they have difficulties with handling noisy data (Chen et al., 2010), (Cook, 2012). However, in many other application areas they provide good results. Thus, they will have substantial impact on future areas such as: development of our society, on ambient assisted living applications, green technologies and comfort related aspects.

Artificial neural networks are some of the non-probabilistic models that researchers have looked into. These models are able to model activities of the user's daily living with an acceptable performance (Fang & He, 2012), (Zheng et al., 2008). However, neural networks do not always reach the same solution for the same input; they use large amounts of resources; and they need a large amount of training data to obtain a reasonable efficiency (Chen et al., 2010).

Other non-probabilistic models that have been tried in smart homes without establishing themselves in the area are:

- Spatial-temporal reasoning that models a spatially pattern and understands how pieces fit together into that space (Jakkula et al., 2007).

- Temporal granularity uses temporal specification of data to qualify them (Nazerfard et al., 2010).
- Causal reasoning is based on the ability to determine the cause given some circumstances in the environment (Song et al., 2011).
- Planning where it is observed if the user follows a certain planned activity (Dominici et al., 2010).
- Case-based reasoning uses the principle of solving a new problem by using the knowledge of a similar past problem (Cheng et al., 2009).
- Decision trees search trees for a correlation to previous activities (Rashidi et al., 2011).

These non-parametric models offer knowledge in the field of spatial-temporal reasoning and temporal granularity. These fields are important in a smart home context because they model sequences and temporal information that are related to the user action sequences and duration.

5.5 DISCUSSION OF SELECTED SMART HOME MODELS

This section discusses the sensor and actuator principles and their shaping of data types. Following this is a presentation and discussion of the most relevant probabilistic classifiers, i.e., the naïve Bayes, the hidden Markov model, and the Conditional Random Field. Finally, the use of these in a smart home context is discussed.

5.5.1 SENSOR EVENTS AND ACTIONS, A NOTATION

As discussed, smart homes are equipped with many sensors of different types. However, for this work a simple sensor framework is needed, i.e., the sensors are simple contact switches which provide open and close states. These switches can be positioned at doors, inside drawers, in closets but also exist as pressure mats, infrared sensors and water sensors. Common for all of them is that they provide a binary result (Section 4.4). This means that some problems exist in form of finding the proper start and stop time for the sensor events, detecting a specific action based on the switch status and finally eliminating noise from falsely activated switches that could also emit false signals. Another problem is the sequence of the predicted outcomes, i.e., actions. A user can do actions in different ways from one day to the other and thereby produce different sequences. An example is that the user enters the kitchen, turns on the light, opens the fridge door and sits down to eat. Another day this

sequence could be changed to the user enters the kitchen, opens the fridge door, turns on the light and sits down to eat. These issues make pattern recognition of the smart home actions a very challenging task (Kasteren et al., 2010).

To transform these challenges into a manageable form a mathematical notation is needed. Consequently, a notation that models the sequence of events and actions into a mathematical form is illustrated in Figure 5.1.

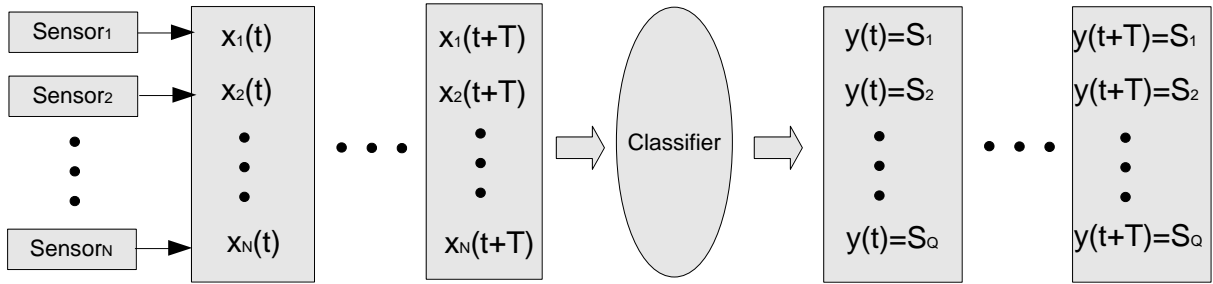


Figure 5.1. The mathematical notation used in this thesis. Leftmost is the sensors which input events $X_1..X_N$ in time window $t..t+T$ into a buffer. These buffered events are fed to the classifier algorithm that predicts state information S_x for the classifier output y in the time window $t..t+T$.

As illustrated leftmost in Figure 5.1 the sensor input events are placed in a time buffer which can be expressed by defining a binary observation vector (i.e., the sensor data):

$$\bar{x}_t = \{x_t^1, x_t^2, \dots, x_t^N\} \quad (5.1)$$

where x_t^i is the event i arriving at time instance t , $i \in \{1 \dots N\}$ and N is the number of sensors. These data are fed to the classifier process located in the center of Figure 5.1. Its output y_t can be in one particular state at a time. Thus, the predicted action y_t in time instance t has Q possible state values $S_0 \dots S_Q$, which in this thesis are written in a short form that leaves out the S . Thus, y_t is denoted:

$$y_t \in \{1, \dots, Q\} \quad (5.2)$$

Often in smart home applications Q is binary, i.e., the state values can be either zero or one (alternatively true and false). Hence, to describe if an action such as eating is ongoing or the light is on only two states are needed, i.e., true or false.

From a time perspective the sensor data arrive at different times. Thus, the time buffer are a matrix, i.e., it needs space for N sensor data (the columns) sampled in T instances (the rows) in Figure 5.1. A notation that expresses this contains one sensor vector (5.1) in T instances:

$$\hat{x}_{1:T} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_T\} \quad (5.3)$$

And for its corresponding output vector:

$$\hat{y}_{1:T} = \{y_1, y_2, \dots, y_T\} \quad (5.4)$$

5.5.2 NAÏVE BAYES PROBABILISTIC MODEL

As already discussed, the naïve Bayes classifier is one of the most important probabilistic models. It has been used by many researchers in smart home context and in this work (Rish, 2001), (Brgulja et al., 2010). In addition, it provides benefits in the form of offering a structure that allows it to be adapted to a simple sensor concept (Subsection 5.5.1), (Kasteren et al., 2008b).

Looking into the naïve Bayes equations a challenge is how to implement the observed probability distribution in agents, i.e., in small low power microcontrollers with very limited resources. This means that the mathematics must be simple, i.e., avoid anything else than adding and subtracting if this is possible. Furthermore the event types must be simple binary based types like Booleans and Characters. To highlight these problem areas and because the smart home usage is a highly specialized application area the theory covering this field is discussed in the following.

The smart home variant of the naïve Bayes classifier assumes that data are independent and identically distributed (i.i.d.) and that it does not take into account any temporal dependencies of data (Kasteren et al., 2011),(Rish, 2001). Often the i.i.d. assumption holds in smart homes if the sensors are positioned so that they do not overlap, i.e., they do not emit correlated

events. However, in this work the temporal assumption is too restrictive. This means that the correlations between the actions are lost. To be able to handle this, more advanced classifier types such as the hidden Markov model are needed.

The naïve Bayes model can be expressed as:

$$p(\widehat{y}_{1:T}, \widehat{x}_{1:T}) = \prod_{t=1}^T p(\widehat{x}_t | y_t) p(y_t) \quad (5.5)$$

This equation states that the observed distribution $p(\widehat{x}_t | y_t)$ represents the probability that y_t would generate the observation vector x_t . The term $p(y_t)$ is the prior probability. To be able to simplify (5.5) in terms of using it in this work, many researchers apply the Bayes assumption that sensor data are i.i.d. which means that the sensor data can be modelled separately. This reduces the calculation complexity from 2^N to simply N if there are N sensors. Using this assumption the observation distribution factorizes as:

$$p(\widehat{x}_t | y_t = i) = \prod_{n=1}^N p(x_t^n | y_t = i) \quad (5.6)$$

Equation (5.6) states that the conditional probability for each sensor in input vector x_t^n can be multiplied with each other independently of the other sensors; given its output state is i . In addition it is assumed that all priori probabilities are equal.

By assuming the sensor outputs are binary it is possible to model each sensor input by using a Bernoulli distribution (Bishop, 2006) with the probability μ given by:

$$p(x_t^m | y_\xi = i) = \begin{cases} \mu_{mi}, & \text{for } x_t^m = 1 \\ 1 - \mu_{mi}, & \text{for } x_t^m = 0 \end{cases} \quad (5.7)$$

A graphical model that represents the use of a naïve Bayes classifier in smart home context is given in Figure 5.2. Leftmost it illustrates a node model for the probabilistic outcomes y_t (the predicted action) based on the independent sensor input x_t^i . Rightmost picture illustrates an

implementation example of an application in the smart home context. The detected action *sleep* is based on the product of all the sensor inputs multiplied by their respective weights w_i .

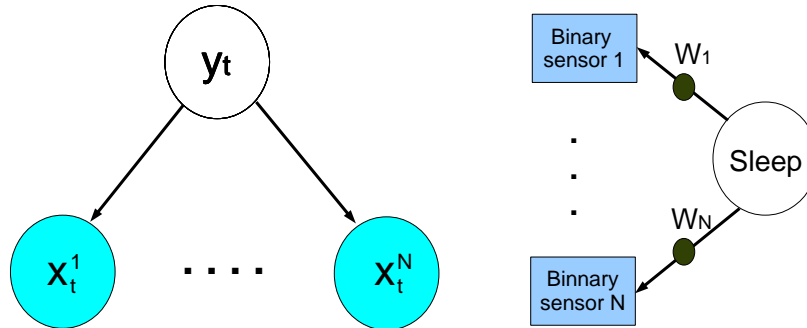


Figure 5.2. The naïve Bayes classifier. Leftmost is a node model for the probabilistic outcomes y_t (the predicted action) based on the independent sensor input x_t . Rightmost is an implementation example using it in a smart home application context.

Inference in the naïve Bayes model is simply performed by calculating the probability of all outcomes, i.e., states and choosing the one with highest probability as best estimate (Kasteren et al., 2008a). In the smart homes context it means that for some given sensor input values the classifier calculates the probabilities for each of the possible actions and chooses the one with the highest score.

Learning in a naïve Bayes model can be expressed in a closed form. The model learns the values of the parameters by using maximum likelihood and finding the joint maximum likelihood parameters. Because of the discussed i.i.d. assumption this task simplifies to finding the maximum likelihood parameters of each of the factors instead. Recognizing that the observation probability $p(x_t^n | y_t = i)$ is a Bernoulli distribution its maximum likelihood parameter can be expressed as (Bishop, 2006):

$$\mu_{ni} = \frac{\sum_Z x_z^n \delta(y_z, i)}{\sum_Z \delta(y_z, i)} \quad (5.8)$$

where the Dirac function (δ) returns one if action at time instant z equals the wanted action (i) or else it returns zero. Z is a vector of action event time instances used to estimate μ_{ni} . Simply

put, training is done by adding up the sensor events (represented by binary ones) present in the given action event window Δt .

The challenge is how to use and implement this model in a distributed environment. Researchers have used this model in centralized environments where the AI processing tasks are performed by a central server with plenty of resources. As discussed, a challenge is to modify and enable this model to run on a small, cheap microprocessor with very limited processing resources in real time.

5.5.3 HIDDEN MARKOV PROBABILISTIC MODEL

The hidden Markov probabilistic model is another very important and often used probabilistic smart home model. It is able to handle more complex predictions than the naïve Bayes model, because it removes the temporal restriction. Thus, this model is able to look back into the past. Even though, it is one of the simpler probabilistic models, it has some complexity challenges such as it needs a complex inference process. In addition, its smart home usage is a highly specialized application area and that is why its theory in this context is discussed in the following. An example of a Hidden Markov Model (HMM) node model is illustrated in Figure 5.3. This example contains M hidden states where it is possible to shift between them (the horizontal arrows). Shifting states are covered by the transition probabilities. The vertical arrows point to the N visible observation probabilities.

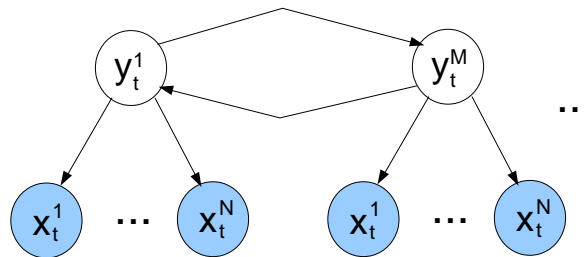


Figure 5.3. A hidden Markov model example. This example contains M hidden states where it is possible to shift between them (the horizontal arrows). Shifting states are covered by the transition probabilities. The vertical arrows point to the N visible observation probabilities.

The HMM uses the Markov assumptions that are:

- The first order Markov assumption (Rabiner, 1989) assumes that the hidden state y_t at time instance t depends only on the previous hidden state y_{t-1} .
- The observable state x_t at time instance t depends only on the hidden state y_t at that specific time slice.

Based on these assumptions the joint probability for the hidden Markov classifier factorizes to:

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}) = \prod_{t=1}^T p(\hat{x}_t | y_t) p(y_t | y_{t-1}) \quad (5.9)$$

For simplicity it is assumed that $p(y_1 | y_0) = p(y_1)$. It is noted that the observation probability in (5.9) is similar to the one used in the naïve Bayes section why these assumptions are used for the hidden Markov model too. Regarding the transition probability it represents the probability for changing state (i.e., action) from one state to the next. The state transitions can be modelled as a Q multinomial distribution, i.e., one for each state.

A graph and an example of an HMM are illustrated in Figure 5.4.

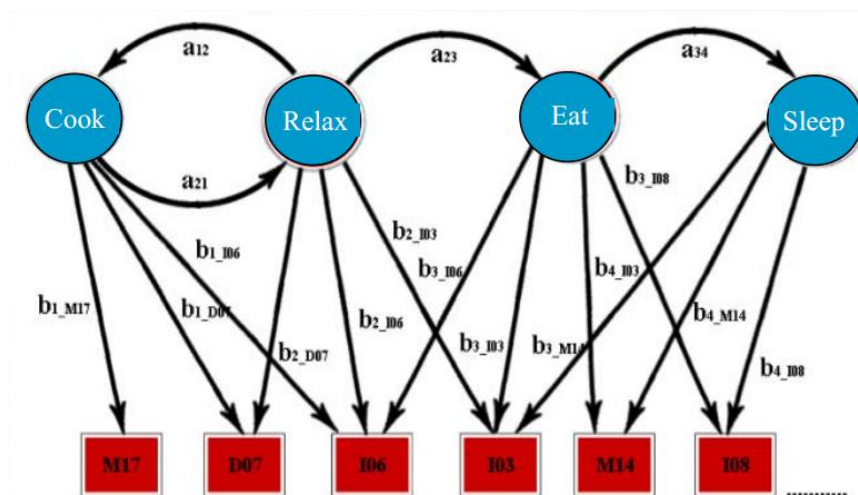


Figure 5.4. An example of using HMM in a smart home context (Cook, 2012).

The picture in Figure 5.4 illustrates an SHS HMM for an activity recognition task described in a paper by Cook et al. (2012). It uses four hidden states (activities) and a set of observable

nodes that correspond to possible sensor events. The a_{xx} values (x is any integer) represent the transition probabilities and the b_{xx} values represent emission probabilities.

The inference process needs to consider every possible state sequence, why it grows exponentially with the length of the sequence. The Viterbi algorithm is very efficient to find the best state sequence and thereby reduces the computational complexity to $O(TQ^2)$, where T is the total number of time windows and Q the number of states (Rabiner, 1989). However, implementing the Viterbi algorithm is a non trivial task.

Regarding the learning problem it can be divided into two problems. Firstly finding the observation probability the same close form solution (5.8) as described for the naïve Bayes model can be used. Secondly, the transition probability can be described as a multinomial distribution whose parameters can be calculated by using:

$$a_{ij} = \frac{\sum_z \delta(y_z, j) \delta(y_{z-1}, i)}{\sum_z \delta(y_{z-1}, j)} \quad (5.10)$$

Where the individual transition probabilities are denoted as $p(y_n = j | y_{n-1} = i) = a_{ij}$ and the Dirac function (δ) returns one if action at time instant z equals the wanted action (i or j) else it returns zero.

Like the naïve Bayes classifier the challenge is how to use and implement this model in a distributed environment. However, an important difference is that this model has a far more complex inference process often based on the Viterbi algorithm. Running this model in real time on a small, cheap microprocessor with very limited processing resources is neither useable nor likely to happen. It is not useable because it disrupts the degree of freedom that the concept of distributed systems gives, i.e., the S.O.s would be coupled together, which creates an unwanted dependency among these. However, the same is not true for the naïve Bayes classifier because it is based on non temporal principles, i.e., it does not depend on previous predictions. Running the hidden Markov model on limited resource processors will probably not happen because the processor resources will be exhausted and its battery lifetime will be short. In addition, the challenge is how to integrate this advanced classifier into a

distributed smart home context with distributed processing devices offering different resource types.

5.5.4 CONDITIONAL RANDOM FIELD PROBABILISTIC MODEL

The conditional random field classifier is a member of the important and often used probabilistic smart home model group. It is different from the naïve Bayes and the hidden Markov Model classifiers because it is based on a discriminative model. This means that it does not model the full joint probability, but it simply optimizes the conditional probability $p(\hat{y}_{1:T}|\hat{x}_{1:T})$. In addition, the conditional random field classifier is a general model-type. As a result, it also exists in different variants such as the linear-chain conditional random field that closely resembles the HMM in terms of structure. This model can be described as:

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}) = \frac{1}{Z(\hat{x}_{1:T})} \prod_{t=1}^T \exp \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \bar{x}_t) \quad (5.11)$$

Where K is the number of feature functions that are used to distribution parameterization. λ_k is a weight parameter and $f_k(y_t, y_{t-1}, \bar{x}_t)$ is a feature function that is non zero for one specific action (inclusive its predecessor) only. The $Z(\hat{x}_{1:T})$ function is a normalization term so that the distribution sums up to one, i.e., it obtains a probabilistic interpretation (Bishop, 2006). As noted the linear chain conditional random field is similar to the hidden Markov model except that its interpretation is non probabilistic, it is discriminative and the observation functions are not conditionally dependent of the states.

Regarding inference it can be performed by using the same principles as for the hidden Markov model. Learning optimizes the λ_k parameter by defining the log likelihood of (5.11) and finding its gradient. This gradient follows a convex function so the minimum is easily found, i.e., the local minimum is also the global minimum. Normally some methods based on Quasi-Newton iteration are used (Sutton & McCallum, 2007).

The linear chain conditional random field has the same graphical picture as the hidden Markov model (Figure 5.4) except the arrows that symbolize the conditional probability should be substituted by a line.

Regarding the challenges they are similar to the ones discussed for the HMM classifier.

5.6 APPLICATION OF PROBABILISTIC MODELS IN SMART HOMES

This section looks into the use of the naïve Bayes, the hidden Markov model, and the Conditional Random Field in different smart home application areas. Following this, these algorithms are compared with respect to the low-level prediction, i.e., they are connected directly to the sensors. Finally, examples of existing distributed SHS are presented.

5.6.1 THE USE OF NB, HMM AND CRF CLASSIFIERS

By looking into the literature covering the AI algorithms they are used in the smart home context to learn and detect human activities. Most papers use these by connecting them directly to sensors that emit measured values (Kasteren et al., 2008a), (Cook, 2012). Then each individual classifier is trained by running a large annotated dataset through it and updates its weights accordingly. Afterwards actions are predicted by the classifier whenever the sensors emit values that provide a pattern similar to the one used for the training.

An example of this procedure is the elderly care system presented by Kasteren et al. (2010), where they collect a huge amount of data by instruments in a home with a wireless sensor network that has many nodes which monitor the ongoing home activity. A 26 years old male lived and annotated actions in the home for a period of four weeks. These data were then processed by different classifiers by using all the collected information as training, but leaving out one day that was used for testing. Rashidi et al. (2011) introduced an automated approach to activity tracking that identifies frequent user activities by using classifiers. Thus, the models are trained to recognize these particular activities, and the resulting findings can be used to assess the functional well-being of smart environment residents. Similarly, Cook et al. (2012) designed the CASAS “smart home in a box” (Subsection 3.3.4). The learning takes place by assigning labels based on a set of “ground labels” to the latest sensor events.

The challenge of today’s (2013) centralized systems is how to train these classifiers in real time, i.e., real-time learning. As discussed all experiments performed start by training the AI system and when fully trained use some unknown dataset for testing their scores. Such an

approach is not very user friendly, because the user needs to produce a large annotated training set and then train the AI system. Thus, a concept of real-time learning is needed.

5.6.2 COMPARISON OF NB, HMM AND CRF CLASSIFIERS

To highlight the usefulness of the three classifier types Cook (2012) has made an experiment across eleven datasets collected from seven experimental smart homes, and these data are illustrated in Figure 5.5. The experiments are carried out in different smart homes (left column) and uses the NBC, HMM, and CRF classifiers (top row). Input to these classifiers comes from sensors spatially placed in the respective smart homes.

Dataset	NBC	HMM	CRF
Bosch1	92.91%	92.07%	85.09%
Bosch2	90.74%	89.61%	82.66%
Bosch3	88.81%	90.87%	90.36%
Cairo	82.79%	82.41%	68.07%
Kyoto1	78.38%	78.38%	97.30%
Kyoto2	63.98%	65.79%	66.20%
Kyoto3	77.50%	81.67%	87.33%
Kyoto4	63.27%	60.90%	58.41%
Milan	76.65%	77.44%	61.01%
Tulum1	59.05%	75.12%	79.45%
Tulum2	66.87%	57.83%	39.76%

Figure 5.5. Recognition accuracies (true positive) for the eleven data set used by Cook (2012). Rows are the smart homes used in the experiment and the columns are the classifiers used.

As seen the naïve Bayes classifier (Figure 5.5-NBC) performs very well compared to the other types that are the hidden Markov model (Figure 5.5-HMM) and the conditional random fields (Figure 5.5-CRF). It is also noted that the HMM and the CRF models are close to being equally good in many smart home types. However, Kasteren et al. (2008a) found that CRFs outperform HMMs in all the cases with respect to the time slice accuracy. However, the HMMs have the overall highest class accuracy. This result is a consequence of a difference in the parameter estimation processes used by these. The HMM uses a Bayesian probabilistic model for learning the actions. In contrast to this the CRF uses a single model for all the classes and learns by maximizing the conditional likelihood. This means the classes have to compete during the maximization. In datasets containing one dominant class, modelling everything this way might yield a different result.

Another work by Kasteren et al. (2010) uses the hidden Markov model to assist in care-giving for elderly people in their homes. They compare the performance of four real world datasets consisting of at least 10 days of data. Based on this they concluded that the CRFs are more sensitive to over-fitting on a dominant class than HMMs. Fang et al. (2012) did a lot of tests to clarify the differences between the recognition performances as a function of feature length. They found that the NBC, HMM and the CRF algorithms have a strong relationship with the dataset features that have been utilized. Kasteren et al. (2008b) support this finding. Conclusively a lot of historical and present activities have taken place in using these classifiers in smart homes. Other classifier types have been presented in the literature, but none of these have achieved the same popularity as the three just discussed ones (Kasteren et al., 2008b), (Cook, 2012).

As discussed, a lot of parameters need consideration in the choice of the most advantageous distributed AI algorithms. Nonetheless, as noted from the presentation and discussion the possibilities of a distributed system are limited.

5.7 DISTRIBUTED INTELLIGENCE IN SMART HOMES

Distributed intelligence in smart homes has been a very limited research area throughout the last decades. Most researchers have looked into using high-level agents and to some extent assign AI to these. One of the earliest works in this field is the *MavHome* project (Cook et al., 2003). Cook et al. (2003) divided a smart home into four high-level agent layers that handle the AI behavior. Mocanu et al. (2013) discuss an SHS with high-level AI agents for elderly people to stay longer in their homes. Their system uses agents at a high-level, i.e., emergency agent, home center agent, etc. In the "ThinkHome" project Reinisch et al. (2010) use agents that controls logic functions at a high-level (Subsection 3.3.4). The same pattern can be found in the work presented by Choi et al. (2005).

Changing the focus to low-level agents, i.e., the S.O. definition and terms (Subsection 4.4.5) used in this work no research in distributed intelligence has been performed. This is because:

- Distributed AI processing requires small, cheap and low power microcontrollers to perform the processing and they did not exist (Radi et al., 2011), (Hannon & Brunell, 2005).
- The algorithms need to be reduced in complexity to a level that fits the embedded processing resource (Basu et al., 2013).
- These algorithms must be able to do complex prediction, i.e., use temporal information (Basu et al., 2013).
- Battery technology was not sufficient to drive small embedded devices (Boegel, 2012).
- The area of WSN interfaces was not mature, i.e., poor battery performance and no allowance for transmit-only sensors (Zhao et al., 2013). In addition, low power integrated circuit especially in the energy harvesting is coming today (2013) and (Boegel, 2012).

However, the world of Internet of Things is an active research areas and it will probably add value to the smart home area in the future (Bandyopadhyay & Sen, 2011), (Zhang et al., 2012).

5.8 SUMMARY

This chapter provided an AI definition which comprises automation of intelligent behaviour and computational agents. In addition some guidelines for integrating an AI framework into the system architecture were provided. To comprehend the AI research field its different models were discussed in the light of properties such as learn-ability, inference ability, and its use of implementation resources.

The challenges of using AI in smart homes were discussed with focus on the variety of different ability models that exists, the required training in advance, and the sensor data that are minimalistic and noisy. Based on this discussion the most used probabilistic models were presented and discussed with respect to their suitability for application in smart homes. These models are the NB, HMM and the CRF.

6 CENTRALIZED AND AGENT-BASED MODELS

This chapter analyses results from similar research and builds two models that are used to generalize information and to base conclusions on. This analysis is performed in section 6.1 where the centralized model is analyzed in subsection 6.1.1 and the agent-based model in subsection 6.1.2. The comparison and elaboration of the results are presented in subsection 6.1.3. These models are used as a basis for discussing an AI framework that supports these (Section 6.2).

6.1 ANALYSIS OF SYSTEM ARCHITECTURE

This section looks into analyzing the research questions stated in section 1.2. By using it in conjunction with the state-of-the-art research a sub-problem that deals with the smart home frameworks arises. In general, a research problem can be interpreted and viewed from many different angles depending on the goals of the author. To be able to manage this variety of different actor-views a perspective filtering is needed. The used perspective in this work is the *technology* view. This means that the *users* and *market / economy* perspectives are only discussed to a minor extent when it is needed to clarify subjects. Rejecting perspectives provides losses in the generality and it has influence on the selections and choices made in this work, but the scope needs to be limited.

Using the just discussed perspective view in combination with the research question (Section 1.2) and a top-down approach, it is possible to look into the system architecture candidates of the existing system types, which are discussed in section 4.3. This is performed in the following sections. The system architecture candidates are discussed and elaborated in Lynggaard (2012a).

6.1.1 CENTRALIZED SMART HOME MODEL

To be able to perform a system architecture mapping the approaches used by different authors are discussed and mapped into a backbone model developed in this work. Most of these approaches use a common centralized structure (Section 4.3) that with minor simplifications

can be mapped into most of the layers in the derived Centralized Smart Home model (CSH) illustrated in Figure 6.1.

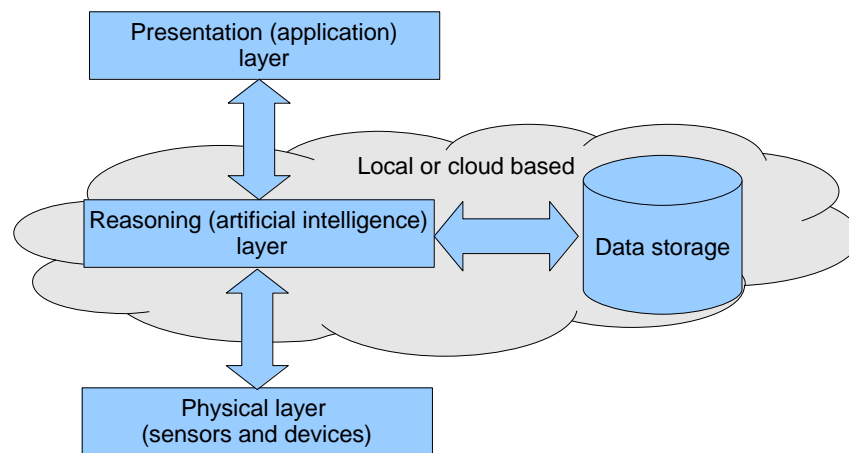


Figure 6.1. Centralized Smart Home model (CSH).

This model defines a *physical layer* that consists of sensors and other devices. It includes both wired and wireless transport of data to the next defined layer the *reasoning layer*. The reasoning layer performs data processing and data pattern extraction, e.g., by employing AI. It deals with the prediction of new data patterns, such as a future user activity, based on sensor inputs and AI processing of the users past behavior. These learned patterns are saved, so they can be used as a reference in the prediction process for future behavior patterns. Thus, if the user does an activity that triggers some sensors the output from these sensors including the triggering order create a sequential pattern. These patterns enable the use of AI methods for future pattern recognition. So, these patterns need to be saved in a *data-storage* (database). Outcome from the reasoning layer are fed to the *presentation layer* that interact with the user either directly by suggestion actions e.g., on a smart phone or indirectly by informing other systems about it. It is noted that the reasoning layer in the CSH model can be implemented as either a local or a cloud service offered by a third party.

A clarifying example of this approach can be found in the work performed by Cheng et al. (2009). Their work discusses a system they named ASBR, which is illustrated in Figure 6.2.

This system uses a centralized approach that is divided into three layers. The first layer contains the home devices and sensors that send raw data to the second layer that handles the

reasoning, i.e., AI. This second layer is supported by a third layer that includes a huge data base containing all the learned patterns. These three layers can easily be mapped into the CSH model shown in Figure 6.1.

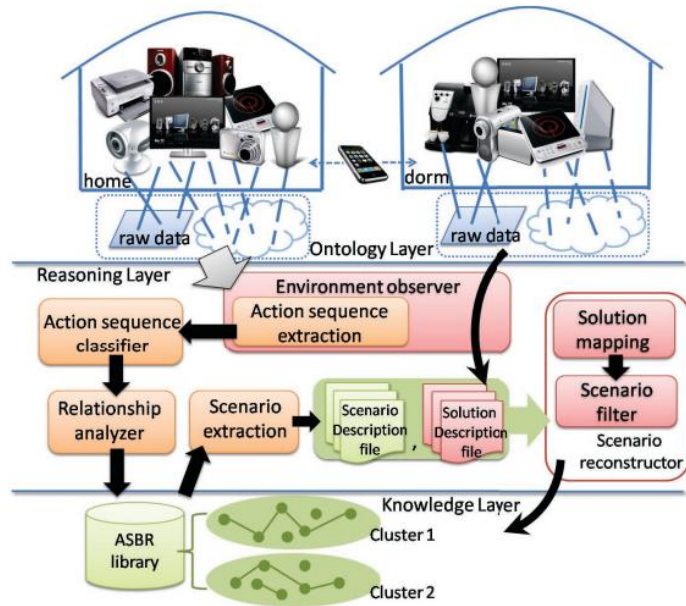


Figure 6.2. Adaptive Scenario Based Reasoning (Cheng et al., 2009).

The “smart home in a box” model is illustrated in Figure 6.3 (Cook et al., 2012). It is part of a CASAS project described in subsection 3.3.4.

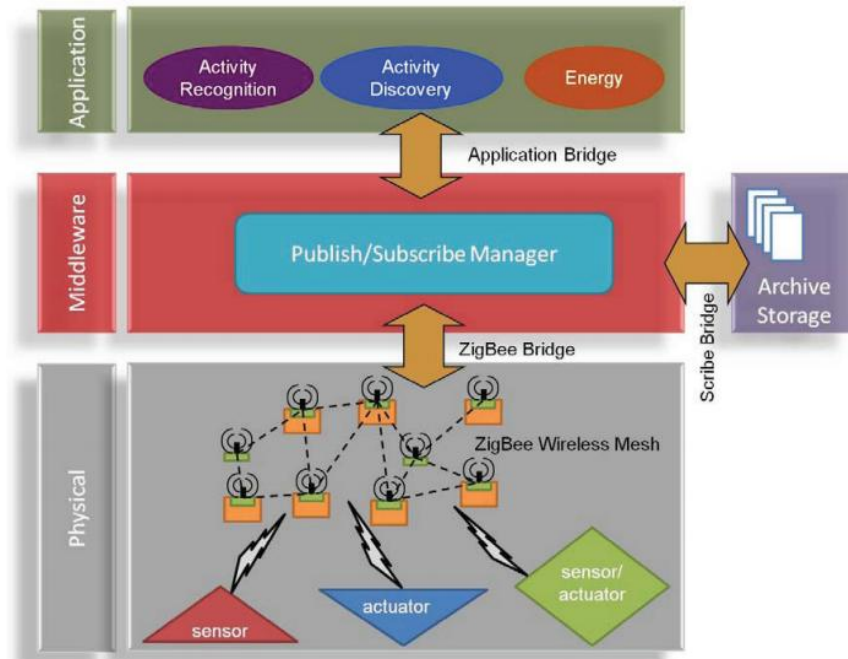


Figure 6.3. "A smart home in a box" model (Cook et al., 2012).

This model uses distributed sensor nodes where each node is equipped with a ZigBee transceiver. A ZigBee bridge transfers the sensor data to the middleware layer that interfaces the storage devices and the application layer. The application layer then runs the AI algorithm on a centralized server. As noted, this model uses a middleware layer that routes the data and performs some simple data manipulations such as adding timestamp and identity to the data.

This model can be mapped into the CSH model. The physical and middleware layers fit into the physical layer of the CSH model. In addition, the application layer maps into the reasoning and application layers in the CSH model.

Ye et al. (2011) have suggested a cloud based approach (Section 4.3) as illustrated in Figure 6.4.

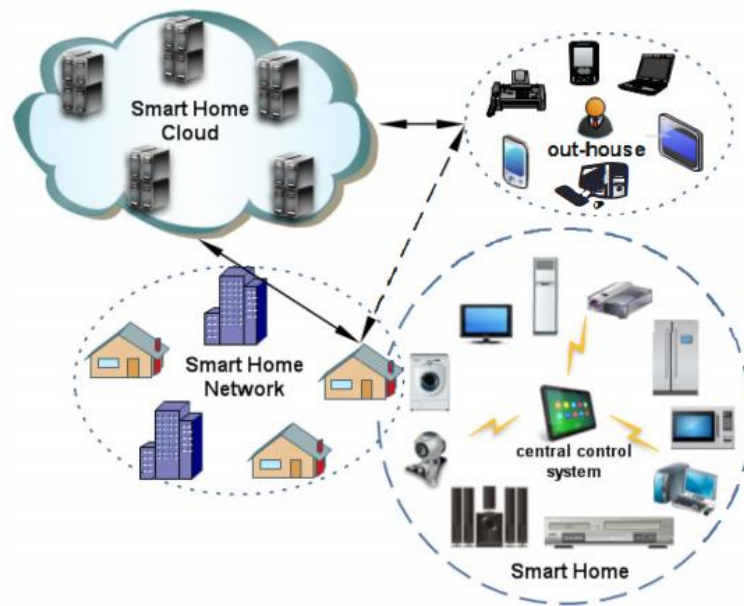


Figure 6.4. Cloud based smart home, (Ye & Huang, 2011).

This approach uses a *centralized control system* that connects all the sensors and devices to it. Because this centralized control system is connected to the Internet it is possible to mirror its behavior into a *smart home cloud*.

The benefit of this system is that many smart homes are able to create a network that can be managed by a smart home cloud service. Such a concept offers AI and high-level user interfaces as a service to its users. It is possible to map this model into the CSH model by mapping the devices and the central control in the cloud based SHS to the physical layer in the CSH model. In addition, the cloud is mapped to the reasoning layer in the CSH model and the user application is mapped into the presentation layer.

Actually, it has been found that most of the centrally based smart home and home automation systems can be mapped into the CSH model by using some minor assumptions and simplifications. In addition, this is also possible for the systems available on the market such as HomeSeer, Control4, PowerHome, Vivint and ActiveHome Pro (Khoo, 2013).

The centralized CSH concept is popular today (2013) among researchers and ordinary customers (Section 4.3) because it is based on the state-of-the-art existing home control system technologies, which are available. Ordinary customers use it for remote controlling

their homes. Researchers use it as a platform for experimenting with AI and context awareness, i.e., they try to transform these home automation based homes into real smart homes.

In relation to smart homes the centralized concept modeled by the CSH model has some flaws. These are the subject of the following discussion.

Analyzing the CSH model interface between the *Physical layer* and the *Reasoning layer* information is exchanged. This gives rise to the following problems:

- It loads the internal SHN with a large amount of sensor and actuator data, including the added load from overhead in form of protocol headers, etc. This means that real small (future) bandwidth SHN protocols cannot be used section 4.5. Furthermore, more power is wasted accordingly to the Shannon information theorem (Subsection 4.5.2).
- If the SHN is (partly) wireless effects such as congestion, noise and fading can destroy data or it can block the wireless channel (Subsection 4.4.4). This problem is accelerated in many small embedded devices because they often use one of the ISM bands that are an unrestricted resource (Subsection 4.4.4).
- To be able to support a variety of smart home devices and communication standards some routing and bridging technologies are needed (Subsection 4.5.1). These can be small embedded low power devices that cannot present sufficient data performance as e.g., queue sizes, i.e., there is a risk of data losses. Data losses can be handled, but it requires bandwidth and processing power which are scarce resources.
- It uses battery resources in embedded devices for transmitting or from time to time re-transmitting including the necessary handshaking needed by the particular protocol (Subsection 4.4.2).

Processing of data from the *Physical layer* are performed in the *Reasoning layer* by some central server that can either be local in the home or based on a distant cloud service. This gives rise to the following problems:

- Using a local centralized server to process all the data means that it can be the bottleneck which slows down the response time. Whereas a cloud based server has

disadvantages such as cost, added response time, and bottleneck problems if the server is overloaded (Section 4.3).

- A centralized server suffers from the “single point of failure” problem.
- A centralized server needs to be always running for the smart home to function properly. This means that costly uninterruptable power supplies and backup systems are needed. Regarding a cloud server these considerations are still valid, because the price of renting the server includes it.

The *Reasoning layer* provides a *data storage* for saving data and patterns. This give rise to the following problem:

- The amount of user events and patterns can be huge meaning expensive data storage devices are needed over the smart home life time.
- These devices need to be fast enough to follow the algorithm processing speed so data can be fed to it in real time. Such devices take up physical space, they are power consuming and they are expensive.
- The storage device needs a backup storage device such as a Redundant Array of Independent Disks (RAID) controller strategy which is expensive. I.e., reliability requirements need to be fulfilled.

As illustrated in Figure 6.1, the *Presentation layer* interfaces the user to the *Reasoning layer*. The following listed problems are identified:

- The user interface application runs on the same server as the Reasoning layer. This means that delays in the processing of the heavy artificial algorithms will be reflected into the user interface, i.e., the prediction of user actions will be delayed.
- Single point of failure for the centralized server will also cause the user interface to fail. This means that the user cannot be informed of the poor system performance, etc.
- Data exchange from the Reasoning layer to the Presentation layer loads the network and its performance.
- There is no one-to-one direct connection between a hand held smart device (e.g., a smart phone) and the related smart home device.

To overcome some of these shortcomings an agent-based systems is often used (Alam et al., 2012). This is the topic of the following.

A Multi-Agent-based System consists of a collection of smart home agents where each one is dedicated to detect a specific activity. To cover the diversity and extent of smart home functionalities many agents must be employed. Cook et al. (2006) express it this way:

The agent seeks to maximize inhabitant comfort and minimize operation cost. In order to achieve these goals, the agent must be able to predict the mobility patterns and device usages of the inhabitants. Because of the size of the problem, controlling a smart environment can be effectively approached as a multi-agent task. Individual agents can address a portion of the problem but must coordinate their actions to accomplish the overall goals of the system (Cook & Das, 2006).

Analyzing this statement, it implies that some kind of predictability or AI is needed. They also state that by organizing the agents according to some hierarchically based system, problems can be subdivided. In addition, structures to make communication paths are needed.

6.1.2 AGENT-BASED SMART HOME MODEL

The methods that were used in subsection 6.1.1 have also been used in the research of the agent-based smart homes, which are discussed in the literature (Cook & Das, 2006),(Del-Hyo et al., 2012), (Hannon & Brunell, 2005), (Mocanu et al., 2013), (Alam et al., 2012). By using the Cook et al. (2006) statement and looking into the literature describing and characterizing the agent-based approach, some common factors are extracted. Based on these factors including some simplification and generalization considerations an Agent-based Smart Home model (ASH) has been derived Figure 6.5. As discussed in the following most of the agent-based smart homes can be mapped into this model.

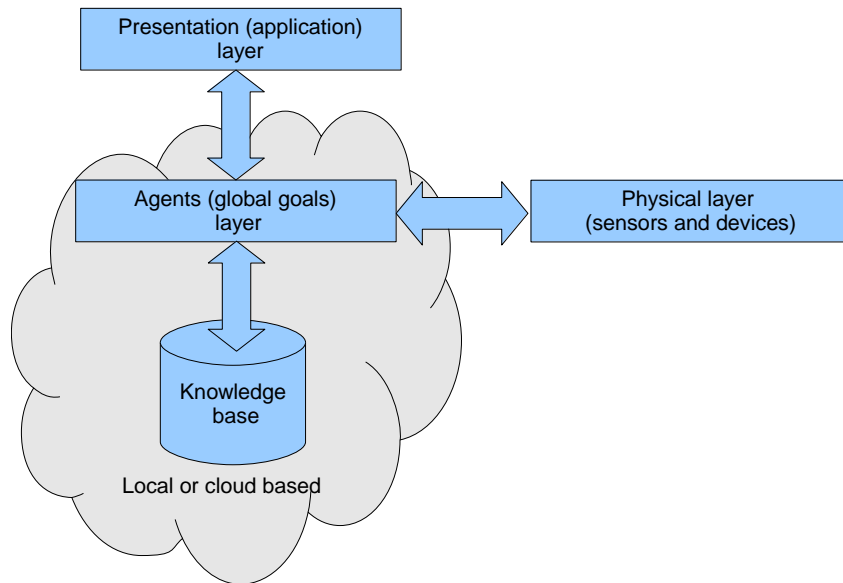


Figure 6.5. Agent-based Smart Home system (ASH).

In most papers the informal definition of an agent is interpreted and used at a high abstraction level. I.e., the agents handle high abstraction level tasks such as disk manager agent, user agent and AI based control agent (Section 4.3).

One example of this approach is provided in the "ThinkHome" project (Reinisch et al., 2010), which is illustrated in Figure 6.6.

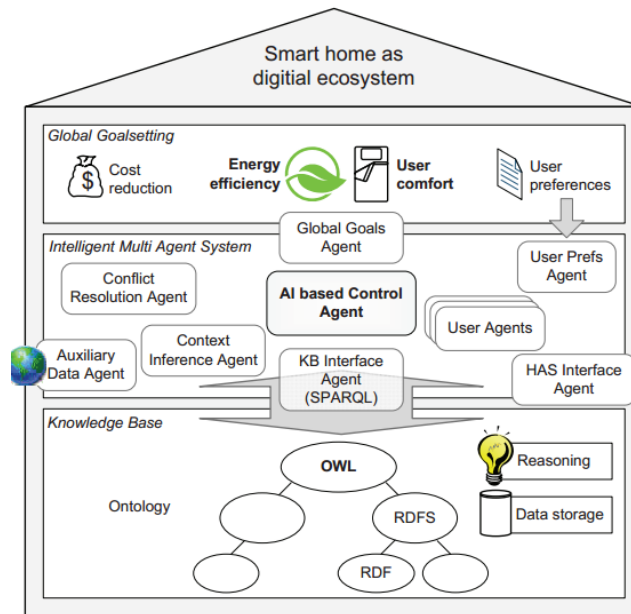


Figure 6.6. The "ThinkHome" smart home project, (Reinisch et al., 2010).

The "ThinkHome" project introduces a concept with three layers: Knowledge base, intelligent multi-agent system, and the UI settings. These layers fit well into the ASH model. Thus, the *Knowledge base* and *Global goals agents* map into the *Agents* and *Knowledge base layers* in the ASH model. Similarly, the *Global setting* maps into the *Presentation layer* in the ASH model. An overview of this project is provided in subsection 3.3.4.

Another concept is presented by Wu et al. (2008). They have used the Open Service Gateway initiative (OSGi) architecture in the research. Their concept is illustrated in Figure 6.7.

This concept contains devices that implement an OSGi agent-based architecture. Each device is able to support a software component (named a bundle in the OSGi context) that is installable from a cloud based service provider. When a bundle is installed it is able to run a contained program. This program detects events and sends them to the service providers. In addition, it receives actions that shall be performed such as turn on the radio and select a particular channel. The service providers also offer a user interface part.

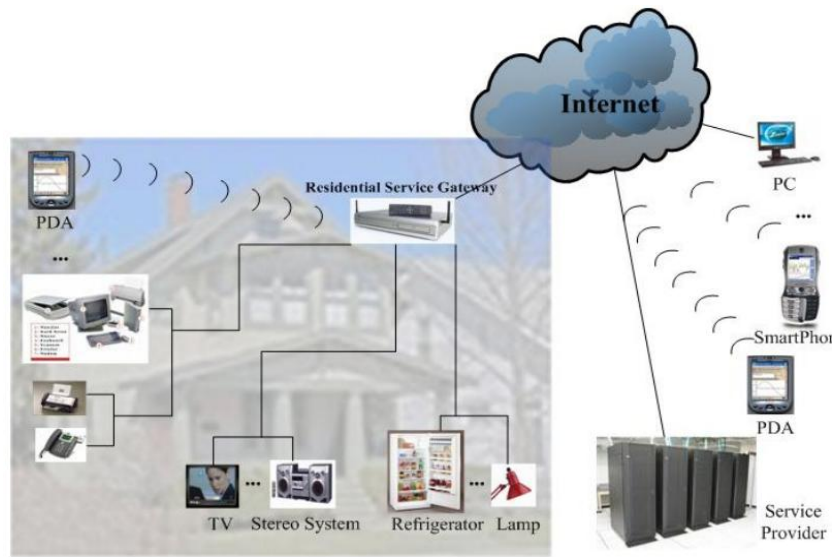


Figure 6.7. The OSGi based smart home architecture, (Wu et al., 2008).

Mapping this model into the ASH model is performed by mapping the OSGi *devices* to the ASH *physical layer*. The OSGi *bundles* together with the *physical devices* create the *agents layer* in the ASH model. Same way, the OSGi *service provider* maps into the *knowledge base* and *presentation layer*.

Das et al. (2012) use a similar approach in a CASAS project as illustrated in Figure 6.8.

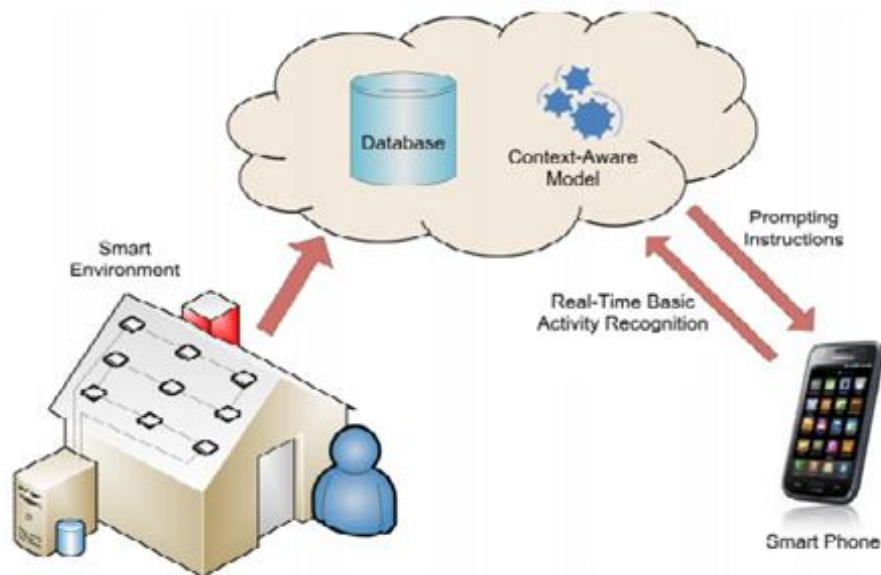


Figure 6.8. A smart home architecture in a CASAS project (Das et al., 2012).

This project uses the extensible message and presence protocol to communicate between the various agents that comprise the smart environment. These agents use a cloud based service to store and process the device and sensor data. The user interface is also connected to the cloud service. It is possible to map this model into the ASH model by using the same methodology as used in the OSGi based SHS.

6.1.3 COMPARING THE CENTRALIZED AND THE AGENT-BASED SYSTEMS

Exploring the pros and cons of the ASH based system versus the CSH system is done in the following. It is noted that if the agents in the AH system are placed in the cloud or on a centralized server there is no difference between this system and the CSH system.

Table 6.1 compares and discusses the problems related to the interface between the *Physical layer* and the *Reasoning layer* in the CSH model Figure 6.1 and the similar problems in the ASH context.

CSH problem	Problem in ASH context
1	<u>Network load.</u> Distributing the agents to a few locally allocated devices, which serve as a processing platform, reduces the impact because the distributed agents also distribute the network load.
2	<u>Channel blocking.</u> Using a processing platform that contains agents, which are distributed to few locally allocated devices, reduces the impact of interferences because the distributed agents shorten the spatial distances between the devices (Subsection 4.5.2). However, more agents also mean more wireless devices in the smart home, so e.g., co-channel and adjacent channel interferences need to be considered devices (Subsection 4.5.2).
3	<u>Insufficient device data performance.</u> Because some of the agents are allocated on smart home devices, which offer some amount of resources, they are able to act as routers or technology bridges for the other devices. Thus, this problem is reduced in the ASH model.
4	<u>Lack of power resources (battery).</u> The primary factor that uses power in a sensor or device node is the transceivers. In the ASH model less power is needed because the spatial distance between the devices and agents is lowered. However, the problem still remains.

Table 6.1. The interface between the *physical* and the *reasoning layers* for the CSH and the ASH models are compared and discussed.

Similar to the previous table the discussion of the problems related to the *Reasoning layer* in the CSH model Figure 6.1 and the same problems seen in the ASH context are presented in Table 6.2. Hence, the problems in ASH context are mapped as a function of CSH problem number.

CSH problem	Problem in ASH context
1	<u>One processing server can be a bottleneck.</u> Since the processing is distributed to the local agents with their own processing resources this problem is solved.
2	<u>Single point of failure if the system is based on only one central server.</u> Same argument as in CSH problem 1. However, if one of the agents drop out it means that the smart home services either degrade or fails. If e.g., the user interface agent drop out the smart home is still able to control the heating by using the heating agent.
3	<u>Central server needs to be always running.</u> Hibernating features can be implemented at agent level because some agents will not be affected by the ongoing device and sensor events produced when the user interacts with the smart home. An example is the heating interface that will only run when the room temperature changes slowly over time. So this problem is reduced, but not completely solved.

Table 6.2. The reasoning layer processing for the CSH and the ASH models are compared.

Regarding *data storage* for saving data and patterns from either the *reasoning layer* or the *agent layer* no change exists between the ASH and the CSH approach. However, some of the devices in the ASH approach can be equipped with some resources that can be used in a distributed approach. An example of this could be a smart TV with Internet connection.

The discussion of the problems related to the interface between the *presentation layer* and either the *reasoning layer* or the *agents layer* in the CSH model Figure 6.1 and the similar problem seen in the ASH context is presented in Table 6.3. Mapping the ASH problem as a function of CSH problem number yields:

CSH problem	Problem in ASH context
1	<u>The user interface application runs on the same server as the either the <i>reasoning</i> or the <i>agents layers</i>.</u> If the <i>presentation layer</i> is executed on a cloud service the server farms available will provide the needed amount of processing power. Whereas if the <i>presentation layer</i> is executed locally on the smart home devices the distributed processing from these devices provide some degree of processing power. However, some overhead must be expected to interface the user interface to the individual agents.
2	<u>Centralized server single point of failure will cause the user interface to fail.</u> If the user interface runs as a cloud service this service is expected to have backup strategies. Whereas, if it is executed locally by a user interface agent and it fails the user interface will not work.
3	<u>Data exchange from either the <i>reasoning layer</i> or the <i>agents layer</i> to the <i>presentation layer</i> loads the network and its performance.</u> When the <i>agents layer</i> update the <i>presentation layer</i> it loads the network almost the same way as the <i>reasoning layer</i> , so only minor or no change in network load.
4	<u>There is no one-to-one connection between a hand held “smart” device (e.g., a smart phone) and the related smart home device.</u> Using the agent-based principle does not provide this connection, because only one agent takes care of the overall user interface. So, no change.

Table 6.3. The interface between the agent, reasoning and the presentation layers for the CSH and the ASH models are compared.

The pros and cons of the initial problems are summed up in the following.

In the discussion of using an agent-based system in a smart home (ASH) compared to using a centralized system (CSH), it is reasonable to assume that distributing the reasoning layer into some agents also distribute the network load. This means that the heterogeneous network elements and its transmission channels connecting them to the distributed agents are able to work in parallel. This lowers the network load in the bottlenecks as e.g., a centralized server would create. Distributed agents also provide some of the benefit given by a lumped network such as shorter distances, i.e., less noise sensitivity; lower power consumption subsection 4.4.2; and the possibility to use different routing and bridging technologies, i.e., an average lower network load is expected.

Regarding single point of failure and power saving issues, the agent approach also benefits from its distributed nature. Agents that are not in use can be powered down and failing agents

will only have minor influence on the others that still are able to work, but with a reduced performance.

There are downsides and non-affected parameters which the discussed distributed agent system does not solve. Firstly, when a specialized agent fails it has impact on the SHS. If e.g., a heating agent fails the complete heating system will also fail, because it depends on this agent only.

Secondly, power consumption of the wireless devices and sensors are not affected much by using the discussed agent concept. Adding more wireless agents, which communicate in parallel, also adds a new radio disturbance problem in form of co-channel and adjacent channel interference in the frequency division multiple access area compared to the combined TDMA and frequency division multiple access schemes used in a centralized server approach.

Thirdly, the user interface does not provide a clear coupling between the smart home devices and a portable interface device. Running the user interface as a cloud based service has the potential risk of creating a network bottleneck problem between the agents and the cloud server farm.

Dividing a smart home into few functional agent units does not solve all the problems. However, taking this approach one step further in the direction of IoT may be beneficial, why this work focuses and contributes in that direction. Additionally, this concept is in agreement with the conclusion presented by Alam et al. (2012) who discuss the past, present and the future of smart homes.

6.2 AI FRAMEWORKS

Analyzing the state-of-the-art for a useable AI framework for smart homes takes offset in the second research question stated in section 1.2. It states that the distributed AI framework must support a distributed system architecture including its S.O. by using existing technologies and devices. By combining these challenges with the analysis of the system architecture discussed in section 6.1 the focus is set. Hence the CSH and ASH models presented in section 6.1 provide the needed structures for dealing with the research question.

Using a centralized framework in a distributed smart home concept has its challenges as discussed in sections 6.1 and 4.3. Nonetheless, placing the high-level AI part in a centralized framework offers the needed processing resources, it facilitates access from the low-level system, effective backup of data and straightforward software updates. Thus, a centralized approach offers the needed resources for complex AI processing. An alternative would be to place it in a distributed framework with the needed resources; however, as stated by Alam et al. (2012) this is suboptimal. In addition, it is suboptimal because the device power consumption and complexity is a function of the offered resources; and the future related IoT research area strives for low resource devices (Section 3.4), (López et al., 2012).

Another advantage of placing the high-level AI system in a centralized framework is that the inputs from multiple devices are readily available for temporal processing. Thus high-level temporal devices benefit from a centralized star topology.

The ASH model (Figure 6.5) is based on a concept where the AI framework is distributed into agents. The existing agent-based frameworks discussed in section 5.7 cover high-level agents which are functionally oriented, i.e., one agent handles all functionality in a specific area. Such a framework is equivalent to a centralized concept with multiple nodes, i.e., one agent handles some specific functionality that needs inputs from all the sensors inside this area, independent of where they are spatially allocated. Noticeably, this is suboptimal as stated in the centralized framework discussion.

To obtain a distributed AI framework which supports the distributed smart home concept stated in the research questions and discussed in the ASH model, a low-level AI framework is needed as discussed by Alam et al. (2012) and in section 5.7. Nonetheless, a pure distributed AI system has disadvantages such as: They do not share resources, they risk duplicating the work, and they need individual software updates. So its lacking ability to share resources means that it does not facilitate high-level temporal AI algorithms and it does not support concepts such as shared processing, and shared data management. Thus, a low-level agent-based AI framework needs to interact with a centralized high-level AI system for combining and using information. In addition, the cost of duplicating work such as sensor event filtering, etc. is present. However, this can be advantageous if the processing cost is low compared to the cost of moving data.

Viewed from a future perspective the low-level AI systems will benefit from the research area of IoT. This is mainly because the IoT area provides frameworks and support for agent-based approaches such as distributed smart home AI frameworks, and distributed processing platforms (Section 3.4), (López et al., 2012), (Chen, 2013).

6.3 SUMMARY

This chapter presented an analysis of distributed AI frameworks and it derived two system architectures named CSH and ASH. The AI framework was discussed in the view of the two system architectures and a hierarchical concept. This hierarchical concept combined the AI advantages of these two models. Because the CSH and ASH models are generic by nature they primarily capture essential points, i.e., they compare and discuss the centralized versus the distributed categories. Mapping some of the models used by researchers today (2013) into these models revealed a good match in one of the categories. By using these models it was possible to compare strengths and weaknesses in terms of the many factors that have influence on these.

The results from comparing these analytic model outcomes were discussed and it was found that the agent-based approach offers numerous benefits in a distributed smart home concept. This is in good agreement with conclusions of Alam et al. (2012) based on their research in this area.

7 DISTRIBUTED SYSTEM ARCHITECTURE FOR SMART HOMES

The distributed system architecture for smart homes is explored and designed in this chapter. The derivation starts with the overall system architecture which is designed in section 7.1. This system architecture mainly consists of a high and a low-level part. The high-level part facilitates advanced AI systems and is allocated on resourceful devices (Section 7.1). In contrast, the low-level part consists of distributed devices, named S.O.s, which offer support for simple AI systems and are implementable on resource constrained devices (Section 7.2). The inter-device communication is explored and discussed in section 7.3.

An overview of the terms used in this and the following chapters is provided in Figure 7.1 and described in the following.

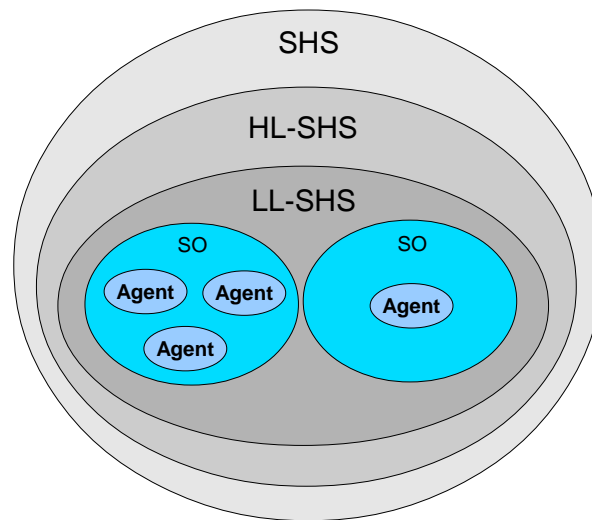


Figure 7.1. Relations between the HL-SHS, LL-SHS, S.O.s and Agents.

At the outmost level the Smart Home System (SHS) encapsulates the two subsystems named High-Level Smart Home System (HL-SHS) and Low-level Smart Home System (LL-SHS). The HL-SHS is an add-on to LL-SHS. It adds more advanced AI prediction and learning capability to the SHS. It is based on the emitted actions from the LL-SHS, which it uses as input. The LL-SHS is a collection of one or more S.O.s, which in turn includes a collection of

agents. In addition, it covers instances of the S.O. software that are implemented on some processing platform such as a PC or a smart-TV.

7.1 DERIVING THE SYSTEM ARCHITECTURE

The smart home concept derived by this work is discussed and presented in the following. It is based on the discussion in chapter 4 concerning centralized versus agent-based SHSs, the discussion of the state-of-the-art provided in chapter 6; and it is guided by research question one. The content in this section is available in elaborated form in Lynggaard (2012a).

7.1.1 ARCHITECTURE DEVELOPMENT

Agent-based systems have many benefits compared to centralized systems (Chapter 6). However, these agent-based systems use high-level specialized agents, which perform dedicated functions (Section 3.3). To decrease this level and transform the smart home concept into a distributed world, it is combined with the distributed concept found in the IoT area (Section 3.4). Such a concept offers benefits from the centralized area as well as from the smart devices (S.O.s) area (Subsection 4.4.5), (Silva et al., 2012), (Alam et al., 2012).

Some of the conceptual ideas used in this work are related to the work of Alam et al. (2012). In their work they discuss the following perspectives concerning agent-based systems:

It seems that home intelligence will be employed in a distributed manner. This distributed intelligence may be applied in the form of smart devices (Alam et al., 2012).

This statement is in agreement with some of the discussed conceptual ideas behind this work (Section 1.2). Similar results are also found in section 6.1.3 that compares agent-based systems with centralized systems.

As discussed, to derive the distributed system architecture including its S.O's, the IoT research area needs focus. This is the target for the following discussion.

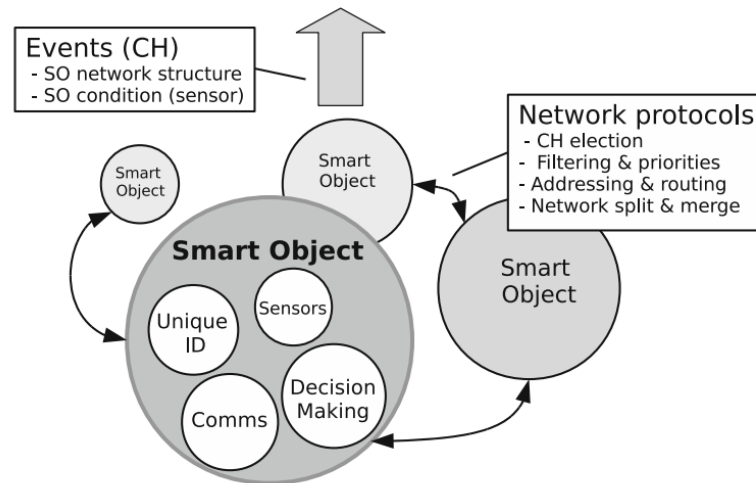


Figure 7.2. Internet of things based smart object (López et al., 2012).

As stated, the IoT area moves commonly known devices and objects to the Internet in a distributed manner (Bandyopadhyay & Sen, 2011), (Sundmaeker et al., 2010), (Section 3.4). López et al. (2012) propose an S.O. framework for IoT devices that is able to provide encapsulated RFID devices, sensor technologies, embedded object logic and ad-hoc networking. Their S.O. framework is illustrated in Figure 7.2. Furthermore, they propose five fundamental properties for an S.O. framework:

- It needs a unique identity.
- It must be able to measure and store data from sensors.
- It must provide a mechanism to hand out measurements on request.
- It must be able to communicate with other S.O.s.
- It must be able to make decisions.

These points are important in the architectural design of a distributed independent S.O. which provides context awareness. This S.O. needs to be able to communicate with the outer world for distributing its measured results and be able to receive input from other similar devices and a management interface. López et al. (2012) also list the ability to make decisions as a fundamental property. They argue that it reduces latency and the processing complexity does not increase with the number of network nodes. However, they do not provide any guidelines on how to solve the challenges in implementing the decision making mechanism.

The "ThinkHome" (Subsection 3.3.4) is a smart home project, which uses distributed intelligent agents (Reinisch et al., 2010). Reinisch et al. state:

To realize optimized control strategies that allow maximizing energy efficiency and user comfort simultaneously and automatically, methods from AI need to be employed.

An excellent means are multi-agent systems that inherently support distributed intelligence and collaboration to act towards defined goals. Different agents are brought together by an agent-based framework that also embeds the intelligent control strategies... (Reinisch et al., 2010)

Their approach supports the theory that artificial agents in form of a multi-agent system is the most useable approach for future smart homes seen in the light of energy efficiency and user comfort. However, their presented approach uses a centralized knowledge base as the AI provider. This means that the agent still needs to consult a centralized device with the flaws discussed earlier in the CSH concept. Another problem is the high-level tasks assigned to the agents as discussed in subsection 6.1.3.

A system architecture has been derived in this work from the above discussions and considerations, the state-of-the-art research (Chapter 6), the centralized versus agent-based systems (Chapter 4), the distributed AI framework (Chapter 5), and the research questions. This system is based on today's (2013) technology, i.e., it is a distributed system based on the ASH concept and it uses multiple autonomous S.O.s. The model is presented in the next section.

7.1.2 PROPOSED SMART HOME SYSTEM MODEL

By adding all the observations and discussed items into the ASH model (Figure 6.5), a design of the suggested SHS can be derived (Figure 7.3).

As illustrated in Figure 7.3 the ASH model is modified into the presented and suggested SHS. It combines the *physical layer* (sensors and devices) with a collection of agents, i.e., it is a multi-agent-based system. This multi-agent system is contained in a *smart object* (Subsection

4.4.5) together with a simple AI system (the *knowledgebase*). It is noted that this *smart object* has its roots in the IoT area (Section 3.4).

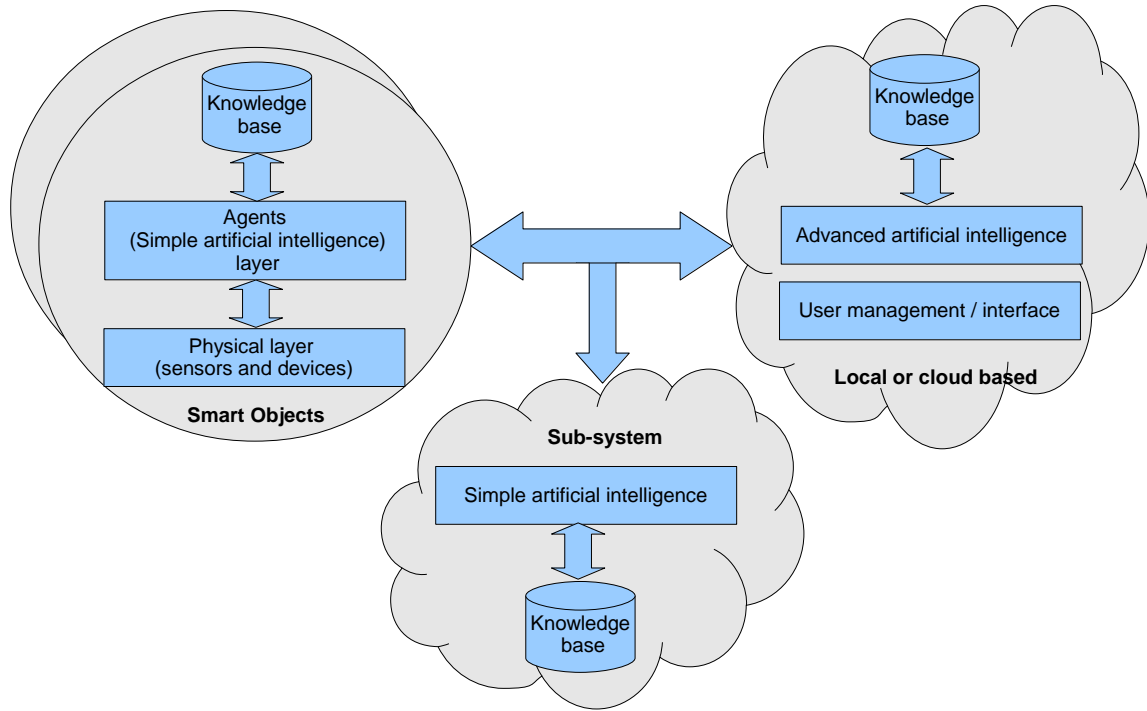


Figure 7.3. The smart home system presented in this work.

This S.O. model offers:

- A standalone entity that has a unique identity.
- Ability to measure and store data from sensors that can be handed out on request.
- Ability to communicate with other S.O.s.
- Ability to handout measurements.
- Ability to make its own decisions.

This model is in a good agreement with the discussed principles stated by López et al. (2012). In addition, the principles described by Alam et al. (2012) are also incorporated.

As illustrated in Figure 7.3 the S.O.s are able to communicate with the user through a user management system allocated on a local server or as a cloud service offered by a third party.

The *advanced AI* system is positioned on a local server or in the cloud. It is able to make complex decisions beyond the capability of the simple AI implemented in the S.O.s.

The illustrated *sub-system* implements a *simple AI* system. This system is able to support the same AI algorithm, as the one running on the S.O.s. However, the processing platform that supports this algorithm is different; it could be a common home router or a Network Attached Storage (NAS) device. Thus, the *simple AI* system symbolizes that the S.O. algorithms (or similar algorithms) are able to run on other processing platforms, e.g., implemented in a hbbTV by its manufacturer.

The basic principles behind the SHS concept are to combine the communication, processing and most of the AI parts into small embedded units or devices, i.e., the S.O.s. These S.O.s are then assigned a simple atomic functionality in form of offering a particular service to the user (i.e., an action). A limited functionality provides saved resources and offers small form factors, why they can be positioned in a spatial context close to their related sensors. The rationale is that when the user performs an action it is bound to the sensor spatial context (Subsection 7.3.4). E.g., when the user opens the fridge, takes a plate, sits down and eats breakfast all these events are bound to the kitchen context. An S.O. that detects the user is eating will be related to this context and therefore be positioned in the kitchen.

One of the major challenges in this distributed S.O. concept is that it has to do all the AI processing. This requires processing, bandwidth, storage and power resources that would limit the usability of such an S.O. device (Alam et al., 2012). Thus, to overcome this problem a distributed AI approach has been developed. It makes it possible to divide the AI system into a simple and an advanced part see chapter 8.

7.2 SMART OBJECT MODEL

One of the most important key elements in the derived SHS is the S.O., why the derivation of its model, communication architecture, and implementation architecture are discussed.

7.2.1 SMART OBJECT LAYERED MODEL

By using the S.O. part, included in the SHS (Figure 7.3), the five described guidelines provided by López et al. (2012), and the model (Figure 7.4) presented by Hannon et al. (2005) some common structure can be observed. These are discussed in the following.

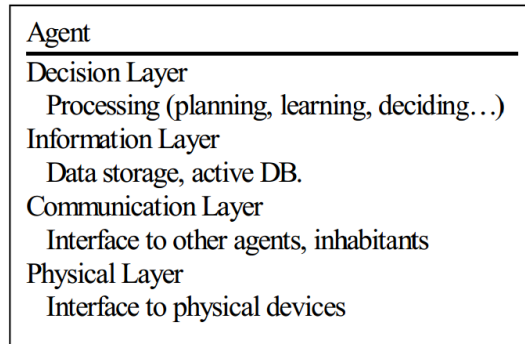


Figure 7.4. Layered agent model (Hannon & Brunell, 2005).

Comparing the elements from the derived S.O.s (Figure 7.3) with the Hannon et al. (2005) model (Figure 7.4) starts with the *physical layer*. This layer simply compares one-to-one because they are functionally overlapping. Same observation and argumentation maps the *communication layer* one-to-one. Regarding the *agent layer* and the *knowledgebase* it maps to the *information* and *decision* layers respectively.

In addition, the derived S.O. model (Figure 7.3) is closely related to the WSN node models (Pensas & Vanhala, 2010), (Sohraby et al., 2007). Hence, the well established theoretical framework that is available in the WSN context can also be used to model parts of the S.O.s.

7.2.2 SOFTWARE ARCHITECTURE

As discussed, the WSN node models and the S.O. model are closely related; that is why the theoretical framework that governs these general WSN node models can be used to establish a generic framework for the S.O. software architecture. Such a theoretical framework is presented by Sohraby et al. (Sohraby et al., 2007) as illustrated in Figure 7.5.

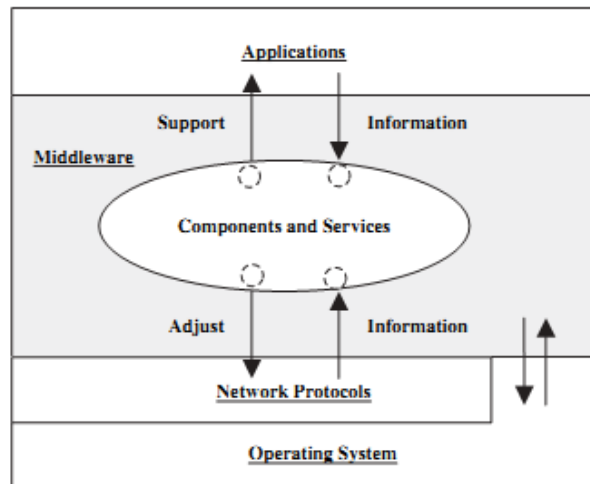


Figure 7.5. A general software architecture for WSN (Sohraby et al., 2007).

The most important component in the WSN software architecture which is usable for S.O.s is the middleware layer (Figure 7.5). Its purpose is to establish a coupling between the network communication layer and the application layer. This component offers the ability to save power by adjusting the communication channel parameters in an optimal way that fits the running application. In addition, the WSN middleware layer offers some functional elements such as event detection, resource management and an application programming interface.

In general, middleware layers have been described and researched in many SHSs by different authors. Pensas et al. (2010) have presented a centralized middleware approach where one node runs the middleware that handles all the device and sensor communication. However, such an approach suffers from the disadvantages discussed previously in connection with the CSH system. A similar paper is presented by Tu et al. (2009). The popular OSGi framework has also been used by many authors (Liutkevičius et al., 2011), but this concept suffers from a high complexity level and a high amount of overhead (Cheng et al., 2008).

Another concept is to distribute the middleware layer to each sensor node, i.e., the S.O.s in the smart home context. Only few authors look into this concept (Zoref et al., 2009), (Bregman & Korman, 2009). One popular project that supports this view was the EU founded HYDRA project (Eisenhauer et al., 2009), (section 3.5.1).

The HYDRA middleware is very complex and aimed towards platforms with considerable processing power. It is limited to microprocessor devices with considerable resources such as 32 bits architecture and hundreds of kilobytes of memory (Gugliotta & Guarise, 2009). This means that it is not well suited for small simple embedded S.O. systems based on microcontroller platforms (Subsection 4.4.5). Another work presented by Pensas et al. (2010) is the EPIS middleware concept for WSN. This work offers a framework for context based reasoning and automated node positioning with focus on integration, maintenance and reliability. However, subjects as performance and minimal memory footprint were not considered, i.e., it is not advantageous in an S.O. context.

Conclusively, none of the existing systems provide a middleware framework that is usable for S.O.s. Based on the WSN framework concepts some guidelines for designing it are discussed in the following. Looking into the related area of Internet of Things (IoT) it provides some abstract but useful guidelines CERP-IoT (Sundmaeker et al., 2010). The most relevant ones in an S.O. middleware context are that it must:

- Be transparent.
- Be delay tolerant.
- Provide storage capability.
- Be able to manage power consumption.
- Support grid and cloud based networks.
- Be able to handle hybrid network technology and devices.

Alkazemi et al. (2012) have a similar list where they outline the middleware content for a general WSN. By combining this work with the work by CERP-IoT (Sundmaeker et al., 2010) a seven point list of S.O. candidates can be derived:

1. Interaction Pattern: Defines the mechanism that a middleware component uses to interface applications and the underlying network. Thus, transparency from the S.O.s public user interface must be offered. It must support hybrid network technology and devices.
2. Data exchange: Middleware must handle the two main data exchange types pull and push. These types need to be delay tolerant.

3. Software service: The middleware component must offer a set of services to the environments. In addition, it must add storage and delay tolerant capability to this.
4. Proxy pattern: That handles requests coming from the applications and sends them to the addressed sensor, S.O.s and management networks. This must include grid and cloud based networks.
5. Parallel Processing: Multi-thread of execution must be offered.
6. Data conversion: Match data type sent by the application to the expected network data type.
7. Manage lifecycle: Ability to manage power consumption.

Finally, combining this seven point list with the models presented by Hannon et al. (2005) and Sohraby et al. (2007) yields a useable software architecture for an S.O. as illustrated in Figure 7.6. Lameski et al. (2011) presented a WSN based framework with a similar architecture, but this work is missing vital S.O. elements such as managing lifecycle, software service and proxy pattern functionalities.

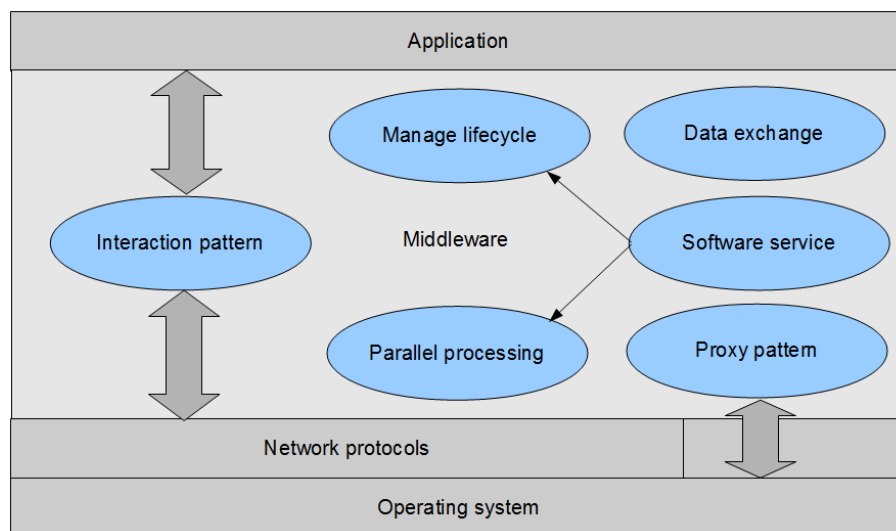


Figure 7.6. Software architecture for the S.O.s used in this work.

As illustrated in Figure 7.6 the suggested software model for the defined S.O.s consist of four layers. Starting with the *application layer*, it handles:

- A collection of S.O. applications such as AI algorithms (a multiple agent framework).
- Calendar and activity schedulers including a system timer unit.

- Sensor input processing such as filtering and user action detector.
- Actuator output processing and general control logic.

Next layer is the *middleware layer* that has already been discussed. Third layer is the *operating system layer*. In a WSN context many operating systems exist. They are all optimized for embedded processing under the condition that the resources are very limited. In general they offer compact small size and small memory usage, i.e., they support small embedded controllers that have limited memory from one to a few tens of kilobytes.

These operating systems offer real time resources for running algorithms such as an AI prediction and learning tasks. This resource supports that a predicted service can be offered to the user as fast as possible. As commonly known designing parallel tasks for a real time operating system is difficult. It is possible to fail from deadlocks and low priority tasks that are preempted in a non-voluntary way. The operating system also needs to take care of memory scheduling in an efficient and fair way because this resource is limited and must be shared between the tasks. Power management is also a task for the operating system because of the limited power resources available for most S.O.s. An example of this could be that an agent (one instance on an AI network) is inactive, which is actually the case most of the time, and therefore can enter sleep mode, i.e., it is preempted by the operating system. When an event arrives from a sensor the agent is re-scheduled for running.

The operating system should also provide support for reliable and efficient code update and distribution. In the model Figure 7.6 the operating system should offer a generic low coupled interface to the middleware layer and offer good support for the sensors and actuators at the physical layer.

From a summary point of view, the software architecture useable for an S.O. has been presented and discussed. It is based on compromises of using the most updated knowledge from different research areas in the field in combination with knowledge from related areas such as IoT and WSN.

7.2.3 SMART OBJECT COMMUNICATION ARCHITECTURE

Smart devices (S.O.s in this work) are derived from a vision in which these form an integral part of the communication structure that connects the smart home devices (Hannon & Brunell, 2005). Actually, the EU commission forecast that the area of smart system communication is a very interesting research area and they therefore have focus on these topics in the FP7 research and development projects (EU_Commission, 2009).

The smart home devices need to be interconnected through a communication network to be able to sense, share and exchange information with each other. Additionally, they need to communicate with other systems such as sensors, actuators and a user management system. It is obvious that this communication must be flexible and configurable so that the devices can communicate with each other. This offers optimal configuration and service abilities for the smart home. However, such an approach involves a complex pattern of communication activities that in some way needs to be handled and managed (Section 4.5). Another complicating factor is the multi-platform approach that smart homes need to be able to handle (Section 4.4).

A detailed overview of the communication context related to an S.O. is given in an S.O. context diagram (Figure 7.7) based on the work of Hay (2011).

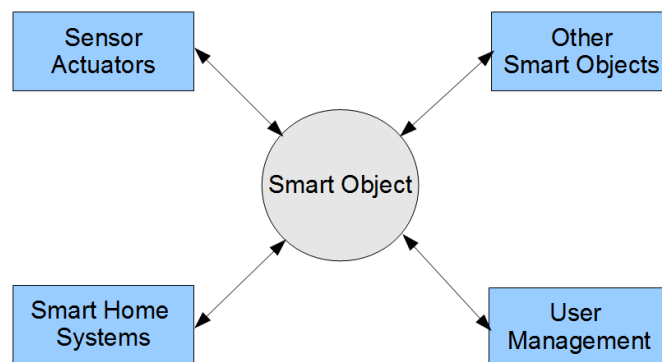


Figure 7.7. Context diagram for a smart object.

Basically, a context diagram describes the developed system in the middle circle and it shows the surrounding terminators (square boxes) that the developed system depends on. Thus, each terminator interfaces to the developed systems by flows that contribute to its behavior. Each of the terminators are described and explained in the following.

The “Sensor and Actuator” terminator is positioned in the upper left corner Figure 7.7. It models the sensors and actuators that are connected to the S.O.s. These sensors can measure anything from temperature, light and pressure to simple on / off door switches. Some of these sensors are hardwired to the S.O., but in some situations and places it is not possible to drill holes and mount wires (Subsection 4.5.3). An alternative solution is to use a wireless link between a sensor node and the S.O., but for the cost of loading the battery in both the sensor and the battery powered S.O. nodes (Section 7.3). Such a wireless link must offer the necessary bandwidth (i.e., bit-rate) to transfer the event messages and support simple setup for some sensor types. Nonetheless, this event information is limited to simple binary states as discussed in subsection 4.4.1, i.e., the most commonly used wireless network types support this (Subsection 4.5.6).

Regarding the actuators they only need to be updated in an asynchronous manner, i.e., whenever an S.O. decides to do it based on its predictions and programmed behavior. The amount of information is limited (Bhardwaj et al., 2012).

The S.O. and its peripheral sensors and actuators are illustrated in Figure 7.8 and discussed in section 4.4.

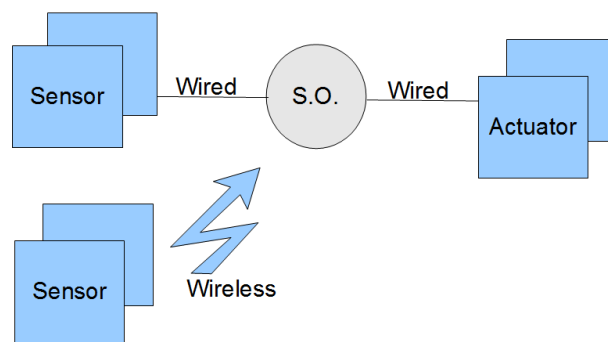


Figure 7.8. A smart object with its peripheral sensors and actuators.

Common for all types of sensors is that they measure some parameters that are relevant in a smart home context and quantize these measurements into usable numbers that are transferred to the S.O.s. How often these numbers are transferred depends on their measurement bandwidth. E.g., a room temperature sensor only needs a low bandwidth i.e., in a fraction of hertz. Thus, according to the Nyquist-Shannon sampling theorem the sampling frequency then

can be selected twice that bandwidth. In general the sensor sampling frequency is in the interval 0.1 - 10 Hz. The amounts of data that are transmitted also depend on the sensor type. More details can be found in subsections 8.5.3 and 4.4.1.

The “Smart Home System” terminator models the AI circuits and their physical positions. Thus, as suggested in section 7.1 and illustrated in Figure 7.3 simple AI is allocated in the form of agents contained in the S.O.s. Whereas the advanced AI components (HL-SHS) can be implemented on available hardware platforms that offer the needed processing power (Section 6.2). Alternatively, they can be allocated in the clouds as a service running on cloud server farms. Common for these systems is their need for communication. Thus, the HL-SHS uses the predicted actions from the S.O.s (LL-SHS) as input for making more advanced predictions (Chapter 8).

The amount of information exchanged between the HL-SHS and the S.O.s is limited because it is mainly based on predicted actions that are closely related to the user behavior, i.e., their frequency is in average below a few hertz. In addition, the information size is limited. However, it should be noted that this communication is based on a communication link that requires some amount of protocol overhead and they are often bandwidth limited to save current in the nodes (Sohraby et al., 2007), (Section 4.5).

The “Other Smart Objects” terminator indicates that S.O.s interacts with the other S.O. inside its context. A detailed coverage of this subject is presented in section 7.3, why this paragraph only provides some overview.

Interactions between S.O.s are fully analog to the WSN world where nodes are able to interact and communicate (Sohraby et al., 2007). This approach allows S.O.s to be combined into units that can handle distant sensor events. An example that illustrates this concept is a room temperature controller. Assume one of the S.O. contained agents control the room heating by using information from the room temperature sensor, the user present sensor and information learned from the room temperature profile as a function of time. But, it would also be sensible to take the windows status into consideration so the heating is turned off when the windows are open. For the heating agent to handle this it only needs an event from the sensors that

monitors the windows. Such a concept saves a lot of wired and wireless communication load compared to a centralized concept.

The S.O.s need to receive control information useable for controlling connected actuators. I.e., they act as a kind of “hardware driver” that is able to control physical objects such as controlling the light level from a lamp. Another use of these control possibilities are the user interface remote controls. Thus the user should be able to remote control an actuator device, as e.g., turning on the light using a smart home interface such as a smart phone. More details are provided in section 3.2.

In general, enabling the variety of S.O.s intercommunication capability some standardization and general abstract data models regarding interfaces and protocols must be defined (CERP-IoT et al., 2010). A suggested general abstract data model in form of a middleware layer has been discussed earlier (Subsection 7.2.2).

The “User Management” terminator indicates that it is possible to communicate with the S.O. at a human understandable level. Thus, S.O.s need to be managed on the fly and they need to be initialized the first time they run by using a setup menu. In general, the management task needs to handle setup of both autonomous and controlled behavior. Controlled behavior means that the S.O.s can be remote controlled and its internal timers and calendar can be programmed to perform timed actions (Subsection 7.2.4). Setup of autonomous behavior needs to provide functionality to erase learned agent AI behavior, connect actuators, and setup internally routes. Thus, all the functionality explained in subsection 7.2.4 needs to be handled.

Many researchers use a concept where a large part of a smart home management system is allocated on a server or offered as a cloud service (Dongmei & Zhiguang, 2010), (Pensas & Vanhala, 2010) and section 4.3. This approach is used by Kim et al. (2012) who propose a home gateway concept where each S.O. is controlled from a centralized application which provides an Internet based user interface. As outlined earlier such a centralized approach is not useable in this work.

Another work by Del-Hyo et al. (2012) suggests an advanced self-management home network based on neural network virtual agents. This work addresses high-level smart home devices

such as TV, Blue-Ray players and Radios. However, this concept requires a large amount of processing power and it uses the home networking (Subsection 4.2) as connection media, so this concept is not useable in this work. A distributed multi-agent framework is presented in a paper by Hannon et al. (2005), (Section 4.3). The user interaction with these agents takes place through a voice recognition interface implemented in each individual agent. Such an approach is unsuitable because centralizing the user interface in a few high-level agents requires that they are actually present in the smart home which in turn put restrictions on the freedom of the smart home device choice. Furthermore, these agents need to communicate with the more low-level agents that have limited resources. This means that factors as network load, agent software maintainability and smart home cost are not reduced considerably, i.e., not enough to justify the increased complexity such an approach adds.

A paper by Liutkevicius et al. (2011) focuses on important factors in a smart home management system. They discuss (1) the cost of devices and maintenance of these, and (2) easy expansion by new devices and services. By researching existing SHSs they found that these offer a limited set of fixed services. This forces the user to wait until the service is deployed and the SHS is updated accordingly. Consequently, the user expenditure for purchasing a new service is affected. By researching existing SHSs, they found that these have a limited set of fixed services.

Looking into the cost of the devices, it is proportional to the development investment. So using a standardized middleware framework as discussed earlier in this work will bring the cost down and allow many manufacturers of common S.O. devices as e.g., lamps, heating system elements and entertainment equipment to compete. This will open up the market and thereby lower the prices. Regarding maintenance of these S.O. devices it would be beneficial to place a unique identity in each device that is addressed by the user interface. So when the user wants to interact with a device its identity is used to present GUI elements provided by the manufacturer, e.g., available on the manufacturer's server. This concept will offer easy maintenance and also support the possibility for easy expansion by new devices and services.

Mounting an RFID tag in the smart home devices can be beneficial because it provides an easy interface for getting a limited amount of device information. Many papers suggest such a concept (Xuenei et al., 2008), (López et al., 2012). An approach that uses RFID tags are

presented by Darianian et al. (2008) where IoT devices are tagged. Their approach is to identify IoT devices by using a reader device, e.g., a smart phone. When a unique device identity is read it is transmitted from the phone using either Wi-Fi or its cellular interface to a server. This server then looks the user menu based on the received identity and returns the GUI element to the smart phone. Even though this approach has been used for IoT it is useable for S.O.s, too.

7.2.4 SMART OBJECT IMPLEMENTATION

Defining an S.O. in a contextual universal manner has many benefits such as the possibility to standardize it. Standardization provides the possibility to define S.O. interfaces, defining its degree of context awareness and defining communication skills in a static manner. These benefits mean that it is possible to mass produce the S.O.s and thereby keep the cost low. A low cost is one of the major drivers for this technology to be accepted by the common consumer market as stated by Brush et al. (2011). They state four factors which are important, they are: Cost, flexibility, manageability and security.

Concerning the flexibility and security the concept of a universal S.O. actually solves these implicitly. Thus, it provides flexibility because it can be embedded into many different places such as a kitchen or in the living room without changing the S.O. In theory it is possible to simply use it in a plug-and-play fashion with some minor setup. Regarding the poor manageability the S.O. concept offers the possibility to use a portable user interface (e.g., a smart phone) to manage these. This interface offers a direct interaction with the S.O.s by using near field communication or it offers a more distant remote control by using its Internet connection capabilities. Thus, the S.O. concept facilitates eased manageability by spatial relating the controlled object directly to the controller or by allowing remote control from Internet based devices.

Looking into an architectural framework for distributed S.O.s only limited research in this area can be found. López et al. (2012) have suggested an S.O. architecture which covers the basic functionality as illustrated in Figure 7.2. However, they do not provide any details on how to implement the architecture in a single S.O. Trevenor et al. (2012) have presented a reduced smart home concept design by using an embedded controller and a radio module.

But, his work is missing the needed generality and flexibility, i.e., they solve a specific problem by designing a dedicated solution for this.

Using the basic architecture from López et al. (2012) and some of the embedded principles from Trevennor et al. (2012) in combination with the discussed software architecture it is possible to derive a general hardware architecture which supports home control and AI. The contained elements are:

- Input and output processing.
- General control logics.
- Event logic in the form of timers and a small calendar.
- Communication logics.
- Agents for AI processing.

These are discussed in the following and a block diagram covering this architecture is illustrated in Figure 7.9.

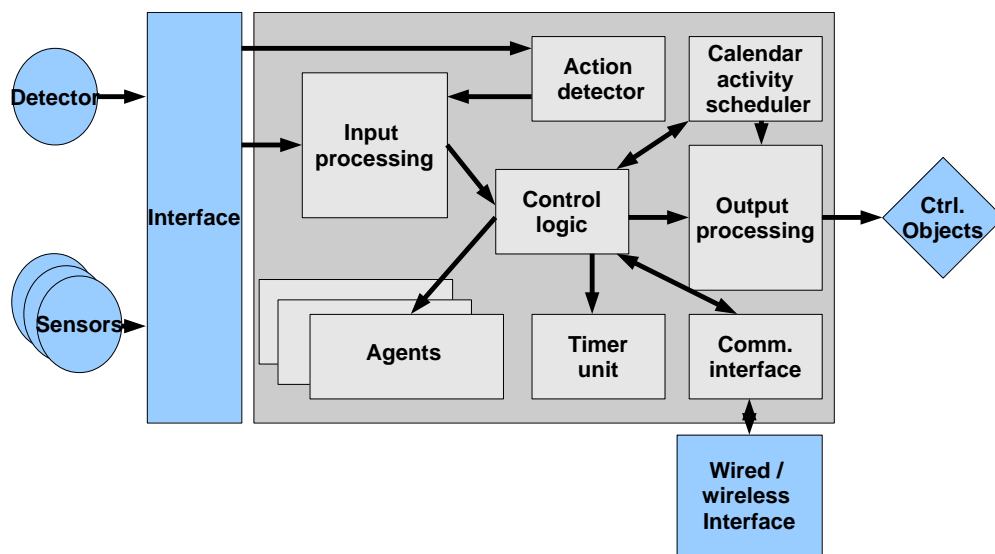


Figure 7.9. A general smart object with its interfaces and internal functionality.

Starting with the sensor signals (i.e., events) they are fed to the *input processing* unit. This unit time stamps the events and passes them through a digital filter. This filtering process is

able to suppress events coming from the same source in a short time interval relative to the system sampling time. Ignoring repeated events saves processing power and network load. It should be noted that dropping the repeating events coming in the same sampling interval normally does not waste information; however, this particular feature can be disabled. Another feature that is also available in the input processing unit is the *capture hold* function. It is able to save an incoming impulse event that is too short for the input sampling process to detect it (Section 4.4). Such a feature ensures that important very short time events are captured.

After input processing the events can be routed to three different devices. Firstly, they can be fed to the *calendar activity scheduler* unit that is able to start a pre-programmed user action e.g., by using its *output processing* unit. This unit drives and controls external objects like a table lamp. An example could be that the user sits close to the S.O. lamp so it detects the user presence by using its build-in *activity detector* unit. Because the user is present it makes a lookup in the calendar to find any pre-programmed matching action. If a match is found the action is performed. Next, the *timer unit* is searched for the same action. If a match is found the *timer unit* will control the action duration. An example could be to switch the lamp off after some time.

Secondly, it is possible to route the predicted action through a gateway in the *comm. interface* and thereby reach any device attached to the network. Examples could be other S.O.s or some user interface devices.

Thirdly, it is possible to route the events to the *agents unit*. This unit contains a collection of trainable agents based on the AI principle. When the trained agents consume the incoming events they will emit information in the form of detected actions. These actions can then trigger the *output processing* unit and thereby control external devices. It is also possible to route these detected actions through the *comm. interface* unit to other S.O.s or an HL-SHS for high-level processing. Seen in the light of the saving power and bandwidth this concept provides it by using agents that consume events so they are not transmitted at all. So, it seems reasonable to assume that this approach compared to the alternative approach of transmitting all the events will provide some power and network saving.

From an implementation point of view it is expected that some of these distributed S.O.s are implemented as small cheap embedded processing units that are powered by small batteries. Such a concept offers an easy and cheap installation procedure that positions the units in an out of sight manner meaning that the environment intrusion is minimized. This is in agreement with the discussed principles from Brush et al. (2011).

Saving processing power and network bandwidth are very important items to be considered as explained in section 4.4.

Implementing this architecture raises some important questions about designing and implementing the different units in a power efficient way. Most of the units can be found as optimized standard libraries in embedded processing source developer kits. But the agent implementation needs to be considered carefully because it involve complex and expensive computations.

7.3 COMMUNICATION BETWEEN SMART OBJECTS

The S.O.s and sensors contained in a smart home must be able to communicate with each other and the outside world. At the high OSI-layer level S.O.s need to exchange information such as setup information and predicted actions (Donninger & Lorenz, 2010). This is supported by the middleware layer in this work, as discussed in subsection 7.2.2. The lower layers, which are needed to support the higher layer information exchange, take care of the actual information transport through the physical media. A transport mechanism for generic device types raises some questions about implementation of a useable network topology model. Such a model must be flexible, i.e., it must be able to support many generic device types that are powered by either a battery or the mains (Shah et al., 2009). This model is explored in the following. A detailed description of the network topology model can be found in Lynggaard (2013a).

7.3.1 A SIMPLIFIED SCENARIO

To explore the network topology model and defining the needs for an S.O. communication system the simplified living room scenario model presented in subsection 4.3.2 (Figure 4.5) is used and reprinted in Figure 7.10 for convenience. In subsection 4.3.2 this model was used to

represent a centralized system with its event sequences given in Figure 4.6. In this context the same scenario is used to explore a distributed model which is achieved by transforming it into a distributed system that uses S.O.s.

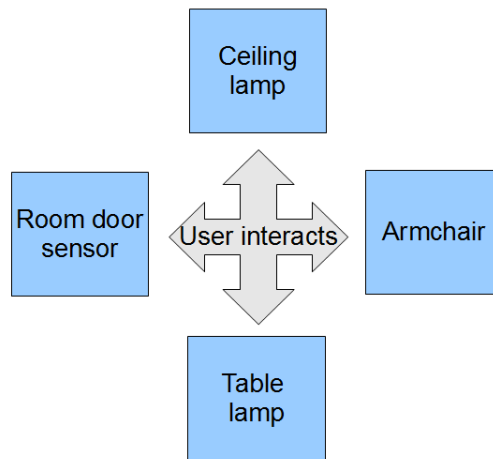


Figure 7.10. Living room - a simple smart home scenario model.

As discussed (Subsection 4.3.2), a general view could be to consider a scenario where the user enters the room, turns off the ceiling lamp, sits down in the armchair, turns on the table lamp and opens a book for reading. The S.O. based systems learn from these repeated actions over time. To be able to automate this behavior it is required that the ceiling and the table lamps are equipped with AI, i.e., they are S.O.s.

The events needed in this simple scenario are illustrated in Figure 7.11. The door (flow 1-2) and armchair sensors (flow 3-4) emit events to the table and ceiling lamps. Based on these events and the local time the table lamp (i.e., it's S.O.) decides to turn on (flow 6). This action emits an action (flow 8) to an external based SHS that handles the HL-SHS. Similarly, when the ceiling lamp S.O. receives the events (flow 2 and 4) it decides to turn off (flow 5) and transmits an action to the external based SHS (flow 7).

Even the simple scenario in Figure 7.11 produces a lot of events that need to be exchanged between the S.O.s. Comparing this sequence diagram with the centralized one shown in Figure 4.6 some notable differences are found.

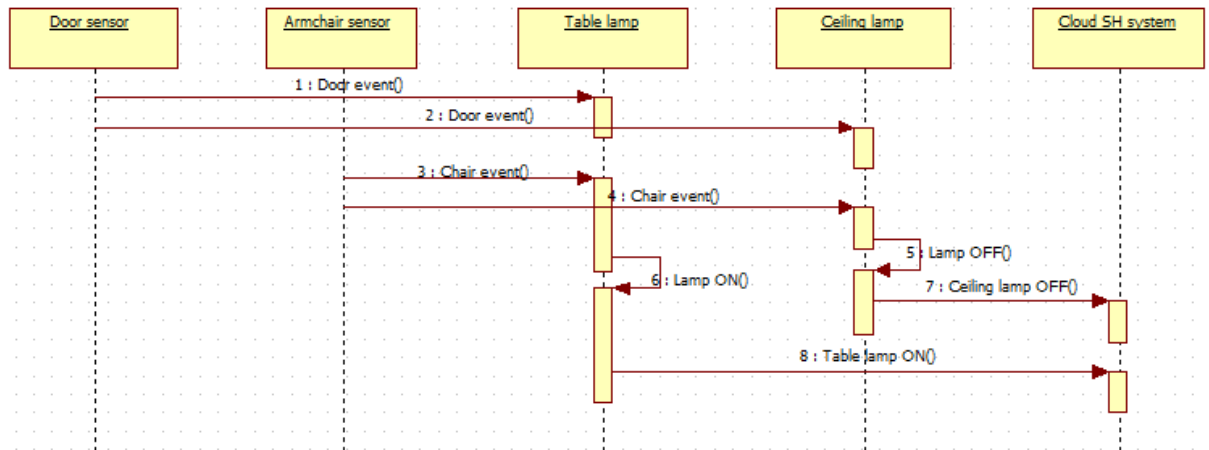


Figure 7.11. Sequence diagram for the living room scenario.

Firstly, in the centralized scenario (CS) sensor events are routed to a server which involves and loads many nodes. In the distributed scenario (DS) sensor events are distributed to the spatially nearby S.O.s. This approach saves network traffic and node load (i.e., power consumption) (Lynggaard, 2013c). Secondly, the CS is hard coupled, i.e., most nodes depend on each other and the centralized server. But, in the DS the dependency and coupling between the S.O.s are reduced. This means that the S.O.s are more independent of each other so rerouting of event messages between the S.O.s is avoided. In addition, it supports the concept of plug-and-play, i.e., different manufacturers are able to supply devices with simple sensor interfaces. Thirdly, the huge AI processing task allocated in the CS server is divided into distributed S.O.s that processes in parallel in the DS concept. This provides benefits in the form of reduced latency, reduced network load, and reduced failure severity.

In the just presented scenario the sensors (door and armchair) are both positioned at places where it is difficult and unpractical to provide wired mains power so these devices must be powered by other means such as batteries. However, the table and ceiling lamp S.O.s are powered by the mains. In fact it is expected that smart homes will have a lot of small easy to install battery powered sensors in the future (Alam et al., 2012).

7.3.2 TOP LEVEL COMMUNICATION MODEL

By combining domain knowledge with a generalization of the scenario presented in subsection 7.3.1 and illustrated in Figure 7.11 it is obvious that different device node types

and node capabilities are needed. These types include wired (Subsection 4.5.3) and wireless sensors (Subsection 4.5.4), wired actuators, and an S.O. based core network as illustrated in Figure 7.12.

Some research has been performed in the concept area of different device types. At the high protocol level Eisenhauer et al. (2009) have suggested the HYDRA middleware platform that integrates physical layer standards into one unit (Subsection 7.2.2). But HYDRA does not solve the problems on the physical layer such as the power consumption problem. Today's research is mostly focused on concepts which involve a centralized server (Zhao, 2010), (Rathnayaka et al., 2012), (Mao et al., 2010), (Section 4.3). Nonetheless, this concept does not provide the physical layer benefits found in the suggested distributed system as discussed in the following.

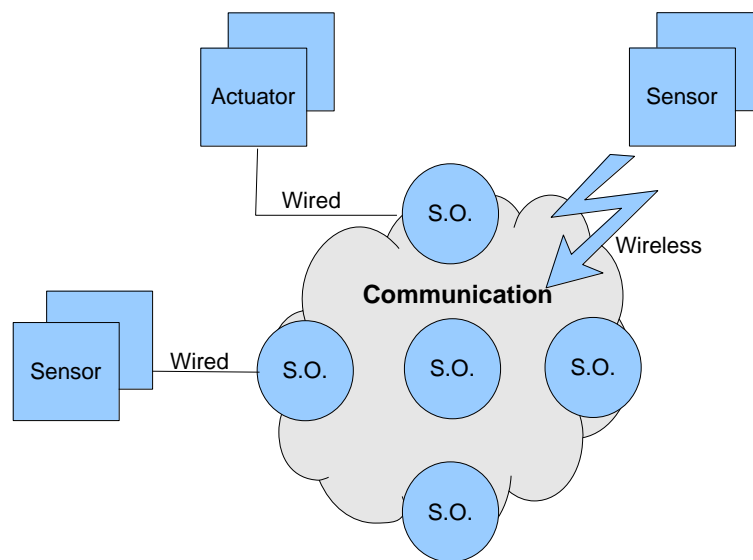


Figure 7.12. A top level smart home communication model.

Wired connections are used to connect specific sensors to specific S.O.s as illustrated in the suggested distributed system Figure 7.12. These connections provide power to the sensors and they also carry sensor communication. Alternatively, if the S.O.s are powered by batteries the connected sensors draw power from these, and consequently the sensors must have a low power consumption. However, this is normally the case with wired sensors because they often are simple switches and electronic elements such as heat sensitive resistors and semiconductors, etc. - see subsection 4.4.2. Another challenge is the limited bit-rate provided

by wired connections (Subsection 4.5.3), i.e., wired systems such as X10 provide only 20 bits/s. Nonetheless, this is a minor problem in this work because the sensors only emit a very limited amount of binary information (Subsection 4.4.1).

The wired actuator in Figure 7.12 symbolizes a device controlled by the connected S.O.s. These S.O.s are always powered by the mains grid as are most actuators.

The wireless sensors Figure 7.12 can be powered by the mains depending on their employment in the application, but usually they are battery powered (Subsection 4.4.2). So from a usability point of view they have to optimize their battery lifetime. In most sensors the major part of the current consumption is from the wireless transceiver interface (Zhao et al., 2013), (Shi, 2007). When the transmitter is on it consumes much power, but this is only the case whenever an event is transmitted. In addition, the receiver is the most complicated and expensive device in a transceiver. This means that cost, power and density savings can be achieved by leaving out the receiver (Zhao et al., 2013). However, leaving out the receiver raises problems about missing protocol handshakes; this issue is discussed later (Section 7.3.5).

7.3.3 THE SMART HOME CORE

As noted, a group of S.O.s are present in the center of Figure 7.12. These S.O.s are able to communicate with each other by using a well-defined protocol and together they constitute an SHN. Such a network is able to communicate with external systems like user interface devices and cloud based services. This S.O. group is the backbone or the core of the presented SHS. They are preferably powered by the mains grid, so they have the necessary resources for collecting sensor events, perform data processing, control actuators and communicate with external systems. However, to provide the flexibility for clustering the S.O.s (Subsection 7.3.4) these can also be battery powered.

The challenge with battery power consumption in the S.O.s used in the IoT context is discussed by López et al. (2012). They propose an S.O. framework based on a battery powered cluster head (i.e., master / slave node) concept. In addition, they suggest a complex scheme that uses battery power to rotate the cluster head between the different S.O.s

(Subsection 4.4.4). Another work by Starsinic (2010) also discusses the limited battery resource problem. He suggests a common S.O. gateway solution for saving battery power.

To some extent, this smart home core is an extension of work presented by Xu et al. (2013) where they present a WSN based mains powered core and battery powered sensors. Their concept requires a tree like structure with one final gateway node. This restriction is not found in this work because it uses a distributed S.O. approach. Furthermore, their concept requires a receiver in all the nodes including all the sensor nodes. As will be discussed later (Section 7.3.5) this restriction does not imply on this work.

The presented SHS needs to handle events from the battery powered sensors as discussed earlier. These events must be distributed to all the S.O.s so their contained AI circuits can be trained (see example in Figure 7.11). Thus, an event distribution mechanism must be defined.

To elaborate over the event distribution mechanism an example is explored. As illustrated in Figure 7.13 a wireless sensor transmits an event to the S.O. core. This event is captured by the gateway S.O.s that are able to bridge between the protocol used by the sensor and the one used by the S.O.s core network. The gateway S.O. then broadcast the event to all the S.O.s in the defined cluster. This method distributed the events between the S.O.s, but it has a downside. If more gateway S.O.s are able to perform the bridging they will follow the same procedure, i.e., multiple event flooding is likely to occur (Subsection 4.5.5).

To handle the multi-event flooding situation this work suggests that a random delay is added to all the S.O.s before they broadcast events. So, when an S.O. receives an event it waits a short random time (less than half a second) before it looks into its own event buffer to see if the event is already there, because another S.O. has already broadcasted it (i.e., the other S.O. has drawn a shorter time from its random time generator). If it is not available in the event buffer the S.O. broadcasts it.

Actually the principle of waiting a random time before accessing a shared resource is commonly known and used in schemes such as Carrier Sense Multiple Access with Collision Detection (CSMA-CD), however, in this work it has been applied to reduce event flooding. An alternative method would be to use the Gossiping algorithm defined and used in WSN

context (Hedetniemi & Liestman, 1988) (Subsection 4.5.5.2 and 4.5.5). It relies on sending the event to only one randomly chosen neighbor node. This node then performs the same thing iteratively whereby the event is distributed. But, because there are more possible sources in the presented system, flooding is not avoided by using this procedure. Other often used protocols are SPIN and LEACH which are discussed in subsection 4.5.5, (Kulik et al., 2002), (Heinzelman et al., 1999). The SPIN algorithm is not useable because it is based on selective distribution of broadcast events. But in this work all core S.O.s need the sensor events, so such a mechanism is unnecessary overhead which wastes resources. Similarly the LEACH algorithm is a point-to-point algorithm, which is unusable for broadcast.

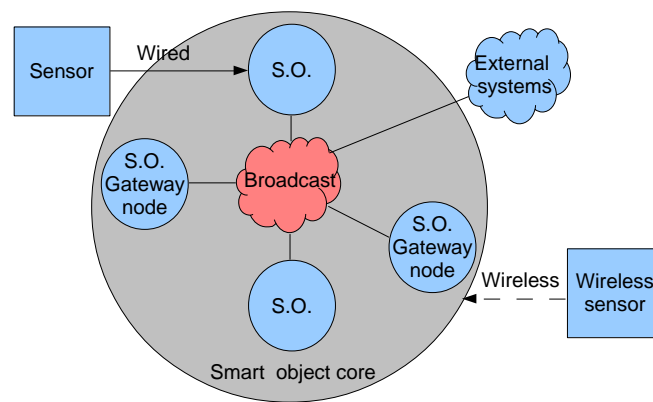


Figure 7.13. Sensor and S.O. event communication model.

Event flooding from the wired sensors in Figure 7.13 does not occur, because they are connected to one particular S.O. This means that the connected S.O. handles the event broadcast from the connected sensor to the other S.O.s and no duality therefore occurs, i.e., flooding is avoided in this case.

7.3.4 SENSOR AND SMART OBJECT CLUSTERING

The reasons for using spatial cell structures are to cluster common functionality, which distributes the load between many nodes; and to enable the use of power control and reduce interferences. To visualize its benefits in smart homes the CASAS smart environment test-bed, used by Cook (2012) in some of her experiments, is illustrated in Figure 7.14.



Figure 7.14. Bosch and Cairo test-bed smart environments used by Cook (2012). The encircled sensor groups are the opportunities for sensor cluster-tree grouping. The small red dots are the smart home sensors.

The potential sensor groups allocation are presented and highlighted as red circles which overlay the CASAS smart homes in Figure 7.14. A similar grouping pattern can be made in the test homes settings used by Kasteren et al. (2008a) in their smart home related work (Section 5.3), (Subsection 5.6.2). So this work defines an SHS where the sensors and S.O.s in each individual room are clustered into one unit as illustrated in Figure 7.14.

Viewing the defined cluster based S.O. and sensor topology from an interference and power consumption point of view underlines its usability. In a radio based system many types of interferences are in play. They are described in details in chapter 4.

The primary reason for clustering the devices in a room based scheme is that these devices (i.e., S.O.s) normally are related. So, e.g., kitchen sensors such as cupboard, fridge and table sensors do not provide any information on how to control the S.O.s in the living room. However, S.O.s in the same room are related so they only need to broadcast a sensor event to the members of that particular room cluster. This means that the transmission power can be kept at a minimum, i.e., battery power is saved and the amount of generated interferences is minimized. Regarding the event routing algorithms inside a cluster it is found that a simple flooding scheme works well, because the number of sensors is limited (Subsection 4.5.5). Alternatively, the SPIN algorithm could be used (Subsection 4.5.5). It is able to move data between S.O. nodes in a structured way, because it uses a handshake based approach. A work

performed by Dominici et al. (2010) explores the possibility to use physically based routing compared to the logically based approach used in most research projects. They found that the physically based approach simplifies information handling and processing. However, they only looked into a few dedicated examples and did not discuss this in the broader S.O. context.

A focus on inter-cluster communication is needed. Beside the flooding based S.O. core event distribution mechanism a second mechanism is needed to collect and transmit S.O. predicted actions to the external systems, such as the HL-SHS. This mechanism needs a cluster head to take care of all the interactions between the S.O. core and the external system.

In this work each cluster is managed by only one cluster head analogue to the WSN LEACH algorithm (Heinzelman et al., 1999). This cluster head handles the data collected from cluster members and the exchange of data out of the cluster context. Furthermore, it creates a TDMA like structure where each node has an assigned time slot. Such a structure is beneficial in the case of using transmit only nodes (Subsection 4.4.3).

Without the presented room based clustering mechanism an unorganized cluster head allocation could take place. This normally happens when using standard protocols such as ZigBee and 6LoWPAN. Thus, it is possible that the cluster head could be allocated outside the room. This means that battery powered sensors must use a high power level to penetrate walls with the cost of reduced battery lifetime (Lynggaard, 2013a), (Bleda et al., 2012). In addition, a higher power level also means that a higher destructive interference level is imposed to the neighbor S.O.s.

In another scenario more than one cluster head candidate could be available, but some of these candidates are blocked by a transmitting WiFi node or a micro oven. If one of these blocked nodes is chosen as the cluster head, communicating nodes need to use more power to overcome this blocking. So, being able to choose the right cluster head inside a room is important.

To be able to choose a cluster head among the S.O. core candidates, the WSN routing framework can be used in an analogue manner. Regarding the SPIN algorithm it is dedicated

to disseminate observations between network nodes and thereby do not provide the wanted functionality (Heinzelman et al., 1999), (Subsection 4.5.5). Regarding the LEACH algorithm it is able to collect and deliver data to an external data sink. The key idea in the LEACH algorithm is to choose the cluster head randomly using a cyclic approach (Subsection 4.5.5). Xu et al. (2013) modified this algorithm and named it LEACH-PI. Their modified approach ensures that the cluster head is always selected in the groups of mains powered nodes. This concept fits very well to the discussed S.O. core concept.

The main disadvantage in a concept where each room needs a cluster head is a lack of freedom as to where to place the devices. This means that some manual clusters setup process must be expected.

Many networking technologies can be used to support the discussed S.O. cluster model. Two good candidates are 6LoWPAN and ZigBee (Subsection 4.5.6). They provide similar technologies (Gratton, 2007). In the light of technology similarities only the ZigBee technology will be discussed; however, a change to Z-Wave or 6LoWPAN technologies is straight forward.

The ZigBee protocol stack uses the IEEE 802.15.4 WPAN standard (Subsection 4.5.6). This standard defines three node operating modes: PAN coordinator, Full Function Device (FFD) and Reduced Function Device (RFD). In addition, it offers a set of possible network topologies, where the cluster-tree is the most suited smart home topology Figure 7.15 as discussed earlier.



Figure 7.15. ZigBee cluster-tree topology (Kim et al., 2008) .

The PAN coordinator forms the root of the network. There can only be one of these nodes in each network. The FFD's act as coordinators and provide synchronization services to the other devices and coordinators. Any of these nodes can be the PAN coordinator. The RFD's may connect to a cluster-tree network as leave nodes at the end of branches.

In the context of S.O.s the FFD nodes belong to the S.O. core, i.e., they are mains powered nodes whereas the RFD is assigned to the battery powered S.O.s and sensors. This setup ensures that only the FFD's will handle routing, become cluster head and handle other power consuming functions. By allocating the RFD's to the battery powered devices these will be able to power sleep, i.e., save battery power.

Similar ZigBee setups have been used in some centralized smart home topologies, especially in remote controlled automated homes. However, a room based cluster topology has not been seen in the context of distributed smart homes, which is essential for this work. To ensure that the clusters are created using the room based approach, the choice made by the RFD's to one particular cluster head can be locked, i.e., some manual setup is needed when the S.O. node is installed.

7.3.5 THE TRANSMIT-ONLY SENSORS

A problem in using ZigBee and many other networks suitable for smart homes is that they do not incorporate transmit-only sensors (Subsection 4.4.3). As discussed earlier these sensors

are very important seen in the light of battery power saving. So, incorporating these into the ZigBee framework would be beneficial.

The main problem with transmit-only sensors is that they do not have a receiver and thereby cannot participate in the beacon based TDMA scheme normally used in ZigBee and other networks. This means that when these transmit-only sensors transmit at a random time (triggered by some user activity) they will destroy the TDMA frame structure. Most likely it causes FFD node communication losses and loss of the sensor events therefore incorporation of transmit-only sensors in a ZigBee based SHN requires some considerations.

Firstly, the transmit only sensor cannot sense if a frame collision has occurred, so to increase the probability that these frames are received they need to be transmitted a specified number of random times. However, this means more power spending, but Sudhaakar et al. (2009) have shown that even with this increase the benefits of power saving, lower device costs and complexity are still intact. They also showed that a delivery probability of 95 percent can be achieved by using four transmission times of the same packet under different test settings. These settings were randomly changed in an office with furniture and a WiFi router.

A second problem is that the TDMA based beacon structure used by ZigBee is sensitive to frame collisions because it destroys the synchronization information. This information is used by all the nodes to synchronize their particular TDMA time slices. To avoid this loss of synchronization information a non-beacon enabled transmission can be used. ZigBee supports such a mode where the communication is based on an un-slotted CSMA-CD scheme instead. Using this ZigBee mode in combination with transmit-only sensors means that frame collisions are likely to occur, but frame synchronization is not lost. In this case the ZigBee nodes will simply retransmit, whereas transmit-only sensors will most likely lose their transmissions. However, as discussed it will be re-transmitted a number of random times, i.e., a high reception probability is achieved.

In some cases both colliding frames would be lost. But a scheme defined by Blaszczyzyn et al. (2008) handles this situation by letting the receiver prioritize, which packet it will receive. Thus, the receiver measures the receiver signal strength indication for each node with which it

communicates. Using this value a packet with a high receiver signal strength indication value is allowed to interrupt a weaker one if the co-channel interference is acceptable.

From a future perspective the use of transmit-only sensors means that they need to be integrated with the WSN nodes (in this work ZigBee nodes). An alternative way to combine ZigBee and transmit-only-sensor nodes is by noting that ZigBee uses a Code Division Multiple Access (CDMA) based scheme in the physical layer. This means that it uses Direct Sequence Spread Spectrum (DSSS) modulated with phase shift keying. Because of the used spreading key mechanism the modulated signals are highly uncorrelated, i.e., more users can use the same channel resource. So, modulating the transmit-only sensors with the same modulation means that it can coexist with the ZigBee signals. The cost for using this scheme is a lower signal to noise ratio for the employed receivers, but this is acceptable in most cases. However, these nodes need to be able to handle this combined signal. A way to implement this is by using some amount of processing power to implement a SDR (Tribble, 2008) that is able to handle a ZigBee signal overlaid by a transmit only sensor signal. Most common ZigBee hardware nodes seen today (2013) do not provide the needed amount of processing power. However, the mains supplied S.O. core that is part of this work has plenty of processing power and is able to do this.

7.4 PROBLEMS EVALUATED

The evaluation and performance of the developed SHS is discussed in the following. It is performed by looking into three refined sub-questions that are derived from the research questions:

1. How does the SHS perform compared with the ASH and CSH systems?
2. Is the derived S.O. concept advantageous compared to existing agent systems?
3. Are the models implementable with existing technologies and devices?

Sub-question one focuses on the performance of the suggested SHS. By relating the suggested SHS with the ASH and CSH layers discussed in Table 6.1 to Table 6.3 the advantages and disadvantages of the suggested system is revealed. Table 7.1 compares the challenges found

in the ASH and CSH systems with the suggested SHS. Thus mapping the SHS context as a function of the ASH problem number yields:

ASH problem	Problem in SHS context
1	<u>Bandwidth usage.</u> The S.O.s contain a collection of intelligent agents that are connected to the sensors and home devices either wired or by using a short wireless connection. This means that the needed bandwidth for sensor information is reduced. Furthermore, the communication between the S.O.s and the management system is lowered considerably because the AI is positioned in the S.O.s and only actions are transmitted (Section 5.7).
2	<u>Channel blocking.</u> Many S.O.s must be expected to coexists, however, they only transmit when an action has been predicted. This lowers the co-channel and adjacent channel interference considerably (Section 4.5). It is expected that many of the wireless sensors transmit and thereby use the channel, but they are positioned close to the S.O. so the transmitting power can be kept low. This reduces co-channel and adjacent channel blocking.
3	<u>Insufficient device data performance.</u> Because the amount of traffic between the S.O. nodes in the SHN is lowered this problem is also lowered (see this table point one).
4	<u>Lack of power resources (battery).</u> The primary factor that uses power in a sensor or device node is the transceiver. Because the distance between the sensors and the S.O.s are lowered less power is needed for the sensors to transmit. Power is also reduced for the S.O.s because they primarily transmit when an action is predicted (Subsection 4.4.2). Additionally, the wired sensors draw power from the S.O. which can be mains powered.

Table 7.1. Suggested SHS system compared with the ASH system (*physical layer vs. reasoning layer*).

Summing up the analysis and results presented in Table 7.1 it is found that the SHN load is reduced and the power consumption in the sensors and S.O.s also are lowered considerably.

The position of the reasoning layer is a problem, which was discussed in the comparison of the ASH and the CSH systems (Table 6.2). By analyzing this result in the context of the developed SHS Table 7.2 has been derived. Hence, mapping the SHS context as a function of ASH problem number yields:

ASH problem	Problem in SHS context
1	<u>One processing server can be a bottleneck.</u> Since the processing is distributed to the local agents with their own processing resources this problem is solved.
2	<u>Single point of failure if the system is based on only one central server.</u> The distribution of processing resources solves this. However, if the user management system is allocated on a cloud server and its connection fails user management is not possible. However, the S.O. learned behavior still works and it is possible to manually control the devices.
3	<u>Central server needs to be always running.</u> Hibernating features can be implemented at S.O. level because most of the out of context S.O.s will not be affected by the sensor events produced, when the user interacts with the smart home.

Table 7.2. The developed SHS system in ASH context (position of the *reasoning layer*).

Analyzing the presentation layer in the context of the *agents layer* in the ASH model against the SHS developed model provides Table 7.3. Thus, mapping the SHS as a function of the ASH problem number yields:

CSH problem	Problem in SHS context
1	<u>The user interface application runs on the same server as the S.O. (<i>agents layer</i>) calculation.</u> In the SHS the <i>user management</i> system is executed as a cloud service. This cloud service has the needed amount of processing power. However, some overhead must be expected to interface the user interface to the individual S.O.s.
2	<u>Single point of failure on a centralized server will cause the user interface to fail.</u> In the SHS system the user interface runs as a cloud service this service is expected to have backup strategies. Whereas, if it is executed locally on a server and it fail the user interface will not work.
3	<u>Data exchange from either the S.O. (<i>agents layer</i>) to the user management (<i>presentation layer</i>) loads the network and its performance.</u> Because the S.O. is antonymous it only needs the user management interface when it is updated. So, the load is proportional with the user management activity and only load when this is performed.
4	<u>There is no one-to-one connection between a hand held “smart” device (e.g., a smart phone) and the related smart home device.</u> The SHS concept offers a one-to-one connection because each S.O. contains a RFID tag with the needed information. So, by spatial approximate a portable user interface device to this RFID tag opens the particular management interface for this S.O.

Table 7.3. The developed SHS system in an ASH context (user interface vs. S.O. layer).

From the discussion included in tables (Table 7.1 to Table 7.3) it is found that almost all the problems discussed in the ASH and CSH contexts are solved by the SHS system. However, the choice of using a centralized user management system running on a cloud server as a service means that problems like “single point of failure” and S.O. management traffic are with some drawbacks. However, these drawbacks are manageable by adding redundancy at the cloud service. Many cloud services today (2013) use server farms, so if one server fails it is preempted and another schedules the job. In case the Internet connection to the smart home is interrupted it is still possible to get S.O. based services.

Second sub-question looks into if the derived S.O. concept is advantageous compared to the existing agent systems.

One of the primary advantages in the suggested SHS is its compatibility with the future development of the Internet heading against the Internet of things (IoT) (EU_Commission, 2009), (Bandyopadhyay & Sen, 2011). Thus, as stated by López et al. (2012):

...the IoT vision: unique and automatic identification, the sensed condition of objects, embedded processing for local intelligence and autonomy, object-to-object networking, and an Internet-based information infrastructure (López et al., 2012).

As presented earlier a similar vision about the future is stated by Alam et al. (2012), which is supported by Zhang et al. (2012). The statement by Alam et al. is:

The trends indicate the increasing popularity of using middleware to integrate heterogeneous devices. Because multivendor devices will coexist in future, the use of middleware is an efficient solution to create networks that will overcome the limitations of diverse device integration. It seems that home intelligence will be employed in a distributed manner. This distributed intelligence may be applied in the form of smart devices (Alam et al., 2012).

So, a distributed agent-based system benefits from being able to sense the context, has communication ability to other objects and the Internet. In addition, it must be based on a middleware concept that offers the possibility to integrate heterogeneous devices and provide processing power, i.e., it must be able to handle local processing tasks, such as AI algorithms.

Actually, the suggested SHS offers these elements, why it is both IoT ready and fits into the future vision stated by these researchers. As stated earlier (Section 4.3) most researched agent-based systems are more proprietary and they do not fulfill all of these requirements.

Furthermore, this S.O. concept is advantageous in its vision of keeping power consumption and network bandwidth low. Network bandwidth is important because it indirectly relates to power consumption and interferences as discussed in (Subsection 4.4.2). Power consumption is important for increasing the S.O. usability, i.e., reduce the change-rate of batteries. It is also important because the technology level of today (2013) offers different solutions of power supply possibilities such as solar-cells and energy harvesting technologies (Graham et al., 2011), (Boegel, 2012), (Lee et al., 2011). Thus, various reasons exist for focusing on low power consumption. In the SHS system different means have been used for lowering the power consumption, such as S.O. clustering and transmit-only sensors, etc. The use of a clustering approach provides benefits in the form of lower interference level, lower power consumption and it decouples devices by allocating them to a cluster head. This cluster head takes care of communication out of the cluster context, which means that the cluster member devices can be low resource devices as well as more complex devices such as an S.O.

In the future it is likely that S.O. like distributed processing units in the form of System on a Chip (SoC) will be offered seen in the light of IoT. They will provide a small form factor, processing resources and probably incorporate the concept of swarm processing.

Third sub-question focuses on the possibility to implement the model with existing technologies and devices. Because the SHS has been derived with this in focus the architectural structure of an S.O. Figure 7.9 adapts what is possible on low-power microcontrollers of today (2013). Finding the edge of what is possible resolves into factors such as cost, power consumption, processing power, and peripheral interface on the microcontrollers and transceivers. Trevennor et al. (2012) implement elements of a smart home wireless node on the small embedded processor ATmega328. They control colored leds, door switches and wireless transmissions without any real-time problems. Another paper by Chong et al. (2011) implements a software layered model with comparable load to a middleware layer on a low-power 8051 microcontroller. They also implemented a gateway between ZigBee and a wired Serial Parallel Interface (SPI) on the same microcontroller. Their

radio system was based on the ZigBee protocol by using a CC2530 module from Texas Instruments. This module has been used in many battery powered home control applications. However, its power consumption in relation to processor speed needs to be optimized (Subsection 4.4.5).

Beside the S.O.s the SHS consists of the HL-SHS that are allocated on either cloud servers or locally in the smart home on processing platforms that offer the needed resources. Thus, these elements are not restricted by today's technology level. So conclusive, most of the needed technology is available for realizing the presented SHS concept.

In addition, the development is going fast developing Moore's law that states a doubling of processing power each 1.8 year. To perspective this statement, (Kawsar et al., 2009) states that:

One of the consequences of pervasive technologies (e.g., miniaturization of the computer technologies and proliferation of wireless Internet, short-range radio connectivity, etc.) is the integration of processors and tiny sensors into everyday objects. This revolutionized our perception of computing. We are in an era, where we communicate directly with our belongings, e.g., watches, umbrella, clothes, furniture or shoes and they can also intercommunicate. These everyday objects are designed to provide supplementary services beyond the primary purpose, an initiative that has been denoted as S.O. (Kawsar et al., 2009).

Thus S.O. will most likely be integrated into the area of IoT in the future using technologies such as energy harvesting, low-power SDR radios, and SoC.

7.5 SUMMARY

The derivation of a distributed system architecture was performed and explored in the light of centralized and agent-based systems. These systems are combined with knowledge from the IoT research area and the design criterions derived by exploring related research in the area.

The most essential element in the derived distributed system architecture is the S.O. model. Its architecture and design were explored in terms of middleware, S.O. communication, and

sensors and actuators. Additionally, an implementation model for the S.O.s was suggested and presented. It includes elements such as interfaces, internal functionality, and ability to be implemented on small low-power (battery) microprocessors. The S.O.s form a smart home core model which was derived together with its communication with multiple sensors.

To deal with wireless interferences and lower power consumption a clustering model based on ZigBee was suggested. This model uses a concept with fixed cluster headers and it supports both battery and mains powered nodes. In addition, it includes a concept that uses transmit-only sensors.

8 ENHANCED FRAMEWORK USING ARTIFICIAL INTELLIGENCE

The mathematical AI framework for the smart home system architecture is derived in this chapter, based on the background presented in chapter 5. The derivation takes the form of a mathematical formulation (Section 8.2) and provides two sub-models named High-level-Smart home System (HL-SHS) (Section 8.3) and Low-level Smart Home System (LL-SHS) (Section 8.5), which support and match the derived conceptual sub-models. Both models are simulated and their performances evaluated and discussed in sections 8.4 and 8.6. Finally, the S.O. (contained in LL-SHS) implementation model is presented (Section 8.7) and its performance evaluated and discussed (Section 8.8).

8.1 RESEARCH QUESTION AND AI FRAMEWORK

8.1.1 RELATION TO RESEARCH QUESTION

Research question two can be split into sub-problems that are useful for a detailed analysis and discussion, as stated in chapters 4 and 5. In addition, a perspective view is necessary for determining the derived sub-problems. As such, the view used in this thesis is the *technology* view, as discussed in chapter 6. This choice has a disadvantage in terms of a loss of generality, but this is the accepted cost when limiting a subject.

In the following, research problem two is used as a guide for subsequent work and thus, for convenience, it is restated here.

Question 2: *How should the AI be distributed to comply with the smart home system architecture?*

As stated, this research question deals with the distribution of the *AI* framework to comply with the derived *smart home system architecture*. In addition, it implicitly involves S.O.s and the use of existing technologies (Section 1.3). These subjects are the focal points in the following section. In addition, background knowledge for these subjects is provided in chapters 4 and 5.

The goal is to derive an AI framework that is candidate for the distributed system architecture and which fulfill the requirements of the subjects discussed above. To accomplish this objective, a mathematical model is derived in the beginning of this section (Section 8.2) and this model is refined into the final AI model by using model knowledge, architecture, and limitations from the derived distributed system architecture (Chapter 7). Because the system architecture uses a distributed approach, the AI framework must also be distributed to achieve an advantageous match (Alam et al., 2012). In addition, the hierarchical concept used by the system architecture, i.e., the LL-SHS and the HL-SHS, has an impact on the possible choices for the distributed AI framework architecture. Thus, two hierarchically based AI models will be derived; one for the HL-SHS (Section 8.3), and one for the LL-SHS (Section 8.5). Vital elements in these models are then simulated on real world datasets published by different universities that are researching this area (Sections 8.4 and 8.6). A comparison of the achieved learning and prediction results with comparable results from papers within this field reveals that the achieved simplified models have similar performance.

Furthermore, the limitations of the research question, given in section 1.3, state that the technology perspective is limited to currently existing technologies and devices. Thus, the derived models must have a complexity level that allows them to be executed on the LL-SHS and HL-SHS. Regarding the S.O.s contained in the LL-SHS, the AI framework needs to support small and cheap low-power microcontrollers that set the state-of-the-art for today (2013). In addition, it must support both the low power consumption and the low network load that is expected by the system architecture elements. These issues are discussed in sections 8.7 and 8.8.

8.1.2 THE AI FRAMEWORK; AN OVERVIEW

Smart environments in the smart home area need to implement context-aware services that are able to deal with daily activities, such as grooming, eating, drinking, taking medicine and cooking. Today (2013), several researchers have designed many approaches and systems for modeling and recognizing users' actions (Rashidi et al., 2011), (Silva et al., 2012), (Chua et al., 2011). These systems must be able to interface with hundreds or even thousands of sensors (Cook, 2012). In addition, they need to be able to deal with voluminous and rich data, which is very challenging for the AI learning and prediction process (Dominici et al., 2010).

Generally, context-aware services are added to smart homes by using AI-based systems. These system need to be able to learn activities from users' behaviors, i.e., when the user move around and perform actions within the smart home. When these actions are learned, the system must be able to detect the "learning situation" with a high degree of probability and suggest it to the user or perform it autonomously. Actually, the "activity detection" part is the core problem of using AI in smart homes and many researchers have tackled this area (Silva et al., 2012), (Chua et al., 2011).

The SHS presented in this work is based on a distributed concept that involves a collection of S.O.s. These are arranged in a grid where they are able to communicate and to receive data from its nearby context, i.e., from connected sensors. This means that the AI processing part could be allocated on a centralized server or it could be distributed together with the S.O.s. Allocating it on a centralized server means that huge quantities of sensor data must be routed to this server and thereby, this loads the network. As discussed in subsection 6.1.1, a single point of network failure is likely to cause a breakdown or degrade the functionality. Another disadvantage is that the detected user activities normally take place in one room at a time, i.e., the sensor events are clustered in this room (Subsection 4.4.4). Dealing with clustered events in a centralized solution is suboptimal.

As discussed in section 4.3, using a distributed AI system has some disadvantages. Nonetheless, it is expected that future AI systems for smart homes will be distributed (Alam et al., 2012). Therefore, by combining the beneficial features of both the centralized and the distributed approaches, a new advantageous AI system will be provided. This system will be able to perform light AI calculations on the restricted processing resources offered by the S.O.s, while undertaking heavier ones on centralized devices e.g., implemented on PCs, smart TVs, or on a cloud server farm. An overview of this projected system is presented in Figure 8.1.

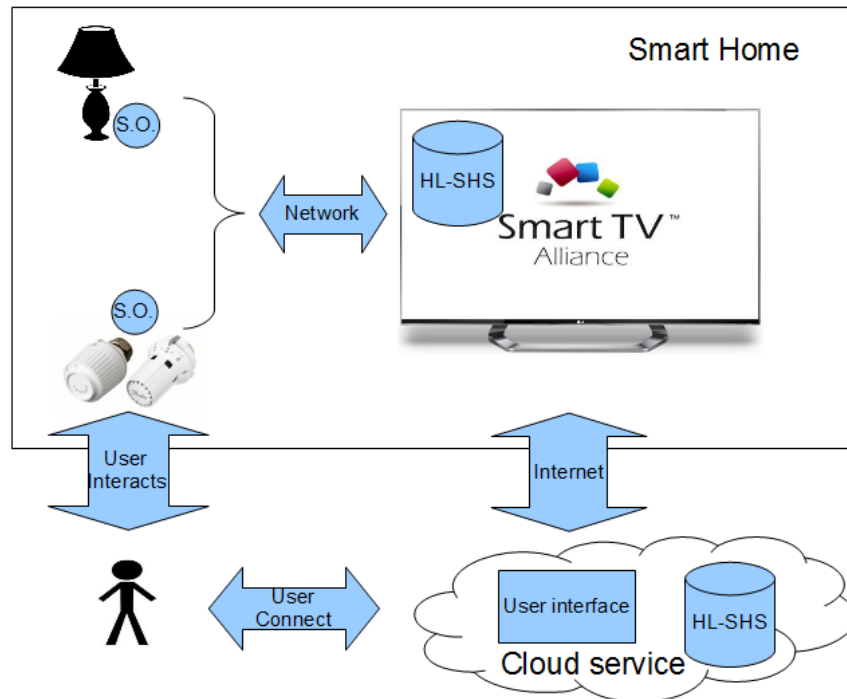


Figure 8.1. Overview of AI components in the smart home system.

When the user interacts with an S.O. through its sensors, such as tuning on a lamp (Figure 8.1), this behavior is learned by the lamp (LL-SHS) over time. Based on this learned behavior, the lamp works as a standalone device which possesses its own intelligence. However, it emits an action each time it detects a user activity. When this action arrives at the HL-SHS, it is buffered and processed. The HL-SHS uses these actions to learn and predict more complex user activities. Therefore, if a user activity pattern is detected, it emits control information to the actuators contained in the S.O.s, e.g., the thermostat could be informed to regulate the heat.

The cloud-based user interface (Figure 8.1) manages the smart home by connecting the user terminal to the logics. Thus, the user is able to control the S.O.s remotely and to set up its behavior. In addition it can contain the HL-SHS.

Using this concept offers benefits in the form of providing smart devices that work autonomously (like IoT), and it offers services to its users based on local context information. However, it also provides the more advanced HL-SHS that uses information from more than one S.O. and which is able to provide learning from temporal sequences of actions, i.e., from

activities. Because these systems work in a standalone mode and only exchange actions and actuator control information, some benefits are noted. Firstly, the network load is reduced because the S.O.s process all sensor events, i.e. they are not transmitted. Secondly, from a security perspective, no sensor information leaves the smart home. Thirdly, the HL-SHS is connected to the Internet, which means that backup and software updates can be managed easily from the cloud-based user interface.

8.2 THE SMART HOME MODEL; A MATHEMATICAL DERIVATION

Temporal probabilistic models provide a suitable framework that are optimal for dealing with the uncertainty of sensor data that reflects user activities, robust in a noisy environment, able to handle temporal sequential data, and that are well covered in terms of developed algorithms (Cook, 2012), (Naeem & Bigham, 2009), (Kasteren et al., 2008b). As stated in section 5.3, the most suited AI algorithms in smart homes are the naïve Bayes (NB) algorithm, the Hidden Markov Model (HMM) algorithm and the Conditional Random Field (CRF) algorithm (Cook, 2012), (Kasteren et al., 2008a). Thus, for this work, the HMM and the NB algorithms are explored, analyzed, and reworked to derive a final model that is useable within the distributed system architecture. This section is based on the mathematics and the models presented in chapter 5.

The formula covering the HMM has been defined earlier (Subsection 5.5.3), but is repeated here in a slightly changed form for convenience (Bishop, 2006):

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}, \theta) = p(y_1) \prod_{t=1}^T p(\hat{x}_t | y_t, \phi) \prod_{t=2}^T p(y_t | y_{t-1}, A) \quad (8.1)$$

where $p(y_1)$ is the initial state distribution representing the probability of starting in state y_1 . The parameter $\theta = \{\phi, A\}$ contains weights that govern the emission and transition conditional probability distributions.

First, the HMM formula (8.1) depends of absolute time, which is transformed into relative time because the sensor data event arrival time is only relevant inside one relative time step. This is conditioned by the Markov assumption (Kasteren et al., 2011). Removing absolute time has been done by rewriting it as a recursive formula (8.2). In addition, the removal of

$p(y_1)$ is performed by assuming all initial probabilities are equal. However, it is noted that these assumptions do not restrict the level of generality in smart homes. Applying these assumptions yields:

$$p(\hat{y}_q, \hat{x}_q, \theta) = p(\hat{x}_q | y_q, \phi) p(y_q | y_{q-1}, A) \quad (8.2)$$

where q is a specific time instance in time vector T . Drawing this equation for Q possible states of the latent variable y_q named $S_{1..Q}$ yields Figure 8.2.

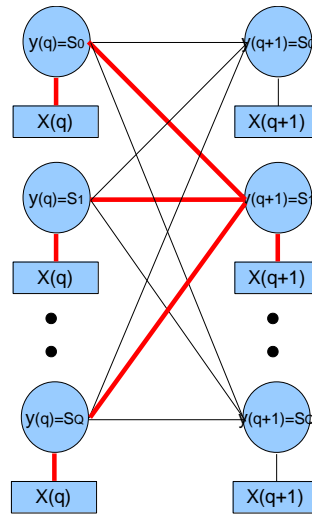


Figure 8.2. Trellis diagram for an HMM.

As illustrated in Figure 8.2 it is assumed that the HMM is in state S_1 at time $q+1$. Therefore, for maximizing its observation vector x_{q+1} , given by the observation probability $p(\hat{x}_{q+1} | y_{q+1} = S_1)$, it needs the transition probability from all the Q possible states i.e., $p(y_{q+1} = S_1 | y_q)$ at time instance q . As noted, this optimization process is costly in both memory and processing power, while it is not optimal for direct implementation in an embedded S.O.

Implementing an HMM model in each individual S.O. will not improve the recognition probability considerably compared to the simpler NB classifier, which performs equally well in this context (Subsection 5.6.2), (Fang & He, 2012). However, these NB classifiers cannot

be used for high-level prediction because they are not temporal, i.e., they do not include history in the form of correlated time steps, as the HMM does. An illustrative example is the user going from the bathroom (an action) to the bedroom (an action) in the late evening, which could mean that the user is going to get a night's sleep. In analyzing this with the NB classifier the two actions would be treated separately, i.e., they are handled as two independent actions only. However, the HMM classifier would use the temporal connections between these actions, which adds certainty to detecting the action sequence, i.e., the scenario. Thus, the detection probability for the NB classifier compared with the HMM classifier that uses both correlated actions is lower in some situations.

Equipping an S.O. with AI based on the NB and the HMM algorithms are possible by using a distributed approach, as explained in the following. Considering the HMM algorithm (8.1), it is clear that it can be combined with the NB algorithm as shown in (8.4). For convenience, the NB algorithm is reprinted in (8.3) and explained in subsection 5.5.2.

$$p(\hat{\gamma}_{1:T}, \hat{x}_{1:T}, \psi)_{NB} = p(\gamma_1) \prod_{\tau=1}^T p(\vec{x}_\tau | \gamma_\tau, \psi) \quad (8.3)$$

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}, \theta)_{HMM} = \prod_{t=1}^T \prod_{k=1}^K p(\vec{x}_{t,k} | \gamma_{t,k}, \psi_k) p(\gamma_{t,k} | y_t, \phi) p(y_t | y_{t-1}, A) \quad (8.4)$$

For simplicity all the initial probabilities are assumed equal and therefore, they are removed from the equations. Parameter $\theta = \{\psi, \phi, A\}$ represents weights governing the emission and transition conditional probability distributions and K is the number of NB classifiers. The other parameters are explained in subsection 5.5.3.

By choosing to restricts the HMM latent variable in (8.4) to have one state per observation variable, transforms it into a usable distributed model. This choice is in good agreement with the S.O. agent behavior because it allows the outcome to be binary; i.e., an action has been detected or not. In addition, this restriction simplifies the inference process in the HMM classifier, as discussed in the following section. From both this restriction and equation (8.4), it is noted that the derived HMM is expressed as a chain of separate functions. That is, one

NB classifier delivers its emission probability to one HMM state only, as illustrated in Figure 8.3.

The inference problem for the derived HMM model (8.4) consists of finding the single best state sequence that maximizes $p(\hat{y}_{1:T} | \hat{x}_{1:T}, \theta)_{HMM}$. By finding the probability for this best state sequence and then thresholding it against a pre-defined limit makes it possible to claim when either an action has been detected or not. It is noted that the NB term in (8.4) from an inferences point of view can be removed. This is so, because later in this work, the NB classifier emission probabilities are thresholded and quantified into the binary values of zero and one. These binary values are then user annotated and considered as the observable variable in the HMM model.

Therefore, the inference problem for the derived HMM model (8.4) reduces to the level normally found for the HMM's. Thus, by looking at the trellis diagram in Figure 8.2, the probability for being in state S_1 at time instance $q+1$ depends on the emission probability $p(\bar{x}_{q+1} | y_{q+1} = S_1)$ and the transmission probability $p(y_{q+1} = S_1 | y_q)$ for all the previous states at time instance q multiplied by their individual emission probabilities $p(\bar{x}_q | y_q)$. The trace of this path is highlighted (red lines) in Figure 8.2. More details can be found in subsection 5.5.3.

Based on the discussed assumptions, the derived equation (8.4) can be drawn (in three dimensions), as illustrated in Figure 8.3. The blue circles in Figure 8.3 are the NB classifiers and the red circles are the HMM classifier elements. It can be noted that the leftmost part illustrates the HMM states at time instance q and the rightmost part at time instance $q+1$, where q is a notation defining a particular time instance in the time vector $\{1..T\}$. Furthermore, it can be noted that the HMM states at time instance q and $q+1$ are connected by lines that illustrate the possible state transitions and also implicitly, the transmission probabilities.

Thus, using the discussed approaches provides the possibility for transforming the HMM equation into a distributed system. The next step is to map the S.O.s to this system, i.e., to

find an approach that minimizes the processing requirements for the S.O.s and at the same time supports a distributed usability level. Such a system is illustrated in Figure 8.4.

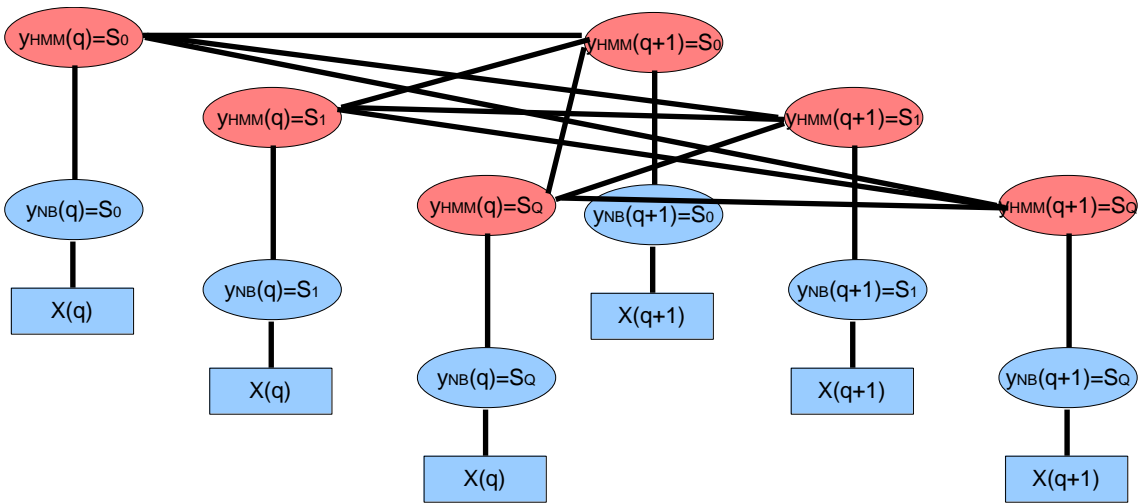


Figure 8.3. A suggested combined Naive Bayes and Hidden Markov model - 3D view.

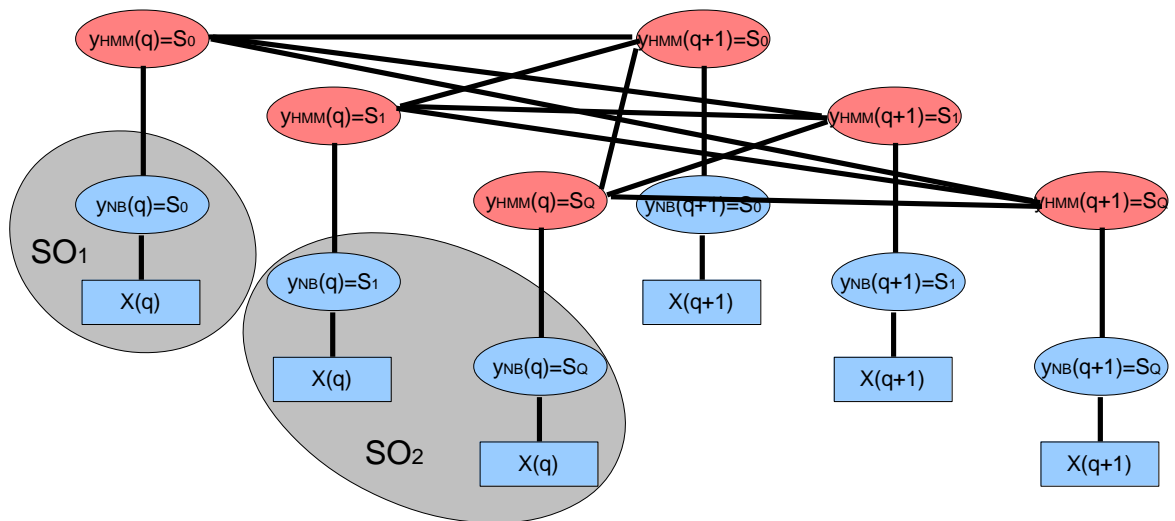


Figure 8.4. A suggested combined Naive Bayes and Hidden Markov model with Smart object allocation - 3D view.

Allocating the embedded S.O.s as illustrated in Figure 8.4 (dark grey area), reduces its processing load because each S.O. only deals with one or more contained binary NB classifier and not the far more complex HMM classifiers. Furthermore, as will be discussed in the

following section, these classifiers can be simplified further, i.e., even more processing power can be saved. An advantageous distributed usability level is also provided because the S.O.s are able to perform standalone processing independently of the HMM model, i.e., they are standalone NB classifiers. Hence, by using a binary NB classifier, it is possible to learn and predict actions from events in a standalone fashion. An example could be the control of a standalone device such as a table lamp based on learning and prediction without any connection to the top layer HMM (the red circles in Figure 8.4).

Describing the NB classifiers (blue circles) in Figure 8.4 from a mathematical point of view can be simplified by using prior domain knowledge. Thus, the purpose of the classifiers in an S.O. is to detect the presence of actions, but most of these actions are binary in nature, i.e., either a kitchen closet is open or closed, etc. This means that detecting one of the binary values also reveals the other because they are mutually exclusive. Therefore, using this prior knowledge means that a function that simplifies the NB classifier calculation burden can be defined. It is noted that by taking advantage of this mutually exclusive behaviour, the calculation burden is reduced by approximately 50 % without providing any significant loss in the action detection probability compared with comparable systems (Section 8.6).

The function that implements the simplified NB classifier is expressed in (8.5). This function, which is named an H-function, predicts the quantized action probability. This prediction and learning is based on a parameter set θ , the number of elements in vector x (i.e., the number of sensors) A , and a pre-defined threshold named ζ .

$$H(\vec{x}, \lambda, \theta) = \begin{cases} 1 & \text{if } \left[\prod_{\alpha=1}^A p(x^\alpha | \lambda, \theta) \right] > \zeta \\ 0 & \text{else} \end{cases} \quad (8.5)$$

As noted the H-function returns a binary result depending on the calculated probabilities and the threshold ζ . By calculating the final conditional probability over all the sensors in vector x and comparing this value to a threshold, either a one or a zero is returned, i.e., the action is predicted or not.

Therefore, each NB classifier in the suggested model of Figure 8.4 performs two tasks. First, it learns from local sensor events and then predicts actions based on this learning, as discussed earlier. Second, it transmits the predicted result (an action) to the top layer HMM classifier, which uses it for temporal learning and prediction.

When these actions arrive at the top layer HMM classifier, they are buffered because they cannot be expected to arrive simultaneously. Thus, when the user performs a scenario, sensors are triggered at different time instances, which cause delays between the arrived actions. These buffered actions are kept in a time-limited buffer, i.e., they are saved for some predefined time and thereafter deleted. Hence, if the actions are available within the buffer time slice (a window), the HMM classifiers use these to predict high-level actions.

Summing up, a distributed smart home AI system can be derived by splitting the HMM equation into different elements and by noticing how these fit into the derived system architecture. Thus, mapping the NB classifier onto the S.O.s agents provides local standalone AI. Similarly, it is possible to map the high-level HMM system as the AI part for the HL-SHS in the system architecture. These mappings are illustrated in Figure 8.5.

Another model that is used in smart home context to perform activity detection is the Conditional Random Field (CRF) model (Subsection 5.5.4). Even though it belongs to the discriminative model group it is similar to the HMM model regarding its mathematical structure (Lafferty et al., 2001), (Kasteren et al., 2008b). A rearranged CRF model is given in (8.6).

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}) = \frac{1}{Z(\hat{x}_{1:T})} \prod_{t=1}^T \exp\left(\sum_j \lambda_j t_j(y_t, y_{t-1}, \vec{x}_t) + \sum_k \mu_k s_k(y_t, \vec{x}_t)\right) \quad (8.6)$$

where the feature function $t_j(y_t, y_{t-1}, \vec{x}_t)$ is the transition function and the feature function $s_k(y_t, \vec{x}_t)$ is the observation function. The partition function $Z(\hat{x}_{1:T})$ normalizes the result into a valid probability range and λ_j and μ_k are the weight factors. Writing out the exponential terms yield (8.7).

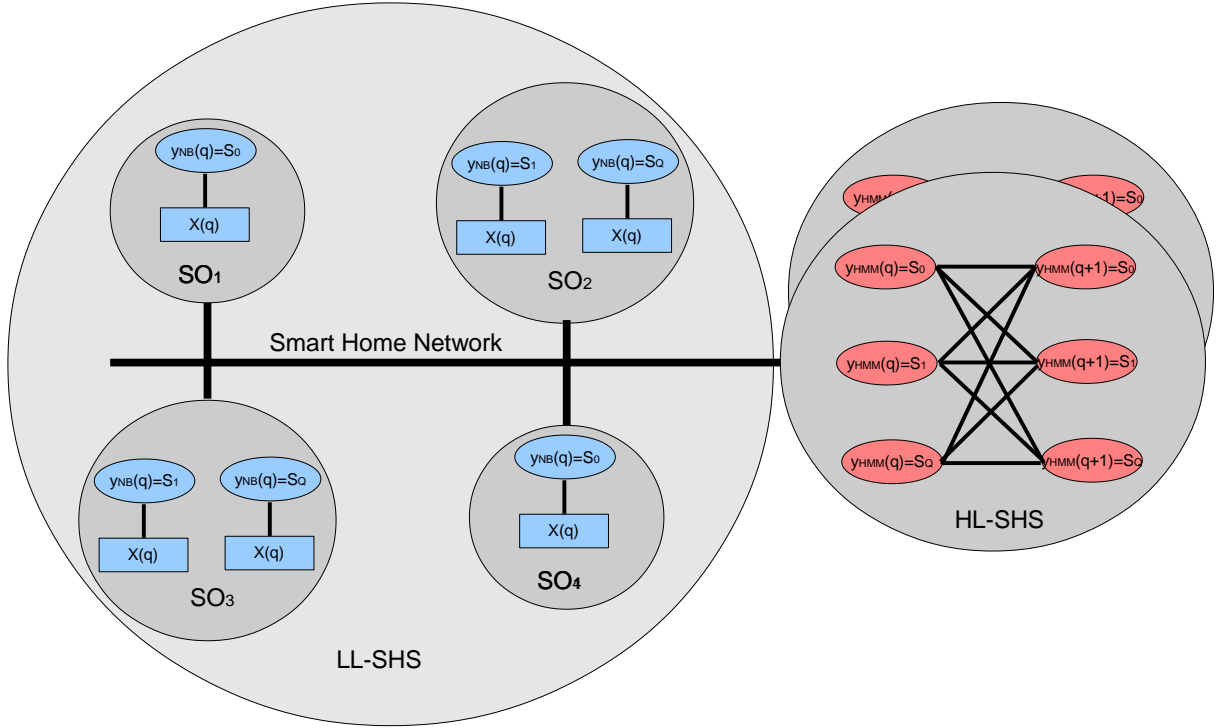


Figure 8.5. The suggested distributed smart home system. Smart Objects contains naïve Bayes classifiers connected to the Hidden Markov models.

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}) = \frac{1}{Z(\hat{x}_{1:T})} \prod_{t=1}^T \left[\prod_j \exp(\lambda_j t_j(y_t, y_{t-1}, \vec{x}_t)) \prod_k \exp(\mu_k s_k(y_t, \vec{x}_t)) \right] \quad (8.7)$$

By comparing equations (8.7) and (8.1) which cover the HMM classifier, their similarities are noticeable. This means that the distributed-HMM-based SHS can also be derived by using a CRF classifier. This derivation is not performed in this work; however, it is a straightforward process that can be achieved by following the principles used for the HMM classifier.

8.3 PROCESS MODEL OF HIGH-LEVEL SMART HOME SYSTEM

The HL-SHS models are described in this subsection. They comprise a process model describing its behavior, an implementation model, and two models that describe its learning and prediction principles. The content in this section has been elaborated in Lynggaard (2012b) and Lynggaard (2013b).

8.3.1 OBJECT PROCESS MODEL

To provide an overview of the HMM-based AI processing that takes place in the suggested HL-SHS and its required support blocks, an object process methodology model (Dori, 2002) is presented in Figure 8.6 and discussed in the following. It is noted that the coupling between the HL-SHS and LL-SHS system is described in Lynggaard (2012a).

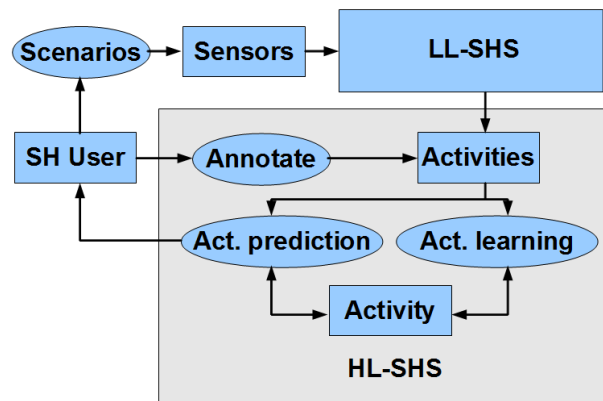


Figure 8.6 Object process model for the HL-SHS.

When the user (SH user object) performs scenarios in the form of a normal living pattern within a smart home (Scenarios process) sensors emit events (Sensors object). These events are consumed by the agents contained in the S.O.s that constitute the LL-SHS. The outcomes from these agents are predicted actions that are fed to the HL-SHS. When the actions arrive at the HL-SHS, they are annotated with a time stamp (Annotate process) and time buffered in a cyclic buffer (Activities process). At this point, the annotated actions are fed to the HMM-based learning (Act. learning process) and prediction parts (Act. prediction process). The activity learning process uses the content in the activity buffer to train the AI whenever the user provides a predefined activity. For example, when the user leaves home for a period of time, the light is set to the *nobody home state*. Thus, the activity learning process uses predefined activities to train the artificial network model positioned in the activity object (Activity object). Based on the trained artificial network, the activity prediction process evaluates the action sequences in the activity buffer to predict and suggest activities to the user.

8.3.2 IMPLEMENTATION MODEL

In general, from a mathematical point of view (Section 8.2), an SHS can be implemented as a collection of HMM classifiers in which the emission probabilities come from a collection of NB classifiers. However, this work uses a different approach, where the NB classifiers only emit the state of the latent variable and not the emission probability (Section 8.2). This approach lowers the transmission load on the backbone network, and save processing resources and power in the S.O.s because the emission probabilities do not need to be calculated and transmitted.

Equation (8.8) expresses the discussed concept. It has been derived by modifying equation (8.4) with equation (8.5).

$$G(t, k, \theta(t)) = \begin{cases} H(\bar{x}_{t,k}, \gamma_{t,k}, \psi_k(t)) p(\gamma_{t,k} | y_t, \phi(t)) p(y_t | y_{t-1}, A), & \text{if } H(\bar{x}_{t,k}, \gamma_{t,k}, \psi_k(t)) > 0 \\ 1 & \text{else} \end{cases} \quad (8.8)$$

$$p(\hat{y}_{1:T}, \hat{x}_{1:T}, \theta(t))_{HMM} = \prod_{t=1}^T \prod_{k=1}^K G(t, k, \theta(t))$$

By comparing equation (8.8) with equation (8.1) it can be noted that the NB emission term in (8.8) is a quantized version of the NB emission term in (8.1), i.e., the probability can only be one or zero depending on whether an NB classifier emits an action or not. From an entropy point of view, discarding information normally has the effect of reducing system predictability. However, in this work, the user is expected to either reject or accept the NB-predicted action and thereby, it provides the necessary ground proof.

It is noted that the training parameters are a function of time. This approach has been chosen because most humans follow the same habits every day; therefore a large amount of correlation information can be derived from incorporating time. Fang et al. (2012) found that using finely gridded time features increases the recognition rate considerably. Thus, their experiment, based on an NB classifier, reveals an improvement rate from 69 % to 82 % by changing the time granularity from one-day intervals to one-hour intervals. They found approximately the same improvement for the HMM classifier. However, the transition probability weights A in equation (8.8) are not a function of time. This choice enables the possibility of having a time difference between the latent variable outcomes y_t and y_{t-1} . This

allows two activities to happen at different times, even though they are correlated, which is often the case in home scenarios.

To illustrate the mapping of equation (8.8) into the SHS context, it has been rewritten into a recursive form and mapped into Figure 8.3 and the result is illustrated in Figure 8.7. Thus, the distributed S.O.s implement simplified versions of the NB equation, i.e., the H function as illustrated next to the NB classifier (blue objects, equation #1). Predictions from these NB classifiers are transmitted through the SHN as detected actions to the HL-SHS. The HL-SHS uses these actions as inputs for the implemented HMM classifiers (red objects). Therefore, the HMM equation has been implemented by multiplying it with the two last terms marked #2 and #3.

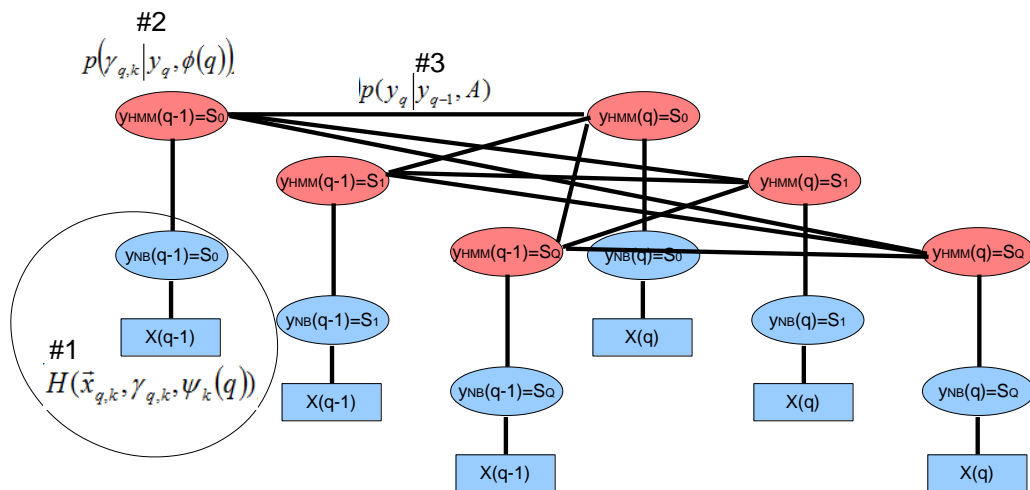


Figure 8.7. Implementation model for the derived smart home equation (8.8).

A model that shows how equation (8.8) has been implemented in the HL-SHS simulation model is given in Figure 8.8 (Lynggaard, 2012b). Leftmost is the action buffer that receives the actions arriving from the S.O.s. These are placed in the action buffer in the time order of arrival. Older actions in the action buffer that are beyond the predefined Look-Back Time (LBT) limit are simply discarded in a cyclical manner, such that the newest actions are always placed first in the buffer.

These actions are processed as follows. Assuming M actions are available in the action buffer and that they can be presented as an action vector A_t containing all actions that arrive in the LBT time window (Subsection 5.5.1) yields (8.9):

$$\vec{A}_t = \{a_t^1, a_t^2, \dots, a_t^M\} \quad (8.9)$$

where a_t^m is the action m arriving at time instance t and $m \in \{1 \dots M\}$.

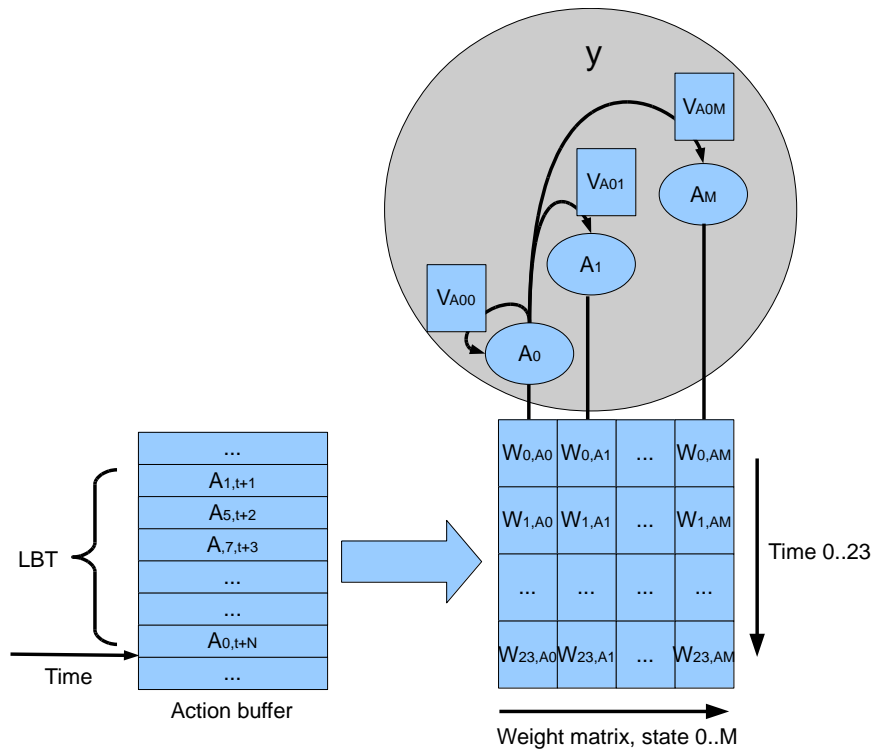


Figure 8.8. Implementation of artificial intelligence in HL-SHS.

Because a particular action can be present more than once time in the action buffer, the following matrix can be defined:

$$\widehat{B} = \begin{Bmatrix} a_{t_0}^1, a_{t_0}^2, \dots, a_{t_0}^M \\ a_{t_1}^1, a_{t_1}^2, \dots, a_{t_1}^M \\ \vdots \\ a_{t_N}^1, a_{t_N}^2, \dots, a_{t_N}^M \end{Bmatrix} \quad (8.10)$$

where the time buffer length (it's LBT) is defined as N and an action at time instance n as t_n , given $n \in \{0 \dots N\}$. At this state, a filter that combines repeating actions in the time buffer can be deployed. Its purpose is to eliminate multiple actions of the same type so they do not exhaust the time buffer. A similar approach has been used by other researchers (Kasteren et al., 2008a), (Fang et al., 2012).

The next step in the processing is traversing the matrix \widehat{B} in a column row order. First, the action name is used as a key to select the respective action column; for example, $a_{t_x}^m$ selects action column m in the matrix, and so forth. When the action column is selected, the action arrival time t_x is used as an index that identifies the action row. By using this action column/row pair, one specific weight in the weight matrix is identified, see Figure 8.8. Thus, each action is related to a weight that expresses its un-scaled probability (the term “un-scaled probability” means that it does not sum to one). In this work the time quantization has been chosen with a granularity of one hour in the weight matrix. This choice is a good compromise between activity resolution and memory/processing usage (Fang et al., 2012). Because these weights quantize time into intervals of one hour it means that there are 24 of them, i.e., no date or year information is provided.

The weight matrix (Figure 8.8) is connected to the states (circles) contained in the (hidden) latent variables (gray circle), where each state symbolizes one of the received actions. Thus, as discussed earlier, this work uses one state per action. These states are connected by transition weights named V_{Aij} , where footnote i and j are the originating state and the arriving state indexes, respectively (Figure 8.8). Therefore, this system offers the preservation of the action arrival sequence and its arrival time.

Looking at Figure 8.8, its relationship with the HMM is noticeable, especially because the action buffer is modeled as the observable variable and the states as a hidden variable. The fact that the HMM offers a relaxation of the i.i.d. assumption often used to simplify classifiers, means that cross correlation between the actions can be handled. Often, a huge matrix is required to handle this, but by using the Markov assumption, it can be assumed that future predictions are independent of all but the most recent observations.

8.3.3 HL-SHS LEARNING

The target in the following subsection is to explore the learning and prediction principles for the HL-SHS AI system that is illustrated in Figure 8.8. Based on the model relationship and the similarity with the HMM model, its theoretical framework together with the well-established Viterbi theory, is used (Bishop, 2006).

The HL-SHS activity learning process is based on a predefined activity to be learned, such as *set home into sleep mode*. This activity and its performed action (e.g., setting the home in sleep mode) are assigned to one of the agents contained in the HL-SHS. Thus, each time the user triggers this predefined action, the HL-SHS runs its learning process and thereby, gradually adapts the action sequence enclosed in the activity. Thus, the HMM learning process used in this work is iteratively based, i.e., the model learns in the form of real-time learning. This is achieved by updating the estimated joint probability $p(\bar{y}, \bar{x})$ each time the user supplies the predefined action. It is assumed that the y -vector contains the predicted action and that the x -vector contains the input from the actions in the action buffer (Subsection 5.5.1.) Rewriting equation (8.8) into an iterative form using the discrete time index q equation (8.11) arrives at:

$$p(\bar{y}_q, \bar{x}_q, \theta(q))_{HMM} = \prod_{k=1}^K G(q, k, \theta(q))$$

Where

$$G(q, k, \theta(q)) = \begin{cases} H(\bar{x}_{q,k}, \gamma_{q,k}, \psi_k(q)) p(\gamma_{q,k} | y_q, \phi(q)) p(y_q | y_{q-1}, A), & \text{if } H(\bar{x}_{q,k}, \gamma_{q,k}, \psi_k(q)) > 0 \\ 1 & \text{else} \end{cases} \quad (8.11)$$

As discussed, the H function in equation (8.11) can be seen as a distant source that emits actions. Using this view, the G function reduces to the two last terms in (8.11). The middle term $p(\gamma_{q,k} = i | y_q, \phi(q))$ expresses the observation probability that observes the value of the emission variable, i.e., the incoming actions. As noted, it is a conditional probability, which depends on the current state in the latent variable. However, as discussed earlier, one action is connected to one state only; therefore the latent variable y_q needs only to be trained in that particular state. The learning parameter for this middle term is contained in the $\phi(q)$ parameter. The final term expresses the transition probability between two states in the latent

variable y at time instance q . It uses an A parameter that adapts the learning parameters. In Figure 8.8, the A parameter is implemented in the form of the V_{Aij} weights. Therefore, from a training point of view, these parameters need to be updated in an adaptive way when the dedicated action arrives.

Regarding the observation probability the parameter vector $\phi(q)$ can be described by using a Bernoulli distribution. This distribution is suitable because the observation probability receives binary values from the NB's H functions. Therefore, finding the optimal parameter set simplifies to optimizing the Bernoulli distribution probability by using the Maximum Likelihood parameter Estimation method. For this distribution, the parameters can be solved analytically by using the Dirac delta function (Bishop, 2006); a discussion about this distribution can be found in subsection 5.5.2. The maximum likelihood solution expresses that the optimum estimated parameter set is found by simply using the parameter frequency as weights (Rish, 2001).

Viewing the learning process from an implementation point of view (Figure 8.8), training is undertaken by summing the same actions in the appropriate LBT time window and adding these results to the respective weights. Actions that occur multiple times in any particular time window are counted as separate actions.

Regarding the right-hand side second factor of (8.11), it expresses the transition probability distributions $p(y_n | y_{n-1})$, which represent the probability of going from one state to the next state. These probabilities can be found by following the same derivation steps as outlined for the observable probabilities. Therefore, training the weights is undertaken simply by summing the number of transitions between the state pairs over all the states, i.e., their frequencies. In general, these procedures risk overflowing the contained weights, but this can be handled by using either relative scaling or a variable saturation scheme.

8.3.4 HL-SHS PREDICTION

Regarding the activity prediction part, it is an inference problem where the goal is to find the state sequence that maximizes the joint probability $p(\hat{y}_q, \hat{x}_q, \theta(q))_{HMM}$. One efficient strategy that is often used is the Viterbi algorithm (Bishop, 2006), (Fang et al., 2012), (Kasteren et al.,

2008a) because it reduces the calculation's complexity. Combining this with the first order Markov assumption and with the knowledge of ending in the dedicated state reduces the buffer look-back to one step only. Based on these reductions, the simplified Viterbi optimization process can be expressed as:

$$\lambda(i) = \max_{y_{n-1}} P(y_{n-1}, y_n = i, \bar{x}_{n-1}, \bar{x}_n) \quad (8.12)$$

which states that given y_n is the specific predefined action (i.e., the final state $\lambda(i)$) for the particular agent and that the vectors x_{n-1} and x_n are the given observation, the highest probability must be searched by varying the choice of the previous state and its transition probability to final state i . By using the transition weights defined in Figure 8.8 and the HMM mathematical description, this maximization process can be rewritten as:

$$\lambda(j) = [\max_{1 \leq i \leq M} p(\bar{x}_{n-1} | y_{n-1} = i) V_{Aij}] p(\bar{x}_n | y_n = j) \quad (8.13)$$

where it is assumed that the initial probabilities are all equal, i.e., they are left out of the equation. This expression can be simplified by assuming that the final state (S_F) is the predefined agent state. Using these assumptions, equation (8.12) reduces to:

$$\lambda_F = [\max_{1 \leq i \leq M} p(\bar{x}_{n-1} | y_{n-1} = i) V_{Ai, S_F}] p(\bar{x}_n | y_n = S_F) \quad (8.14)$$

From this expression, the most likely state pairs consisting of the present final state (S_F) and the previous state, including its transition probabilities, can be found. The complexity of the Viterbi algorithm is in general $O(T \cdot Q^2)$ (Bishop, 2006); however, this simplified algorithm has a complexity of $O(Q)$. This simplification means that the HL-SHS AI system has a reduced load on its implementation device.

The general HMM theory states that a state sequence is the best classification candidate (Bishop, 2006). However, in the suggested HL-SHS, a state sequence and its related probabilities do not provide the full information. Thus, in this system, a binary decision is needed to state whether the S_F action is taking place or not. To make this decision, a quantization of the probability and a comparative threshold are needed. Such a threshold must be a function of the weight matrix in Figure 8.8 because it changes its values each time it is

trained, i.e., each time the user accepts the predefined agent state S_F . Therefore, the threshold must be updated each time the weights are updated. This requires that the calculation of the threshold algorithm be simple, i.e., the amount of power needed by an embedded implementation is limited.

A useful concept is to consider each action (i) as a random variable $RV(i, j)$ that is distributed over a time (j) window from 0 to J where $j \in \{0..J\}$. The problem is that this distribution is unknown and it changes slowly as a function of time; therefore, using the finite limit theorem to assume that the distribution is normal is not infallible. Thus, finding the mean of this distribution by using an estimator based on the expectation theorem is meaningless. Given these unknown conditions, the best choice is to assume that the probability is the same for all time instances. This means that the time can be marginalized out, or more simply put; the mean estimation process degrades to calculate the simple average of the values (8.15).

$$E[RV_A(i)]|_{p=constant} = \frac{1}{J} \sum_{j=1}^J RV_A(i, j) \quad (8.15)$$

Using this estimator to find the two largest mean values in the collection of learned action probability distributions ensures that the predefined agent state and its most related action are found. This principle is in good agreement with the principles in the Viterbi algorithm. Finally, combining these mean values with their linking transition probability provides the scalable threshold used in this work.

A summary of the prediction process is as follows. When a new action arrives in the action buffer, it is compared to dedicated actions S_F for a match. If a match is found, the buffer is traversed by processing the actions one at a time. Therefore, based on each action (i) its time-quantized weights are located, i.e., the un-scaled probability $p(\bar{x}_{n-1}|y_{n-1} = i)$. This weight is then multiplied with the transition weight connecting that state to the specific predefined action S_F , i.e., the un-scaled probability weight V_{Aij} . At the end of this process, the highest product value is found and multiplied by the weight of the un-scaled probability $p(\bar{x}_n|y_n = S_F)$. After using the previously discussed threshold limit, action values that exceed this limit are considered likely candidates, i.e., they are emitted. Note that the threshold process controls the

importance of the selected activity and thereby, whether it is presented for the user. The output of all the agents is predicted using these principles.

8.4 SIMULATION AND PERFORMANCE OF THE HL-SHS

This section covers the HL-SHS simulation models, which are part of the larger model developed and described in Lynggaard (2011). They have been used for simulating, describing and discussing two smart home datasets. Finally, the results of the simulations are evaluated and discussed. The content in this section has been further elaborated in Lynggaard (2012b).

8.4.1 SIMULATION MODEL

The described SHS is modeled using a model developed in Java that runs on a personal computer. All the essential algorithms discussed earlier are implemented in this model. Similarly, the model is part of a more complex model described in Lynggaard (2011). Datasets to feed this model are derived from a real-world setting. This is important for reflecting the real-world uncertainties and sensor noise in the datasets. These elements are vital in testing algorithms to ensure that they are able to filter out the correlated information and are able to converge in the learning process.

A Unified Modeling Language (UML) based class diagram for the HL-SHS simulator model is provided in Figure 8.9.

The principles of the Object Orientated Method (OMT) are used in the design of this simulator. This enables the possibility to encapsulate common functionality, which reduces coupling between objects and uses OMT principles, such as generalization and polymorphism. The central attributes are allocated in the model class, i.e., the “HL-SHS model parameters” class. This class inherits the datasets from the “Hndl. Dataset” class. Using this principle ensures that the model context is decoupled from the model itself, but that the dataset is still available for the model classes. The model classes are “PredictModel” and “TrainModel”. These classes are tightly coupled to the model parameters by the used aggregation. This tight coupling is necessary because it combines the data and the method

parts of the HL-SHS model into one unit. Finally, the “Hndl. score” and “UI” objects are coupled to the model, such that it is possible to interact with it and receive results.

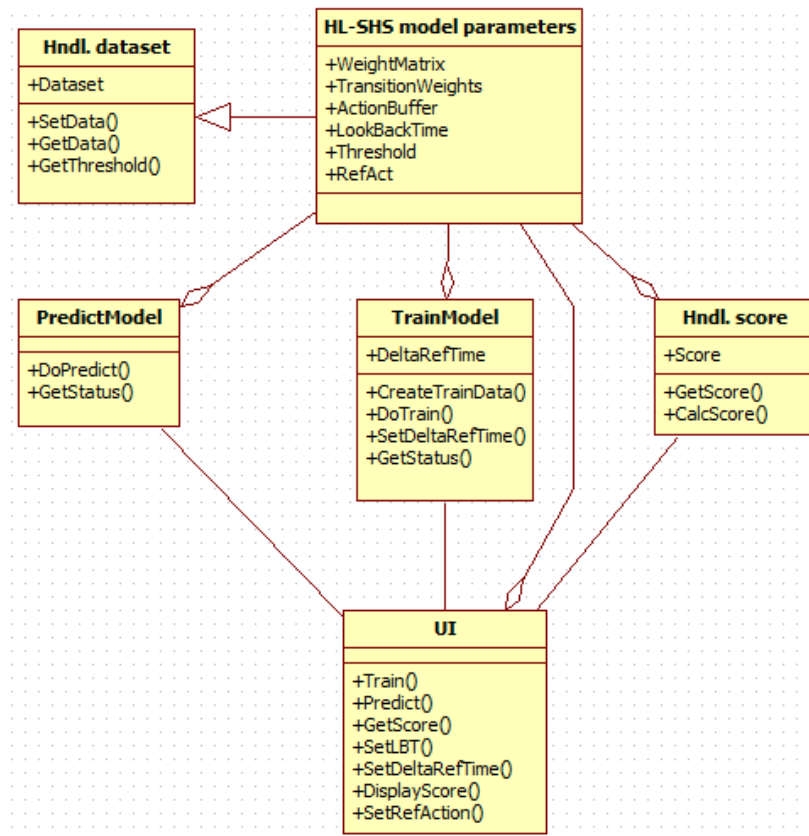


Figure 8.9. HL-SHS implemented model - class diagram.

Many methods are available in the HL-SHS model. Basically, these methods follow the OMT strategy of using getters and setters. Thus, methods are available in each class for getting access to the attributes and changing their values in a safe way. Furthermore, the “UI” class provides a collection of operations, such as “Train”, “Predict”, and “DisplayScore” that allows the user to interact with the model in a structured way.

An example of a user-triggered event flow in the simulator is provided in Figure 8.10. Flow 1 illustrates the setup of the simulation-relevant attributes such as the predefined activity “RefAct” action, etc. Flows 2,3,4,5 and 6 cover the model training where the aggregated data

are updated, i.e., the weight matrix, transition weights etc. Prediction takes place in flows 7, 8, and 9. Finally, the result is provided to the user in flows 10 and 11.

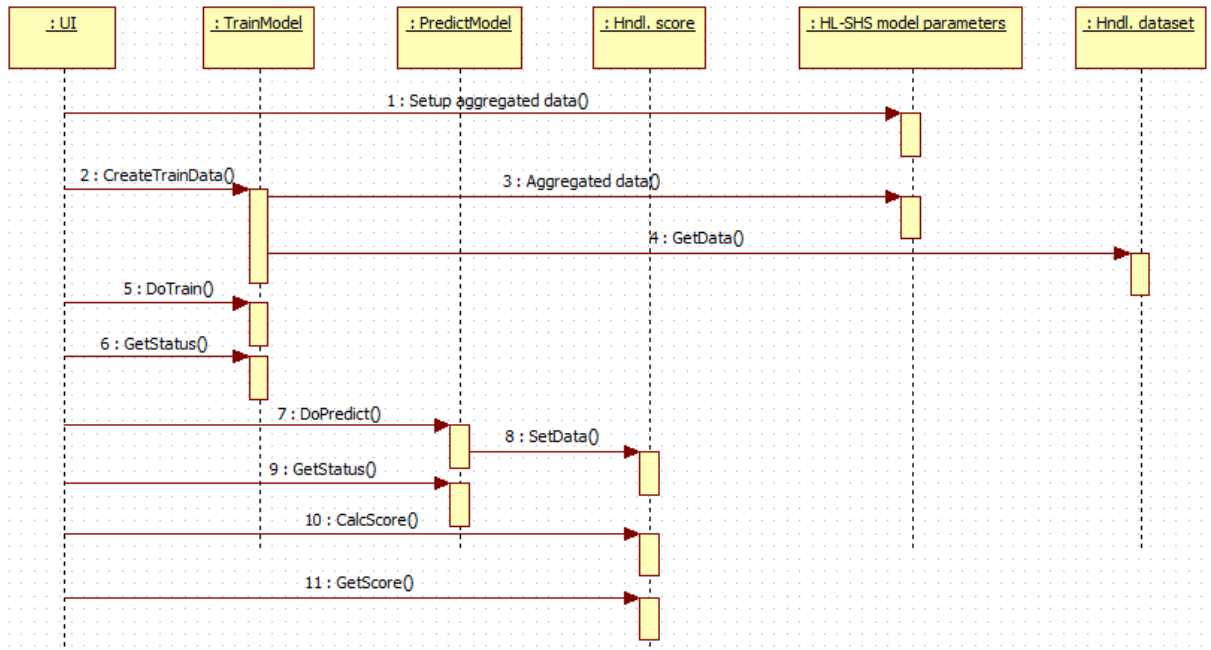


Figure 8.10. HL-SHS implemented simulation model - sequence diagram.

8.4.2 SMART HOME SIMULATION EXAMPLE I

To validate the HL-SHS performance, the Aruba 2010–2011 dataset from the WSU CASAS project has been used (Cook, 2012), (Section 4.3). This was recorded in a house with 26 sensors in which a woman lived for approximately 7 months (Figure 8.11). The woman’s children and grandchildren visited on a regular basis. This resulted in 6468 user annotations and in 1719558 sensor events. The reason for choosing this dataset is that it reflects a general real-life usage situation, i.e., it provides a long time general dataset from universal sensors placed at common logical points.

The previously presented simulator has been used to simulate the HL-SHS with an LBT set at 120 minutes and where the buffer length was restricted to 100 actions. These choices are in good agreement with the temporal resolution used in the HL-SHS model of one hour and they

are supported by the work of Fang et al. (2012) which found this temporal resolution size advantageous.

The common way this dataset has been used in many papers is by training a classifier with most of the data and omitting a small part of the dataset to be used for testing. This approach is not feasible in this work because learning must happen “on the fly”, i.e., if the user performs a predefined activity, it is used immediately in the probability estimation process to update the weights. Therefore, in this work a predefined activity has been assigned as a trigger for updating the weights, i.e., activity learning. To be more specific a leaving home scenario is presented:

Leaving home scenario: When the user leaves her home she often sets the home in power sleep mode by using her mobile phone. The term power sleep in this context means that the heating system setting is set to a predefined minimum and that all unnecessary electric installations are turned off, and so forth. A criterion for setting the home to power sleep could be that the user does not expect to return within one hour. This leaves some reaction space for the inertia in the heating system.

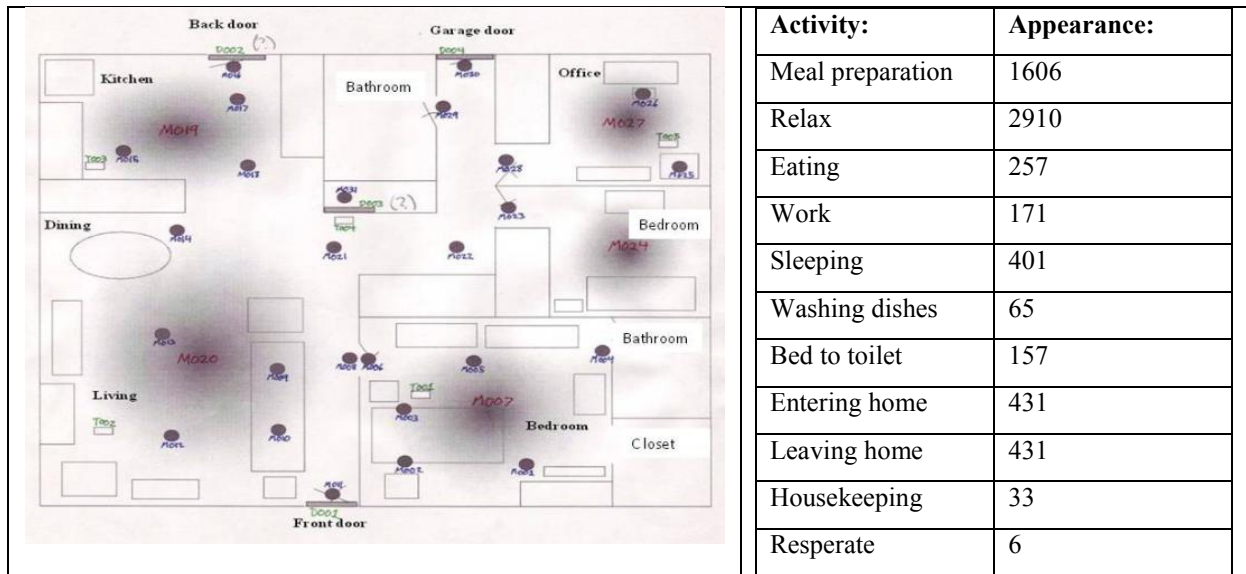


Figure 8.11. (Left) a CASAS experimental house (Aruba) equipped with 26 sensors. (Right) annotated activities used and their number of appearances in the dataset (Cook, 2012).

To be able to detect this specific scenario the “power sleep” command must be related to a dedicated action type that can be used as a trigger for the SHS learning. Because the power sleep command does not exist in the CASAS dataset, it has been associated with a “leave home” and “enter home” difference of more than one hour, where the “leave home” is the dedicated action (i.e., the triggering action).

By running the HL-SHS simulation model with the CASAS dataset and by using the discussed assumptions, it is possible to train the weight matrix Figure 8.12, which represents the actions given in Figure 8.11. Exploring the results of the trained weight matrix provides much information about data correlation, i.e., how the data are related. The result is presented in Figure 8.12, about which it should be noted that the weights are normalized.

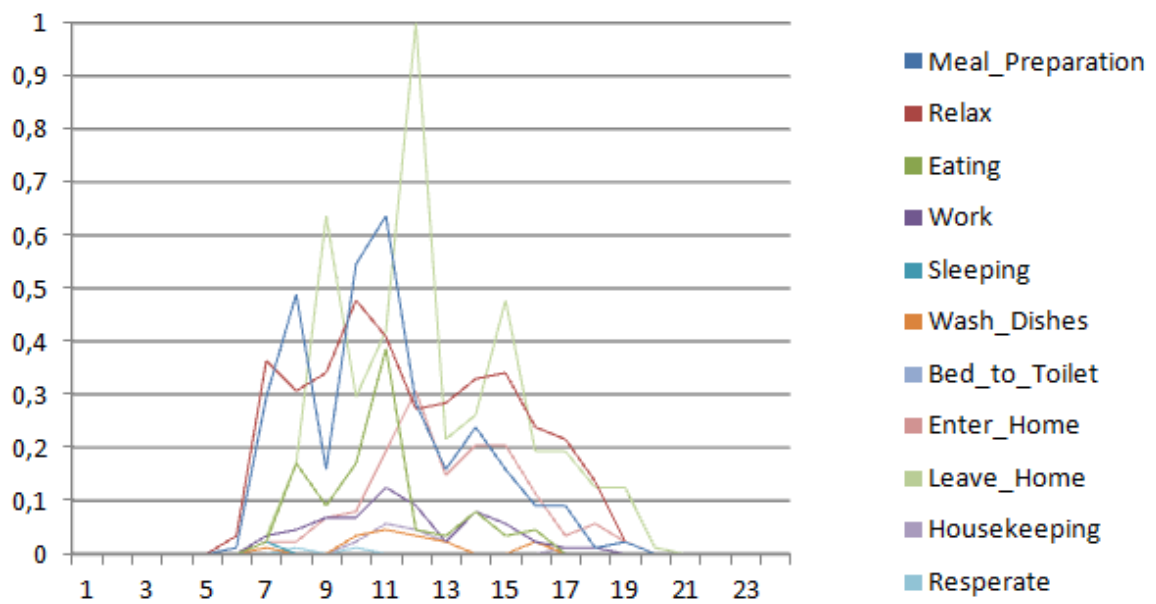


Figure 8.12. Weights for the trained HL-SHS model, i.e., when the user leaves form more than one hour. The normalized weight values (y-axis) as a function of time quantized into chunks of one hour (x-axis).

By looking at the “leave home” curve, it can be seen that it has peaks at 8, 11, and 15 hours, meaning that the user is most likely to leave home at these times. By focusing on “meal preparation”, it can be seen that this peaks just before the “leave home” peaks. This suggests that a correlation exists between preparing meals and subsequently leaving home. Actually, this is what most people would be expected to do, i.e., eat breakfast and then leave home for work. The action “relax” also seems to have some correlation with other actions, because it has similar peaks. The question is then whether those cases where the user leaves for more than one hour can be differentiated from the cases where the user will be back within one hour, based only on the weight matrix. To explore this, the three actions “meal preparation”, “relax”, and “leave home” are redrawn (Figure 8.13) for those cases when the user does not leave home for more than one hour.

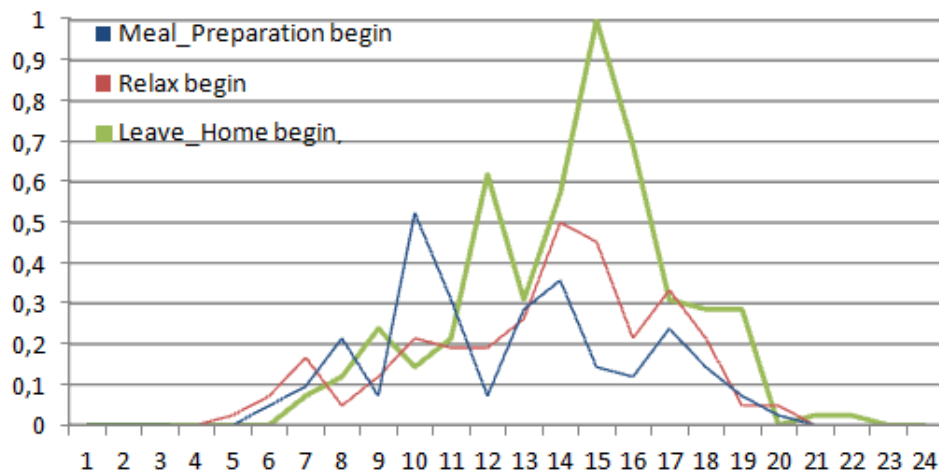


Figure 8.13. The activities "meal preparation", "relax", and "leave home" when the user does not leave the home for more than 1 hour. The y-axis is normalized weights and the x-axis is quantized time in one hour chunks.

By comparing the results when the user leaves for “more than one hour” Figure 8.12 and for “less than one hour” Figure 8.13, it can be observed that the user does not leave home very often at eight o’clock and stay away for more than one hour. Both diagrams have a peak at 12 o’clock for the “leave home” action, i.e., this information is ambiguous. To use the principles from HMM theory to increase the detection probability means involving previous actions. By comparing the “meal preparation” activity in both cases, it can be noted that this action is close in time to the “leave home“ action in the “more than one hour” case. This means that the “leave home” for “more than one hour” action has detectable features that can be used by a trained HMM model to increase the prediction accuracy.

As discussed, the presented HL-SHS algorithm (Section 8.3) has been evaluated by using the developed java based simulation model (Section 8.4) and the leaving home scenario. Scores for this evaluation are presented in Table 8.1.

Rating	Score (percent)
True Positive (TP) in percent	75
False Negative (FN) in percent	25
False Positive (FP) in percent	36

Table 8.1. Score for the fully trained classifier for the leaving home scenario (user leaves home for more than one hour).

The rating method used in Table 8.1 is explained in the following. The term TP means true positive, i.e., the annotation is positive and the tested algorithm provides the same result. FN is false negative, meaning the annotated action is present, but that the algorithms did not detect it. Therefore, the score of TP + FN gives 100%. Finally, FP (false positive) means that the annotation provides no action, but the tested algorithm estimates one. The scores percentage is found by dividing the counted scores in each category and then dividing them with the ground truth user-annotated action event number. The TN does not provide any information because it indicates the situation where nothing happens, even though the user triggers different sensors and the tested algorithm predicts same the result.

From the numbers in Table 8.1 it can be seen that the HL-SHS algorithm estimates the leaving home activity correctly (TP) approximately 75% of the time after learning from 295 user-annotated activity events. It is also found that it fails to estimate a user-annotated activity (FN) 25% of the time. This is not a serious problem because from the user’s point of view it means that an activity is not suggested, to which the user response will probably be that the user performs this activity manually. However, because the user performs this action manually it produces a dedicated action from which the system learns and thereby improves, i.e., it “bootstraps”. More serious is the false positive (FP) outcome, because this means that the algorithm suggests an action that is not requested by the user’s behavior, which would probably be annoying for the user over time. This FP rate can be reduced by either adjusting the classifier parameters or by using additional sensors to provide finer granularity of the information produced by the user’s actions (Fang et al., 2012).

A similar simulation (Table 8.2) that is also based on the “leave home” scenario has been performed. It uses a changed condition for the training set, i.e., it requires that the user must

leave the home for more than a half hour, otherwise the parameters are the same as in the previous simulation run.

Rating	Score (percent)
True Positive (TP) in percent	86
False Negative (FN) in percent	14
False Positive (FP) in percent	33

Table 8.2. Score for the fully trained classifier for the leaving home scenario (user leaves home for more than half an hour).

It has been found that in this case the algorithm estimates the user behavior correctly approximately 86 % of the time. Also notable is that the FP rate stays at almost the same level as in Table 8.1. The reason for an increase in the TP rate is that more training sets become available because this training set includes values in the window of a one-half to one hour, plus the values from the training set used in Table 8.1 (one hour and above). The reason that the FP rate is almost the same is because the number of actions that are able to overcome the threshold limit is almost constant and because the half-hour scenario includes the one-hour scenario.

A direct comparison with others' work is not straightforward because, to the author's knowledge, similar distributed SHSs have not been published before. However, it seems reasonable to compare the HL-SHS with other systems that predict user activity by using a sequence of events. One example is given in the work of Cheng et al. (2009). They designed a scenario based on a user-activity prediction system named ASBR, (Section 5.2), (Subsection 6.1.1). They also compared the performance of their system with a Case-Based Reasoning (CBR) approach.

The results of the comparison between the presented system and the ASBR and CBR systems from Cheng et al. (2009) can be found in Table 8.3. It is noteworthy that they all perform almost equally well. However, it should be noted that these systems are different both in their approach and their design (Section 5.2), (Subsection 6.1.1), even though the results are comparable.

System	Score (percent)
This work	86
ASBR (Cheng et al.)	80
CBR (Cheng et al.)	75

Table 8.3. The presented work compared with ASBR and CBR from Cheng et al (2009).

8.4.3 SMART HOME SIMULATION EXAMPLE II

A second simulation has been performed to explore the HL-SHS performance in another context, which is illustrated in Figure 8.14. This uses the Cairo 2009 dataset from the WSU CASAS project (Cook, 2012). This was recorded in a house with 31 sensors, in which a volunteer adult couple lived for approximately three months. The residents in the home were a man (R1), a woman (R2), and a dog. The couple’s children also visited the home on at least one occasion. This resulted in 726534 sensor events and 600 annotations by the users. Thus, these data are used for the “go to sleep” simulation presented in the following. The reason for choosing this dataset is that it supplements the Auruba dataset used in example one, and because it is generalizable for many similar settings. Thus, this dataset provides a more challenging environment in which two people activate the sensors. All sensors are placed at logical positions and they provide generalizable data.

The simulations performed in this experiment are similar to those used in subsection 8.4.2. Hence, the same HL-SHS simulator model has been used with the same settings. The only exception was that the dedicated action was set to detecting when the man (R1) goes to sleep and sleeps for more than three hours. Such a predefined action is usable for setting the home in a “night mode”, i.e., the room temperature is lowered, all unnecessary electrical equipment is turn off, and the lighting is set to sleep mode. By using these conditions and settings in the simulation model produces the results presented in Table 8.4.

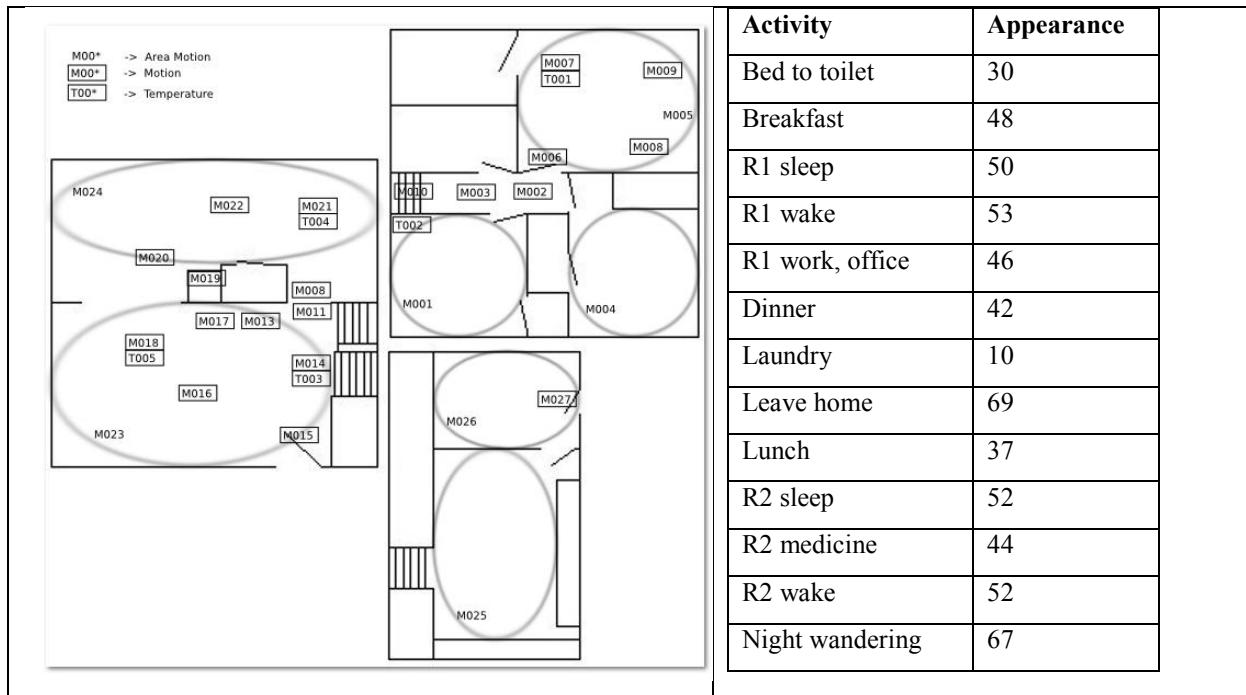


Figure 8.14. (Left) a CASAS experimental house (Cairo) equipped with 31 sensors. (Right) annotated activities used and their number of appearances in the dataset. This house is used in the “go to sleep” simulation (Cook, 2012).

As illustrated in Table 8.4, the score for the “go to sleep” activity is 94 % after learning from 50 user-annotated actions (Figure 8.14). In addition, it can be noted that the FN rate is 6 %, i.e., 6 % of the actions are not detected. As discussed earlier, this is not a problem for the user; however, the FP rate is generally an annoyance for the user, but in this case it is only 4 %.

Rating	Score (percent)
True Positive (TP) in percent	94
False Negative (FN) in percent	6
False Positive (FP) in percent	4

Table 8.4. Score for the fully trained classifier for the “go to sleep” simulation (user R1 sleeps for more than three hours).

Another interesting measure is how fast the learning occurs in this HL-SHS model, i.e., the score as a function of training times. This is relevant because the HL-SHS model uses real-time learning. These results are presented in Table 8.5.

Rating	Score after training round (percent)				
	1	2	10	20	25
TP	14	86	86	92	94
FN	86	14	14	8	6
FP	2	4	4	4	4

Table 8.5. Score for partly trained classifier for the "go to sleep" simulation (user R1 sleeps for more than three hours). Numbers in x-direction are score after trained the given number of times.

The rating as a function of training times can be found in Table 8.5. It can be noted that the HL-SHS model converge quickly in the beginning, such that the TP rate increases from 14 % to 86 % after the second training round, i.e., when the predefined action is seen a second time. In addition, it can be noted that after the second round the ratings only increase slowly and asymptotically. This is the case because the actions are correlated to a very high extent, such that only a limited amount of new information is provided following each training time. Furthermore, the noise level is more significant after additional training times, i.e., a greater number of sensor permutations are seen over a longer time. Some of these increase weights that are uncorrelated to the predefined action and thereby, they increase the risks for making false decisions. Actually, this behavior is normal for most classifiers used in smart homes (Kasteren et al., 2008a).

8.4.4 HL-SHS SUMMARY

A high-level distributed SHS has been presented. It offers a concept that combines a simple low-level activity classifier named LL-SHS with a high-level one named HL-SHS, which has been explored in section 8.4. Research and simulations have revealed that the system is able to learn and predict user activities based on action time and a sequence of user actions provided from a low-level activity classifier. The research started by exploring whether user activity information is available as a sequence of user actions within a given time window. It was found that this is the case, but that the overlying noise sets the detection lower limit. Next, the publically available CASAS dataset was used to evaluate the system algorithms and provided good results. Thus, a TP rate of 75% and 86% was found in the "leave home" simulations and 94% in the "go to sleep" simulation.

Comparing the achieved results with comparable systems shows that their performances are similar. Thus, a TP rate of 86% in the described half-hour scenario is at the upper range of that normally achieved in comparable systems.

The future perspective of this work is to investigate the possibility of implementing HL-SHS on different hardware platforms. Furthermore, an investigation of the look-back depth in the HL-SHS action buffer also needs further investigation.

8.5 PROCESS MODEL OF LOW-LEVEL SMART HOME SYSTEM

As discussed earlier, the smart home AI concept can be viewed as a combination of an HL-SHS and an LL-SHS, i.e., an HMM model and an NB model as illustrated in Figure 8.4 and Figure 8.5. It is noted that the LL-SHS actually consists of a collection of S.O.s that in turn contain a collection of NB-based agents. Therefore, in this section, the NB model for the LL-SHS is presented including its use of S.O.s. The LL-SHS including its S.O.s are discussed in Lynggaard (2013c) and Lynggaard (2013b).

8.5.1 OBJECT PROCESS MODEL

An object process methodology model (Dori, 2002) for the LL-SHS is presented (Figure 8.15) and discussed in the following.

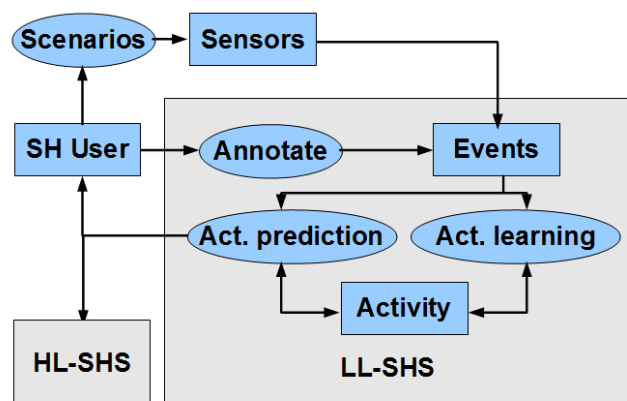


Figure 8.15. An object process model for the smart home system part one (LL-SHS).

As seen in Figure 8.15, the user (SH-user object) carries out scenarios (scenario process) in the form of a normal living pattern within a smart home. When the user interacts with the

smart home through these scenarios, its sensors are activated, i.e., triggered (Sensor object). Each of the triggered sensors emits events that are received by the LL-SHS. When the events arrive they are annotated with a time stamp (Annotate process). At this point, the annotated events are divided into two streams: one that is fed to the action-learning process (Act. learning), and the other that is fed to the action-prediction process (Act. prediction). The action-learning process uses the received events to train the AI model (Activity object). Based on this trained model, the action-prediction process assesses the annotated events and predicts an upcoming user action. If an acceptable action posterior probability level is found, the action-prediction process schedules the action to the user (SH-user object) and thereby, offers the predicted service to the user. Furthermore, the action is offered to the HL-SHS for high-level processing.

8.5.2 LL-SHS DETAILED OVERVIEW

A simplified overview that covers the agents including their NB classifiers used in the LL-SHS concept is provided in Figure 8.16 (Lynggaard, 2013b).

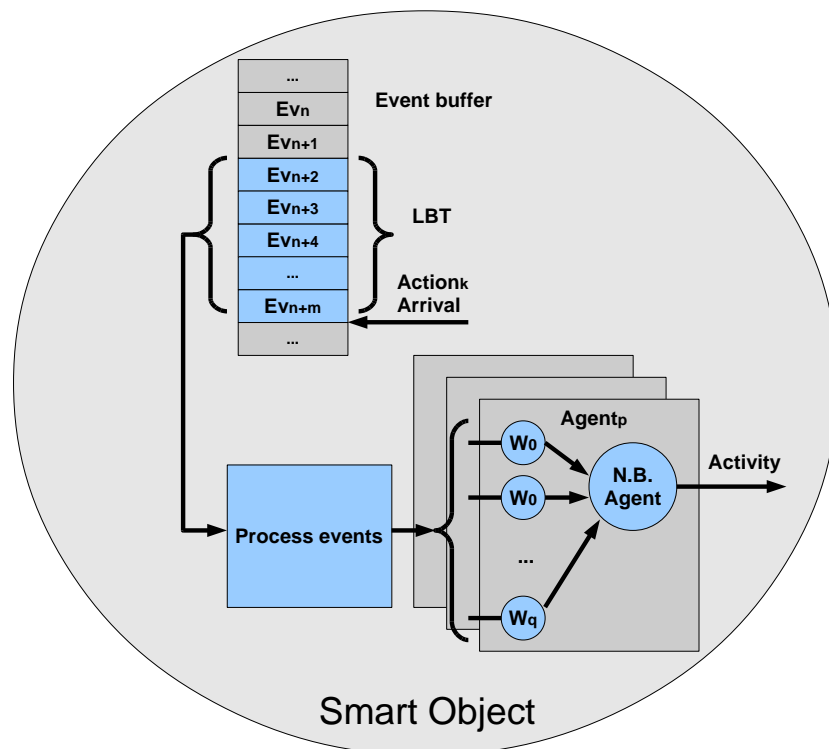


Figure 8.16. The naïve Bayes agent concept that is used in the smart objects. These smart objects constitute the foundation of the agents in the LL-SHS.

The LL-SHS (Figure 8.16) runs in a forever manner, monitoring its input for available events. From an implementation point of view, this means that it powers down and waits for wakeup interrupts emitted when sensor data arrives. The prediction process is performed every time an event arrives. It puts the events into the event buffer ordered by their time of arrival. From this buffer, a group of events that is limited by the length of the event buffer time window is fed to a process event transformer. This combines the input events and runs a balance system that prevents the agent weights from overflowing (i.e., scaling). Output from the transformer is multiplied by the weights w_0 and its approximated logarithmic values are summed into one value. This value is compared with a dynamic threshold limit and a binary decision is made as to whether or not to emit the detected action.

Similar to the prediction process, the process of activity learning occurs every time a predefined action arrives. This action is marked in the event buffer and all events present in the event buffer time window are processed in the process event transformer. Output from this transformer is used to update the weights. Therefore, this means that the system is adaptive and that it offers real-time learning without any user intervention as long as the user supplies ground truth actions by performing activities.

The following section looks into the details of the presented LL-SHS.

8.5.3 SENSOR OBJECTS FOR LL-SHS

A smart home is assumed to contain many complexes S.O.s. These can be simple S.O.s, such as a collection of binary sensors connected to an S.O., or more advanced embedded device objects such as a television. A factor common to them all is their ability to emit an action based on sensor information that is processed by a trained AI system. However, as discussed previously (Section 4.3), S.O. simplification is important in this work; therefore, the general multi-valued sensor concept is transformed into a simpler one that uses binary values (Section 4.4).

It is assumed that these sensors are placed in the smart home at optimal positions and that they are able to transmit simple events with only a small delay (less than a fraction of a second). As stated (Section 4.4), these simple sensor events contain a limited amount of information (identity) and they are expected to occupy only a few bytes of payload in the transmission

context. It is also assumed that each sensor samples its context every second and that this sampling period is fast enough to avoid aliasing (i.e., it obeys the Nyquist-Shannon sampling theorem). Therefore, from a transmission point of view, every second, only a limited amount of information is sent from each sensor when it is activated; for the wireless sensors, this keeps the network load low. Such a simplification comes at a price because a real-time continuous event is quantized into an impulse event, which is annotated with a unique time stamp. This approach causes losses in the form of time precision.

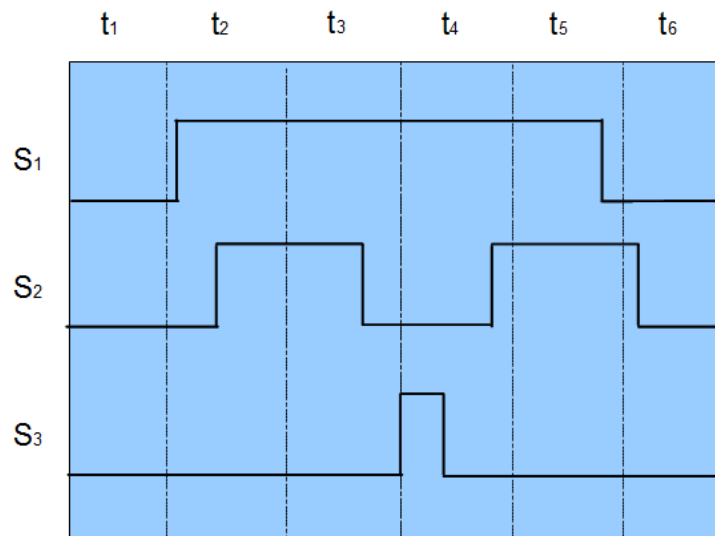


Figure 8.17. Example of three sensor outputs S_1 - S_3 . X-axis is divided into sampling intervals named t_1 - t_6 .

To explore these effects in detail they are illustrated in Figure 8.17. As shown, three sensors $S_1 \dots S_3$ are changing between on and off as a function of sample interval time $t_1 \dots t_6$. The rising edge for sensor S_1 inside sampling interval t_2 is detected at the beginning of sampling interval t_3 where the sampling takes place (vertical dotted lines in Figure 8.17). Therefore, it is quantized into a Dirac impulse transmitted at the beginning of time t_3 . Only this event is transmitted to the SHS, i.e., the off event is simply ignored. From this, it can be noticed that the on value for S_1 is delayed by approximately one sampling time step (or 1 second). It can also be noticed that the short on-off pulse from sensor S_3 will probably be lost by aliasing because its duration is too short for the sampling period. To avoid this loss, the local sensor circuit could either sample more frequently and downsample afterwards, or alternatively use some peak hold feature. However, this requires additional sensor resources.

Thus, it is expected that the worst-case time offset between the real-time and the stamped sensor events would be approximately one second (transmission delay is small, so it is ignored). As discussed, the off time (falling edge) information is ignored to save sensor resources. This choice comes at a price because the event duration information is also lost. However, Kasteren et al. (2008a) have shown that such a concept does not degrade the recognizability significantly.

8.5.4 EVENT BUFFER FOR LL-SHS

When events arrive they are stored in the event buffer. Often it is necessary to sort events according to their arrival time and to mark multiple events from the same source in the event buffer, as discussed earlier (Subsection 8.5.2). However, in the LL-SHS, the arrival time information is not used because the AI part is based on the non-temporal naïve Bayes classifier. Hence, this classifier assumes that all data points are i.i.d. and it does not take into account any temporal relations between the events. It is noticeable that this simplification does not lower the recognition probability considerably compared with predictors that are more advanced (Subsection 5.6.2). The multiple event information is also dropped for the same reasoning.

Because the arrival time is not needed, it is marginalized out by combining all rows in the event buffer. This combining operation is performed by using a logic “or” process and only allowing each event to be present once. This operation gives the sample buffer equation (8.16):

$$\vec{S} = [e^1, e^2, \dots, e^N] \quad (8.16)$$

where $e^n \in \{1,0\}$ for all n . Hence, a simple example of the use of the sample buffer (8.16) could be:

$$\vec{S}_{bin} = [1,1,0, \dots, 1] \quad (8.17)$$

where a one in the first vector position of the binary sample buffer (8.17) indicates that sensor one has been on within the time window T ; a one in vector position N indicates the same for sensor N , etc.

8.5.5 LL-SHS ACTIVITY LEARNING

The LL-SHS NB classifier learning process is iteratively based, i.e., the model learns in the form of real-time learning. This is achieved by updating the estimated joint probability $p(y_{1:T}, \hat{x}_{1:T}, \psi)_{NB}$ whenever the user supplies a predefined action as stated in equation (5.5) and (5.6) (Subsection 5.5.2).

It is assumed a y -vector contains the predicted actions and an x -vector contains the inputs from the sensors in a given time instance window T . Thus, when a predefined action $y=i$ has arrived, the conditional probability for x being the vector with N elements that produces the $y=i$ action in the event buffer LBT window, can be expressed as (Subsection 5.5.2):

$$p(\bar{x}_T | \gamma_T = i, \theta) = \prod_n^N p(x_T^n | \gamma_T = i, \theta) \quad (8.18)$$

where θ are the training parameters, i.e., its trained weights. As can be seen, this is a product of the individual sensor conditional probabilities given that y is in evidence (i.e., in state i). Because the sensor information is binary-based, its probability distribution can be described by a Bernoulli distribution as discussed earlier (Subsection 5.5.2). For convenience this is restated in equation (8.19):

$$p(x_T^n | y_T = i) = \mu_{ni}^{x_T^n} (1 - \mu_{ni})^{1-x_T^n} \quad (8.19)$$

where it is assumed that μ_{ni} is the estimated probability for sensor output x^n given y is in evidence, i.e., active state is i . With regard to equation (8.19), three values of μ_{ni} are interesting for this work. First, if μ_{ni} has a high probability, i.e., it is close to one, then sensor x^n must be closely correlated to activity $y = i$. Second, if μ_{ni} is close to one-half, sensor x^n can be considered as not contributing. Finally, if μ_{ni} has a low probability, i.e., it is close to zero, this means that it has a high correlation and x^n must not be present. For this work, it has been chosen that only sensors that are present with a high correlation count. This choice is based on the a priori knowledge that a smart home often implements many sensors, but where the user only activates a few of them at the same time. By using these assumptions, (8.19) can be simplified as:

$$p(x_{\xi}^n | y_{\xi} = i) = \begin{cases} \mu_{ni}, & \text{for } x_{\xi}^n = 1 \\ \text{not used}, & \text{for } x_{\xi}^n = 0 \end{cases} \quad (8.20)$$

where it is noted that probabilities close to one-half are marginalized out. This is achieved by using a plus-minus-one implementation method where the average is zero, as explained in the following section. The probability in (8.20) is used in implementing the H function given in equation (8.5).

As stated in subsection 5.5.2 the maximum likelihood estimate for the μ_{ni} parameter is found by simply adding the x-vector values to the weights when the latent state y is in evidence and then normalizing it. Thus, training is done by adding up the sensor events (represented by plus and minus binary ones) present in a given event buffer LBT. Sensor events that occur multiple times within the event buffer are counted only once. In theory, because the values used are $\{+1, -1\}$, the weights can grow infinitely. To eliminate this behavior maximum and minimum values are defined such that the weights remain inside these by using a simple scaling approach. Additionally, for facilitating real-time calculations, the division operations needed in the probability parameter estimation of μ_{ni} are ignored, i.e., the un-scaled weight values are used. Nonetheless, it is noted that this approach does not reduce the detection likelihood.

The plus-minus-one implementation method also provides the ability to forget learned behavior and to adapt new learning. Thus, if the user changes behavior the new behavior will be learned and the old behavior will be cancelled over time.

8.5.6 LL-SHS ACTIVITY PREDICTION

Each time an event arrives in the event buffer a prediction is performed. Thus, the event buffer content inside the LBT window is fed to the trained NB classifiers, which use these to solve the implicit inference problem of finding the maximum observation probability. If this probability is beyond a certain threshold limit, the binary latent variable y is in the state with value one, which means that the action handled by this particular NB agent is emitted.

Because the discussed NB behavior is implemented in multiple instances in each embedded S.O. and a prediction is performed each time an event arrives, some optimizations are

beneficial for efficient implementation, as long as the prediction performance is similar to, or even better than comparable systems.

From the equation (8.18) it can be observed that the predicted outcome of a given activity y is based on the product of the individual sensor observed and an estimated conditional probability $p(x_t^n | y_T = i)$, multiplied together over N sensors. These multiplications are demanding of resources to implement in an embedded processor environment⁸ (Kevin et al., 2000). Therefore, to optimize the product of the observed and estimated probability $p(x_t^n | y_t = i)$ from each individual sensor, it is approximated by a sum to support a simple calculation strategy and thereby, to enable real-time prediction. This approach can be explored by using the logarithm function on the NB model (8.18) as discussed in the following.

$$L(p(y_t, \bar{x}_t)) = \sum_n^N \ln[p(x_t^n | y_t = i) + \xi] \quad \text{where } \xi \text{ is an inf. small positive number} \quad (8.21)$$

$$L(p(y_t, \bar{x}_t)) \approx \sum_n^N [p(x_t^n | y_t = i) - 1] = \sum_n^N p(x_t^n | y_t = i) - N$$

Equation (8.21) expresses that a product series of probabilities can be written as a sum of the logarithm to these $L(p(y_t, \bar{x}_t))$. This method is often used in optimization problems that use the log-likelihood method. Taking this one-step further by only using the first element in the Taylor expansion⁹, an approximation of the logarithm function can be derived, as shown in the second line in (8.21). Because it is possible to scale this equation by an arbitrary constant and still have a valid expression for the maximum measure, the substitution of multiplication with the summing of the un-scaled probabilities also has a preserved maximum.

8 Even though modern embedded processors have hardwar- accelerated multipliers a multiplication operation uses more resources than an adding operation, i.e., more power is consumed and more implementation logic is allocated.

9 It is noted that the requirement that the Taylor expansion restricts the argument to below one is fulfilled because the argument is a probability.

Thus, the activity prediction algorithm consists simply of adding the estimated probabilities (i.e., the weights multiplied by either a zero or a one) for each sensor event in the event buffer LBT window. This sum is then compared with a threshold and the predefined action is emitted if the value exceeds this. Such a simple procedure is efficient in small low-power microcontrollers, especially because multiplication is avoided (Abdelgawad et al., 2009).

The threshold is calculated by using the same procedure as used in the HL-SHS (Subsection 8.3.4) except that the mean value is found over all the weights which are then scaled by a constant limit to create the threshold. This approach ensures that the threshold is dynamic and thereby, that it tracks the values of the weights.

8.6 SIMULATION AND PERFORMANCE OF THE LL-SHS

A dedicated S.O. agent is simulated by using the essential NB classifier algorithms discussed earlier, i.e., only the binary on-event is used, the event buffer is collapsed into a simple binary vector (8.17), only highly probable probabilities are used (8.19) and finally, the sum form substitutes the product form (8.21).

8.6.1 SIMULATION MODEL

The simulation takes place in a derived Matlab-based LL-SHS simulation model, which are part of the larger model developed and described in (Lynggaard, 2011). Parameter settings for these algorithms are similar to the ones used by the HL-SHS simulator (Section 8.4).

The Matlab-based naïve Bayes simulator consists of different modules that handle the program functionality in a structured and organized way. Thus, the program initializes by loading the dataset (i.e., the dataset to be simulated) and formatting this into a local representation. Such a representation ensures that different data formats are supported by simply building new loading routines. Then, the data are fed to the learning program module that places a multiple window over data that each has a fixed point at its dedicated action. These windows have a time length of the same size as the LBT constant. All actions inside these windows are used to update the weights by adding a one to these, for each instance of the dedicated action. If a particular action is not present in a particular time window, a minus one is added instead. It is possible to set this program module to use a fixed amount of

training dedicated actions. After all weights are updated, they are thresholded such that all negative weights are set to zero, i.e., they are not used as described in equation (8.20).

The left-hand part of Figure 8.18 provides a descriptive pseudo code that covers the main algorithm functionality in this program module. The predicting program module loops through all events by placing them in the event buffer if they are inside the LBT window. Every time that a new event arrives, its buffer representation is set to a binary one. By multiplying this one with their respective weights and then adding gives a score. This score is then thresholded; if it is above, the dedicated action is claimed to be evident. A pseudo code description of this program module is provided in the right-handed part of Figure 8.18. However, it can be noted that this pseudo code illustrates only the main functionality and does not include all the necessary handling, such as scaling and presenting the data.

<pre> loop through all actions and events in data; if dedicated action in data; set all weights W to -1; loop through event buffer; if event E is inside event buffer; set weight for E to 1, i.e., $W(E)=1$; end if; end loop; add W to final weights FW; </pre>	<pre> loop through all events in data, i.e., move eventbuffer one step forward; loop through event buffer B; if event E is inside event buffer; set E in B to 1, i.e., $B(E)=1$; end if; end loop; score $S = S + B(E)*W$; end loop; </pre>
---	--

Figure 8.18. Pseudo code for the essential algorithms in the Matlab based naïve Bayes simulator. Left-handed code is the learning algorithm and right-handed code is the prediction algorithm.

8.6.2 SMART HOME SIMULATION EXAMPLE I

As simulation input to the system, the Kasteren et al. (2008a) dataset has been used. This was recorded in a house with 14 sensors, in which a 26-year-old man lived for 28 days. The reason for choosing this dataset is that it covers a generalizable scenario and because all sensors are placed in logical positions that provide generalizable data. The sensor positions are shown in Figure 8.19. This experiment resulted in 2120 sensor events and 245 activity events or simple instances that are annotated by the user. One of these activities is the “breakfast” scenario,

where the user eats breakfast and annotates this event. This happens 20 times out of the 245 actions.

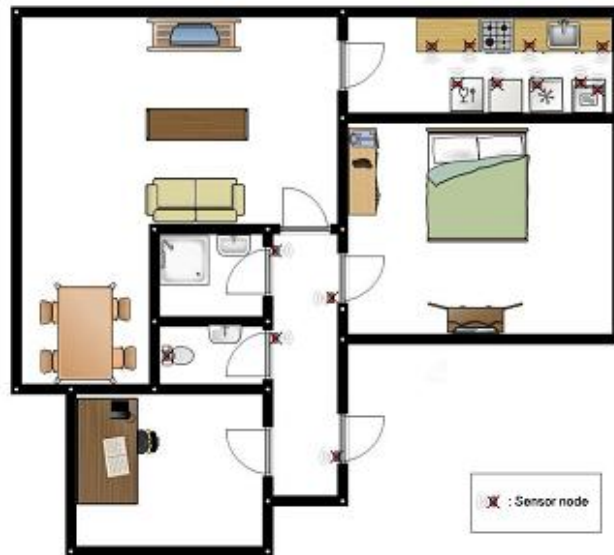


Figure 8.19. The Kasteren et al. (2008a) experimental house equipped with 14 sensors (red crosses).

As described earlier, the common way this dataset has been used in many papers is by training some classifier and omitting the data for one day, which is used subsequently for testing. This approach is not feasible in this work because learning must happen on the fly. Alternatively, the approach of looking at the weight update as a function of time throughout the 28-day period has been used. Thus, some selected action events annotated by the user have been fed through the algorithms together with the raw sensor events. The resulting weights are observed and presented in the following.

By looking at the weights in Figure 8.20 for the “breakfast” scenario (Kasteren et al., 2008a), it can be observed that some sensor weights are negative. This is a purely virtual but informative result of using the balance algorithm, i.e., it increments or decrements weights depending on the hit or miss rate. Thus, from the “breakfast” scenario example, it can be seen that some weights are not carrying any information, which means they are given a large negative value (saturated). The most positive weight is number 12 (dark green curve), it hits 16 times but misses 4 times (out of 20 runs). Thus, the weight for sensor number 12 is incremented 16 times, but decremented 4 times; therefore, its total value is 12.

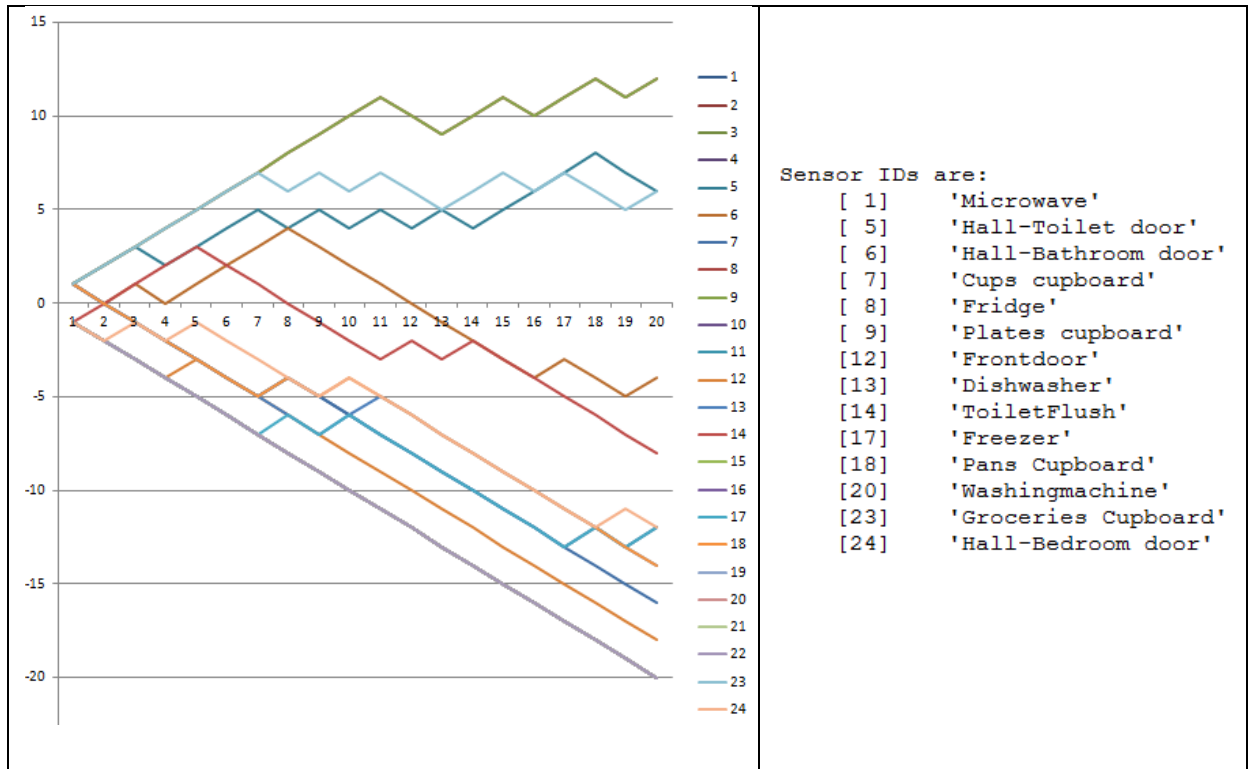


Figure 8.20. Left: the breakfast scenario weights update count (y-axis) as a function of seen user-annotated action events (x-axis). Each color curve refers to a sensor number in the color scheme used.

Right: sensor identity for color code in left-handed figure.

From the figure, it is obvious that four weights (one is covered behind the green curve) are dominant, i.e., they are above the selected threshold limit. These weights are sensor numbers 8 and 9 for the highest weights (dark green curve), 5 (dark blue), and 23 (light blue) for the next highest, respectively. Translating these numbers into sensor placement shows that 8 is the fridge, 9 is the plate cupboard, 5 is the toilet door and 23 is the grocery cupboard (Figure 8.20). Thus, conclusive good agreement is found, i.e., it is expected that sensors for the fridge, cupboard, and grocery cupboard would be activated when the user prepares breakfast. Surprisingly, the toilet door sensor is also activated when making breakfast. This most likely indicates that human habits are a strong factor and therefore, ignoring connections between states wastes correlation information. Actually, this is the main disadvantage in using the NB classifier as discussed in subsection 5.5.2. From these four sensors, it can be observed that the “breakfast” classifier averages out the useable sensors after approximately 10 actions (Figure

8.20). It can also be observed that to some extent, these weights saturate. This means that a fixed signal-to-noise ratio exists, i.e., random user behavior overlies the human habits (Kasteren et al., 2011).

The score for this scenario is found by using the partly trained NB Matlab-based model, i.e., part of the data has been used for training, whereas the full dataset has been used for predicting. Because this approach includes the training data in the predicting dataset the scores will be slightly higher compared with a method that uses two separate sets. Therefore, by letting the model learn from this reduced dataset and afterwards running it through all 20 annotated “breakfast” actions the results in Table 8.6 are obtained.

Rating	Score after training round (percent)		
	5	10	20
TP	35	65	70
FN	65	35	30
FP	0	30	30

Table 8.6. Algorithm score as a function of training times for the breakfast scenario. The ratings are divided into TP (true positive), FN (false negative) and FP (false positive).

By looking at the numbers in Table 8.6, it can be seen that the TP rate increases quickly in the beginning where few training actions have been seen. Then, it settles in an asymptotic manner when further training is provided. This behavior is in good agreement with the progressing of the training weights as illustrated in Figure 8.20. It is also noted that the FP rate settles quickly as a function of the used amount of training data, i.e., it reaches the noise floor where all non-temporal correlated information has been used.

8.6.3 SMART HOME SIMULATION EXAMPLE II

This example uses the smart home from subsection 8.6.2, but with a different dataset (Kasteren et al., 2008a). The reason for choosing this dataset is that it supplements the dataset used in subsection 8.6.2 without being coupled to it directly. Thus, it provides a second independent dataset for performing uncorrelated testing in the same environments. In addition, this dataset is generalizable for many similar settings.

The dataset covers a “bathroom” scenario where it is detected whether the user has been there. Hence, if this has been detected a bathroom action is emitted. By learning this action, it is possible, for example, to set the hall to a “night lightning” state, which enables the user to return to bed safely and afterwards returns it to the “night sleep” state, i.e., all unnecessary lighting is turned off and alarm and fire detector systems re-enabled. However, detecting this action is general in nature, which is why it is usable in many scenarios.

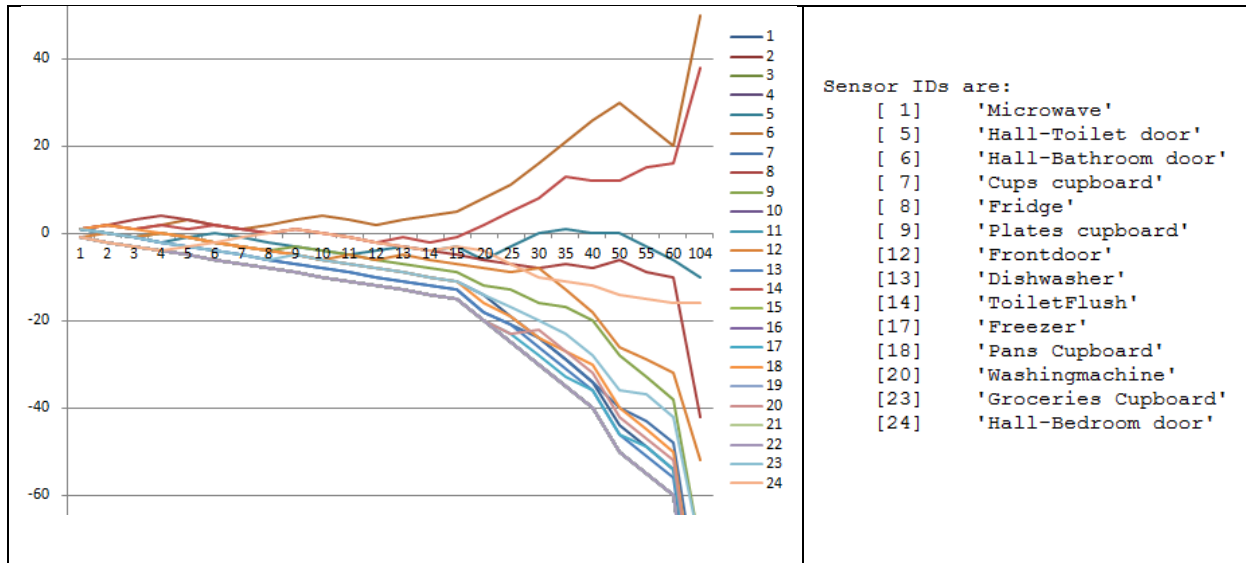


Figure 8.21. Left: the bathroom scenario weights update count (y-axis) as a function of seen user-annotated action events (x-axis). Each color curve refers to a sensor number in the color scheme used. Right: sensor identity for color code in left-handed figure.

By comparing the “breakfast” scenario in Figure 8.20 and the “bathroom” scenario in Figure 8.21 scenarios similarities can be noted. Thus, it is noted that some weights stand out when the number of user-annotated actions go beyond approximately 10. These are sensor 6 (light brown, toilet door), which has the highest value and sensor 14 (dark brown, toilet-flush). Correlation is observed between the sensors that stand out and the action “bathroom”. The NB algorithm estimates the actions correctly (TP) approximately 66% of the time. It failed (FN) to estimate the annotated user actions 34% of the time. The user does not see this as a serious problem as discussed in subsection 8.4.3. Similarly, FP outcome (21%) is annoying for the user (Subsection 8.4.3). The FP rate can be reduced by either adjusting the classifier parameters or by using additional sensors to provide finer granularity of information produced by the user’s actions (Fang et al., 2012).

The same experiment as the previous ones was performed, but with an attempt to detecting a “leaving home” situation from the sensors by using the same setting as discussed earlier. In this scenario, two sensors: 12 (Front door) with the highest score and sensor 5 (Toilet door) stand out, and the final scores for this experiment are that the TP rate is 82 %, the FN rate is 18 % and the FP rate is 21 %.

Table 8.7 compares the work done by Kasteren et al. (2008a) with the results of this work.

Algorithm	Scenario		
	Leaving	Breakfast	Bathroom
naïve Bayes (this work)	82	70	66
HMM (Kasteren et al.)	98	56	73
CRF (Kasteren et al.)	91	57	55

Table 8.7, Comparing real-time score between naïve Bayes (this work), HMM (Kasteren et al. (2008a) Table V) and CRF (Kasteren et al. (2008a) Table VII).

Table 8.7 shows that the results for the naïve Bayes simplified approach used in this work are comparable with the more advanced classifier technologies. Thus, for the “breakfast” scenario, a benefit is actually achieved by truncating the low score weights. This is probably the case because it eliminates non-correlated sensors, i.e., it improves the signal-to-noise ratio. Regarding the “Leaving” and “Bathroom” scenarios, the HMM benefits from using the temporal correlation information contained in the previous action as discussed in section 8.2. The CRF classifier scores a little worse than the HMM, which is probably because it uses a single model for all activities where each activity competes during maximization, i.e., it is possible to maximize around a dominant activity and ignore minor frequent activities.

Another noteworthy comparison is against an unsupervised approach (Section 5.4), e.g., using the k-nearest neighbor classifier. This work has been done by Fahim et al. (2010) who achieved an average score of 71% when using their own training set. Thus, the scores of this work are comparable.

8.6.4 LL-SHS SUMMARY

By using a combination of a naïve Bayes probabilistic framework and sensor inputs provided from simple binary switches, a simple and robust SHS model has been developed. This

system is able to suggest actions to the user that it has learned from previous user behavior patterns in a real-time fashion.

It has been found that the explored LL-SHS model has a performance close to that of the more advanced systems based on the HMM and CRF. Furthermore, the presented system reduces the complexity level, sensor events demand and processing load, compared with the more advanced systems. This facilitates its implementation in a distributed manner, on simple low-power microcontroller-based devices.

In addition, it has been found that in spite of a reduction of the quantity of sensor information (fixed sample time and missing off-states) the proposed system still performs very well. A consequence of this reduction is that the network load is reduced, which facilitates employment of simple low-power network topologies and supports the use of simple low-power sensors.

The future perspective of this work is to investigate the possibility of implementing probabilistic classifiers that are more complex with simple algorithms and to explore their results with simple binary sensor events. Finally, the suggested SHS needs to be integrated with a user-friendly interface for interacting with the S.O.s.

8.7 SMART OBJECT MODEL, REVISITED

As discussed earlier the LL-SHS consists of a collection of S.O.s in which each S.O. contains a collection of interfaces and functionality. From an AI point of view, an S.O. contains a collection of agents, where each agent is able to learn one dedicated action that is pre-assigned to it. An example of such an S.O. could be a table lamp where the dedicated actions are light on, light off, and dim light. This requires three agents that are able to learn this behavior from the user's habits. Thus, these three agents are allocated in one S.O. that is placed inside the table lamp.

The naïve-Bayes-based AI system contained in the agents is described in the section 8.3 and will not be discussed further. Instead, this section covers the implementation and performance of the S.O.s.

As stated and discussed earlier, the main problem addressed by this work is the reduction of the power consumption, the delay and the network load within a smart home by deploying a distributed universal S.O. concept. A suggested solution is given in the form of a distributed embedded S.O. architecture that uses general software elements in combination with specific agent-related software elements. These agent-related software elements are the most complicated and resource demanding part in an S.O., i.e., if these elements can be implemented, the other software parts will be manageable. Thus, the agent-related software elements are investigated and simulated in a real-world embedded context.

The recorded dataset, which is publically available from the Milan CASAS experiment, was used for the simulations performed in this work (Cook, 2012). A floor plan of their experimental smart home is provided in Figure 8.22. This dataset contains sensor data collected in a smart home, in which a volunteer adult woman lived for three months. She had a dog and she was visited by her children on several occasions. During the period, 2310 annotated actions and 433656 events were collected.

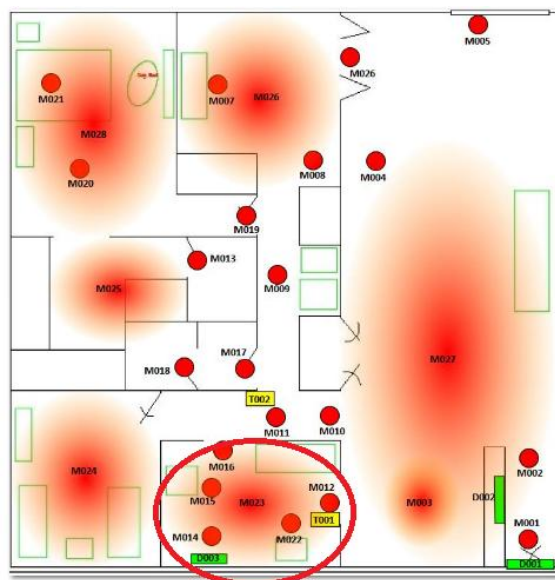


Figure 8.22. Smart home used in the Milan CASAS experiment, 2009 (Cook, 2012).

The red circle located in Figure 8.22 encloses all the kitchen binary sensors (M012, M014-M016, M022, and M023) that are combined and simulated as one S.O. agent in this work. Therefore, by looking at the average number of events emitted by these five sensors and

comparing this with the number of S.O.-emitted actions it is possible to calculate the power and network load savings. A detailed discussion of these savings is provided in Lynggaard (2013c). As discussed, the sensor event information is provided from a CASAS smart home project dataset.

8.8 SIMULATED PERFORMANCE OF THE SMART OBJECT MODEL

An S.O. agent based on a simplified naïve Bayes classifier with five sensor inputs is constructed in C-code and simulated using MPLAB v1.51 from Microchip (Figure 8.23).

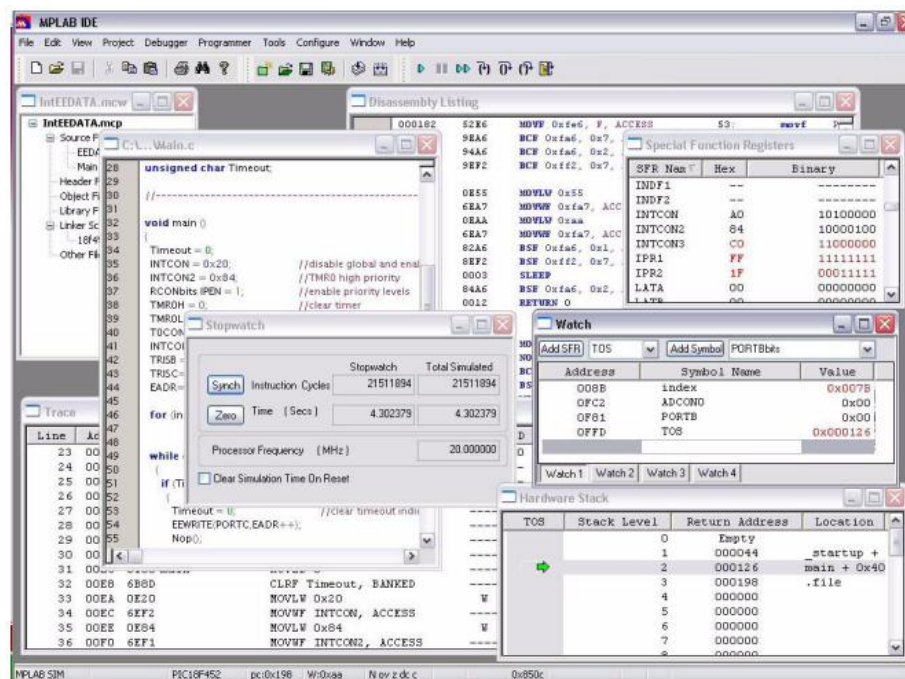


Figure 8.23. MPLAB simulator (Microchip) screen view.

This C-code is compiled for the simple state-of-the-art low-power microcontroller PIC18F46J50 from Microchip (schematics in Figure 8.25) without code optimization (Subsection 4.4.5). The controller uses extreme low power and is state-of-the-art in its field¹⁰. In this work, it is assumed to run with a clock speed of 1 MHz and consume 1.2 mA when it runs and nothing else. Using this microcontroller family has the following advantages: it is a

¹⁰ <http://www.microchip.com/pagehandler/en-us/technology/xlp/products.html>

cheap consumer product, it is supported by a comprehensive tool suite, and it offers very low power consumption, i.e., it can run from months to years on a small battery depended on the processing load. In addition, it includes a variety of sleeping and processing modes together with built in hardware elements, such as timers, wakeup on port change interrupts, etc. All these features are useable in an S.O. component.

The software is implemented using C-code compiled by the Microchip XC8 compiler without its optimization features enabled. This means that the presented results can be improved further. However, the actual implementation of the software uses some optimized architectural design. The implemented software including some test scripts, is executed in the MPLAB v1.51 simulator. Using this environment makes it possible to predict processing load and thereby, to calculate indirectly the consumed power. All tools are available from Microchip.

Implementation of the naïve Bayes based agent Figure 8.16 can be divided into two software parts: an event First-In-First-Out (FIFO) buffer, and a summed weight multiplication part (Figure 8.24). Regarding the FIFO, it uses a circular buffer approach, which optimizes its efficiency.

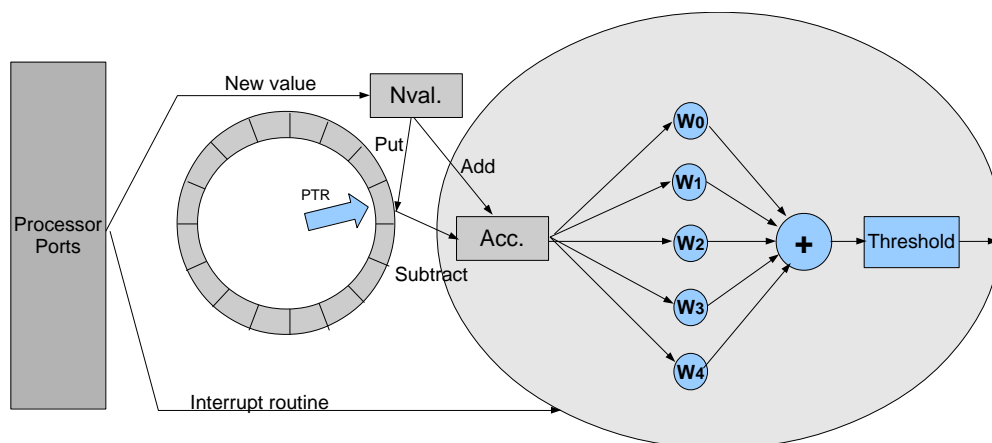


Figure 8.24, naïve Bayes embedded implementation model.

The FIFO buffer contains a pointer “PTR” that always points to the oldest value. When a new event arrives, the interrupt routine in the microprocessor is activated. The interrupt routine handles a sequence of steps:

- It fetches the new event and stores it in “Nval”.
- It fetches the oldest event from the circular buffer (pointed to by “PTR”) and decrements the event counter (contained in “Acc”) for that particular event.
- It then adds the new event “Nval.” to the event counter, i.e., the “Acc”. Thus, the accumulator keeps track of how many instances for each sensor are present in the FIFO buffer.
- It puts the new value into the circular buffer at the “PTR” position, i.e., it overrides the oldest value.
- It then increments the pointer value to the next field, i.e., the (new) oldest value in the circular buffer.
- It multiplies the accumulator with the weights and sums the products. Actually, it sums the weights that are multiplied by a binary one, i.e., no multiplication takes place. This provides efficiency.
- It compares the summed weights with the threshold and emits an action if it is exceeded.

This concept has been optimized for time so instead of traversing the full event buffer which takes 38.5 mS for 1800 values (simulated values), it use the accumulator balancing approach code discussed earlier. This occurs whenever an event arrives, i.e., it generates an interrupt, which runs the discussed code in 0.630 mS. Thus, a saving of approximately 98 % is achieved compared with the “traversing the full event buffer” concept. Added to this is the weight summing of the five weights (Figure 8.22) that in total uses 0.346 mS. Therefore, in total, the processor load for performing one agent prediction calculation based on five sensors is 0.976 mS. Assuming the sensor sampling frequency is 1 Hz and that they fire asynchronously the total processor load is below 0.5 %.

It is noted that this is the maximum load that happens when the sensors are changing all the time, but this is seldom the case in a real-world scenario. As an example this means that 100 S.O. agents each connected to five sensors will use 10 % of the processing resources. Because this is the main activity in an S.O., it also means that the processor will be able to save power

by sleeping for approximately 90 % of the time¹¹. As noted, these calculations are performed for the S.O. action-prediction process because it is more resource consuming than the learning process, which only updates its respective weights (Subsection 5.5.2).

As discussed earlier (Section 4.4), many SHSs use sensors that transmit wireless events and often, these systems are based on a concept that uses low-power ZigBee transceivers (Li et al., 2011a). Thus, in this section it is assumed that the SHS calculations for the sensors and S.O. nodes use the popular CC2420 ZigBee communication unit from Texas instruments (Yan & Dan, 2010), (Park et al., 2013).

In a traditionally centralized setup the five sensors in the CASAS smart home kitchen (Figure 8.22), (Cook, 2012) must transmit events every time they are triggered. Therefore, finding the cost of transmitting one sensor event and multiplying this by the average number of sensor events from these five sensors give the total cost per chosen unit time. Thus, an estimate of the average number of sensor events can be calculated by using the CASAS dataset and looking up the five sensor events over a time unit. It has been found that the average number of sensor events from these five sensors on a daily basis, is 1795 events.

Based on the calculated number of events the energy consumption can be found. It is assumed that only a three byte sensor identity is sent in each event and that the ZigBee transmitter uses 32.5 mA for the carrier sense multiple access sequence of length 2.9 mS, 13mA in 13 mS for activating the microcontroller, and 30.5mA in 1 mS for actually transmitting (Casilari et al., 2010). Adding up these contributions and multiply it with the average number of event and a 2-volt supply voltage (minimum operation voltage for the PIC18F46J50 controller) gives an energy consumption of 1.055 Joule per day. Assuming the sensor nodes are power by a standard LR416 battery that offers approximately 43 Joule means that these five sensors in total will use approximately 1 battery per 40 days.

However, using an S.O. to handle the five sensors as discussed earlier results in a power consumption of 1.2 mA for the processor running in 0.976 mS, which yields 1.17 uJ per

¹¹ It is assumed that wired or wireless communication is handled by an external chip connected to the processor.

event. Therefore, processing 1795 events costs 4.2 mJ. Assuming that the S.O. sends all the average detected kitchen actions (approximately 7 per day in the dataset) to the HL-SHS means that the cumulative transmit consumption is 4.1 mJ. The final S.O. consumption is 8.3 mJ per day.

Regarding the network load the number of ZigBee frames or Internet packets if the SHS is located in the cloud is reduced from 1795 to 7 per day. These results are summarized in Table 8.8.

Quantity	Average consumption	Network packet load
Energy, traditional solution (transmitting 1795 events per day using Texas ZigBee module CC2420).	1.06 Joule per day	1795 per day
Energy, this work (using the S.O. concept transmitting 7 actions per day using Texas ZigBee module CC2420).	0.0083 Joule per day	7 per day

Table 8.8. Comparing the traditional sensor framework implemented in the CASAS kitchen smart home with the same one that uses S.O.s to provide the savings shown.

The results in Table 8.8 show that a noticeable saving in both processing power and network load has been found. However, using the S.O. concept also has a downside, because consuming the events in such a distributed approach means that they will only arrive in the more centralized HL-SHS as actions and not as detailed events. This means that the detailed granularity and correlated information in the event information is lost. Nevertheless, as discussed earlier, this loss of information does not reduce performance noticeably (Section 8.6).

Another simplification that needs attention is that this work assumes that only one action is detected by the kitchen S.O. agent. However, transmitting all events to a centralized system opens up the possibility of detecting further actions based on these. Nevertheless, this does not change the overall conclusion, because using the HL-SHS with actions only as discussed earlier still provides comparable results when compared with similar systems that use a traditional approach (Section 8.4).

The network load saving will be slightly lower than stated because the distributed S.O.s need a centralized management system to collect the suggested actions and to enable the user to

update the S.O.s built-in calendars. In contrast, the saving can be much larger because some of the routing ZigBee nodes could be battery powered, which means that they need to transmit 1795 events, as opposed to 7 when using the S.O.-based concept.

Also noticeable, is that the overall battery saving achieved by using S.O.s is not linearly scalable with the used number of S.O.s. This is because some of these will be implemented in equipment that is supplied from the mains grid.

As discussed, some optimizations have been made for facilitating implementation of an S.O. on an embedded platform. Firstly, the sensor events are defined to be binary, i.e., their values can only be zero or one. Secondly, the weights in the used NB classifier are un-scaled probabilities, i.e., they are positive and negative integers only. Thirdly, the multiplication required to calculate the final product of the observed and estimated sensor probability $p(x_t^n | y_t = i)$ is approximated by a sum. As discussed these optimizations only have a minor impact on the learning and predicting performance (Sections 8.4 and 8.6). Nonetheless, when looking into the savings of processing resources in an embedded implementation a significant difference is found. By using the MPLAB simulator and running a C-code-based implementation of an NB classifier, it is possible to compare the outcome in the two cases, i.e., with and without the optimizations. Therefore, these simulations are performed by using the same conditions and settings as previously discussed and additionally used to derive Table 8.8.

Based on the processing calculation time, it is possible to determine the used energies in the two cases. Thus, the non-optimized NB algorithm uses a processing calculation time of 17.53 mS that should be compared with the earlier discussed results of 0.976 mS for the optimized algorithm. By using the previous discussed power consumption numbers, including 1.2 mA for the processor, the final energy saving has been found (Table 8.9).

Quantity	Average consumption
Energy consumption non optimized NB algorithm and non optimized buffer with 1800 places.	0.243 Joule per day
Energy consumption non optimized NB algorithm, circular optimized buffer with 1800 places.	0.080 Joule per day
Energy consumption optimized NB algorithm, circular optimized buffer with 1800 places.	0.0083 Joule per day

Table 8.9. Energy consumption for optimized and non-optimized naïve Bayes and buffer algorithms, simulated on a PIC18F46J50 device by using the MPLAB simulator from Microchip. It includes 5 sensors, 1795 events and 7 actions that are transmitted per day using the Texas Zigbee module CC2420.

As illustrated in Table 8.9 the energy saving for the optimized algorithm and buffer concept compared to the non optimized algorithm and buffer concepts is approximately 96 %. In addition, it is noted that these savings do not have significant impact on the algorithm performance, as shown in Table 8.7. However, for these calculations, it is assumed that the microcontroller and the ZigBee module are able to power sleep when they are passive. This is a reasonable assumption because the chosen clustering concept (Subsection 4.4.4) enables the use of transmit-only sensors, so the battery-powered sensors do not need to run the receiver all the time. Regarding the microprocessor it only consumes a few nano-amperes and is able to wake up quickly, relative to the sensor sampling rate, based on a timer or input change interrupt.

As discussed in section 7.2 an S.O. contains a collection of smart agents in which each of these implements an NB classifier. Thus, an S.O. contains many instances of NB classifiers. This is necessary because each agent is able to handle only one action, e.g., a vacuum cleaner equipped with an S.O. needs one agent for each logic function it offers, such as “start cleaning living room”, “start cleaning hall”, “stop cleaning”, etc. In general it is found that at least one agent is needed for each logic state found in the smart home device. Actually, more than one agent per state is needed because a particular action covered by a particular state could be activated more than once during the same day.

To obtain a quantitative measure of what is possible, it is assumed that each agent can handle up to S sensors and that it offers a time resolution of R . Therefore, to clarify the resources that are needed for an embedded implementation it is necessary to look into the processing power

and the amount of memory needed. It can be assumed that each weight is limited to a signed byte, one byte is used for the action name (allows 256 different actions), two bytes are allocated for threshold limit, and that five bytes are used as bit flags, etc. for controlling behavior. Then expressing this as an equation gives: $M = (S \cdot R + 8)A$ where M is the number of bytes needed for A agents. The PIC18F46J50 from Microchip offers 3776 bytes of static memory. Allocating 1000 bytes for the agents leave 2776 bytes for the other S.O. software items. By using this number and assuming 8 sensors and 24-hours resolution per day, the formula yields 5 agents. Hence, an example could be a lamp with an on-state and an off-state, i.e., two agents are needed. Another example could be 5 sensors, 1500 bytes of memory, and 5 agents which provide a time resolution of approximately ten minutes. Regarding the processor calculation resources, it was found earlier in this section that 100 agents use approximately 10 % of the available processing power; therefore, this is not the primary limiting factor.

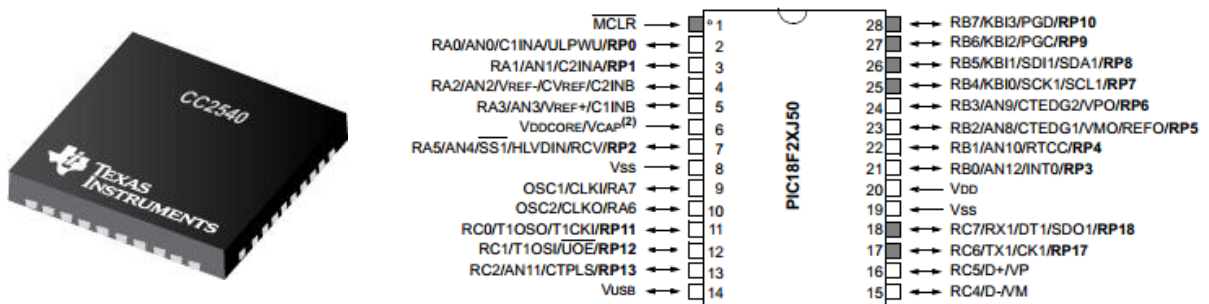


Figure 8.25. Left: the CC2540 integrated Bluetooth chip from Texas instruments. Right: the PIC18F26J50 schematic.

Other wireless technology such as the Bluetooth low energy (BT-LE) has approximately the same energy consumption as ZigBee. Texas Instrument have a BT-LE device named CC2540 (Figure 8.25), (Kamath & Lindh, 2013) that uses approximately the same amount of energy in the transceiver; however, for the built-in microprocessor no data are available from Texas Instruments (Greenja, 2013). Nonetheless, by assuming that the built-in microprocessor consumption is almost the same as for their ZigBee variant CC2420, it is concluded that the previously presented energy calculations will be similar if this device was used instead.

8.9 SUMMARY

A mathematically based distributed AI framework fitting smart home system architecture was derived. This framework is hierarchically distributed, i.e., a low-level part is dedicated to the S.O.s and a high-level part fits a server or cloud-based framework. It is mathematically founded so learning and prediction behaviors are described in these terms.

The derived framework was modeled numerically by implementing it in software and simulated by using publically available datasets from different researchers. The simulations revealed that the derived distributed systems perform similarly to other comparable systems. A key part is the S.O. that has been researched and explored by designing and implementing its vital software elements on a state-of-the-art embedded microcontroller. Significant savings were achieved in the implementation structures by using derived algorithms, structures, and strategies.

In general, the simulated results revealed that huge savings were achievable in both battery power consumption and in network load. These saving were calculated by using results from comparable state-of-the-art research.

9 OVERVIEW OF THE FINAL CONCEPT AND CONTRIBUTIONS

This chapter provides an overview of the suggested system and it lists my original contributions to the respective research areas. The overview is divided into two parts where the first part (Section 9.1) provides some scenarios that illustrate the system interaction and the responding system behavior. Section 9.2 presents a picture of most parts in the system and gives a short description of the parts. Following this is my original contribution in section 9.3.

9.1 A HIGH-LEVEL OVERVIEW

An overview is presented in the following subsection 9.1.1, and in form of two examples in subsection 9.1.2. Other application examples are provided in subsection 9.1.3. The content in this section is available in elaborated form in Lynggaard (2012a).

9.1.1 THE SYSTEM

This section presents an overview of the suggested SHS, where the presentation level is kept at a high abstraction level to prioritize focus on its basic functionality.

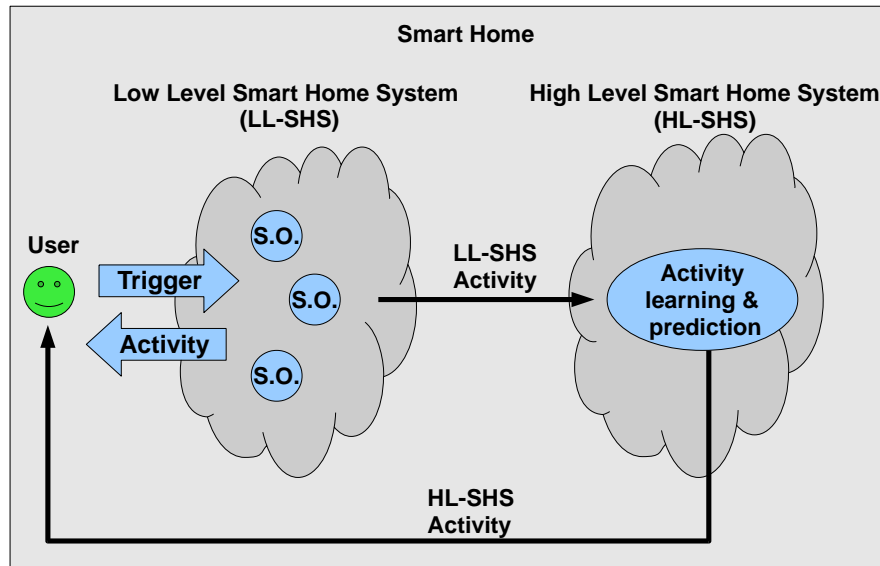


Figure 9.1. An overview of the presented smart home system.

Figure 9.1 illustrates the components in the presented SHS. Focusing on the S.O.s they are combined into the LL-SHS (Section 7.1). This LL-SHS label has been added to indicate that this system can exist as a standalone system implemented on embedded S.O.s or on other devices equipped with equivalent processing power. An S.O. based standalone system is independent of server availability, which means that the network of S.O.s is self-sustaining and can be used as a minimalistic SHS.

Because the LL-SHS is minimalistic in many ways it needs the HL-SHS for more advanced data processing (Section 7.1). The HL-SHS cannot be used as a standalone system because it requires some preprocessed event based information. This information can be made available by either the LL-SHS or by some similar systems as long as the semantics and interfaces are compliant. It is noted that the HL-SHS can be allocated either on a local server or as a cloud based service.

Focusing on the system functionality, the user (green figure in Figure 9.1) makes some activities in the smart home. These activities are monitored and captured by the sensors interfacing the S.O.s. Based on the “triggered” sensor events and its learned behavior the S.O.s are able to predict activities and suggest services to the user. In addition, it can pass the

predicted actions to the more advanced HL-SHS, which is able to predict and suggest more advanced activities (Section 8.5). The feedback to the user can come by two different paths, one path from the LL-SHS and another from the HL-SHS. These paths indicate that both systems are independent and that they work in a parallel approach.

9.1.2 APPLICATION EXAMPLES

To clarify and illustrate these ideas some scenarios are presented in the following. The first scenario is the “Bob scenario”, which illustrates the LL-SHS behavior. Following this is the second scenario named the “Alice scenario”, which illustrates the use of a combined system including both the LL-SHS and the HL-SHS.

Bob scenario:

Bob enters the kitchen because he is hungry, so he fetches a plate from the cupboard and sits down at the dinner table. When he does this the kitchen light shifts automatically so the dining table light is switched on and set to a colour that visualises the food in a nice way...

Even though this scenario seems simple, a lot of complex processing and communication is going on behind the scene. Firstly, the scenario needs to be learned by the AI system that adapts and tracks the user behavior in a real-time fashion. Secondly, the artificial system needs to be able to predict the user activity pattern and suggest actions based on this.

To illustrate these parts, i.e., the learning and prediction sequences a sequence diagram covering the scenario is shown in Figure 9.2.

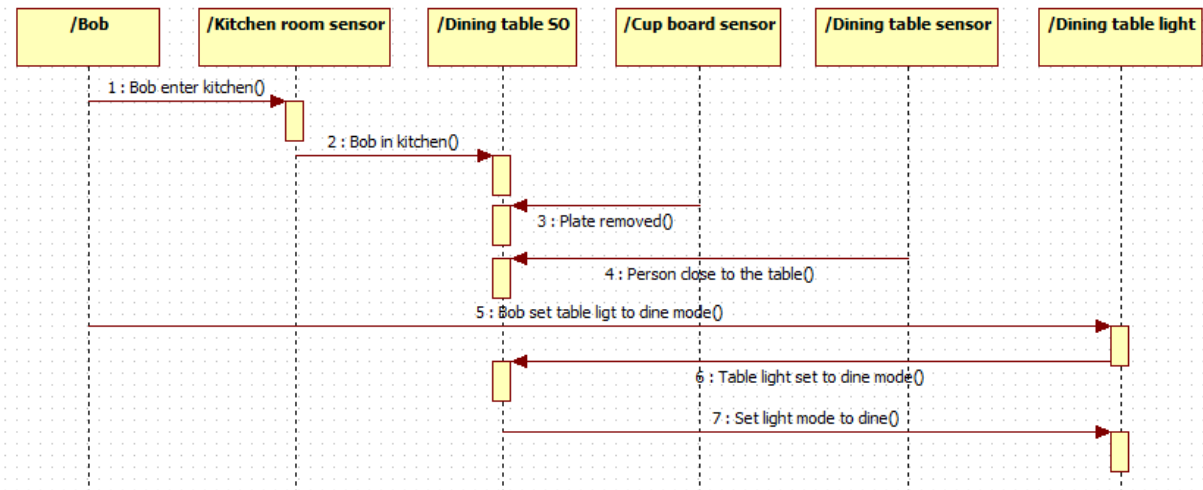


Figure 9.2. The Bob scenario - sequence diagram for smart object learning and predicting.

The flows in Figure 9.2 are:

1. Bob enters the kitchen
2. Kitchen room sensor detects his presence and notifies the dining table S.O..
3. When Bob takes a plate and moves it to the dining table, the cupboard sensor notifies the dining table S.O..
4. Bob sits down at the dining table; this is notified to the dining table S.O. by the dining table sensor.
5. Bob uses his smart phone to remote control the dining table light.
6. This action is observed by the dining table S.O. that learns this behavior from the past sensor events and this user performed action.

From this scenario the LL-SHS (it contains only one S.O. in this example) learning and prediction processes can be elaborated. Learning takes place when the user performs an activity like turning on the room light. This action is captured by the connected S.O. that uses this to update one of its contained agents by looking into the events that have arrived in the past. The prediction process works the other way around, i.e., the agent monitor the incoming events and if some degree of match relative to the learning events can be found the action that triggered the learning in the first place is suggested to the user. Flow 7 in Figure 9.2 illustrates this, i.e., the table light is set to dim when the events described in flow 1-4 has been

performed. If the user disagrees to this activity the suggested activity can either be removed in the user interface or unlearned by doing the reverse action, i.e., switch off the table light.

It is noted that the actual learning and prediction processes are stochastically based. This adds some complexity to the scenario in the form of delaying the learning and prediction process, i.e., the system needs some training times to learn (Chapter 5) – just like a human.

To illustrate and provide an overview of the role that HL-SHS plays in connection with the LL-SHS a scenario that involves both systems in a cooperative manner is presented.

Alice scenario:

The time is almost 6.00 in the morning and Alice leaves the bedroom after a good night's sleep. She goes to the bathroom for a morning shower. After finishing her bath she goes to the kitchen for breakfast. She looks at her watch, it is close to 7.15 so she finishes her breakfast and leaves the kitchen. In the hall she puts on her coat and leaves the home for going to work. On her way to the bus she uses her smart phone to set the home to hibernating state, but then she remembers that the smart home system already has done this.

Analyzing this is performed by using a sequence diagram as illustrated in Figure 9.3.

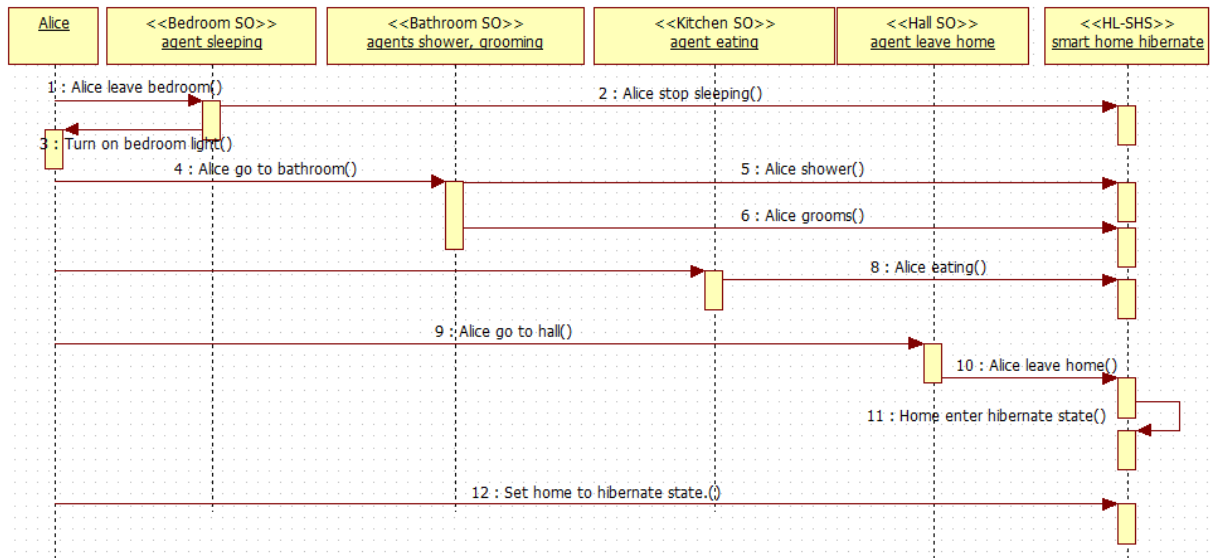


Figure 9.3. The Alice scenario uses a combination of the LL-SHS and the HL-SHS.

The flows for the LL-SHS and the HL-SHS interplay, learning and adaptation processes are:

1. Alice leaves the bedroom.
2. The bedroom S.O. runs different agents where one of them is agent sleeping. This agent emits the action *Alice stops sleeping* which is captured by the HL-SHS.
3. The bedroom light is turned on by agent sleeping
4. Alice goes from the bedroom to the bathroom.
5. The bathroom S.O. runs the agents shower and grooming. Agent shower emits the action *Alice showers* to the HL-SHS.
6. Agent shower contained in the bathroom S.O. emits the action *Alice grooms* to the HL-SHS.
7. Alice leaves the bathroom and goes to the kitchen
8. The kitchen S.O. contains agent eating that emits this action to the HL-SHS.
9. Alice leaves the kitchen and enters the hall for leaving the home.
10. Agent leave home contained in the hall S.O. emits action *Alice left the home*.
11. The home enters hibernate state.
12. Alice sets the home to hibernate using her smart phone.

As illustrated by this scenario Alice's behavior pattern is captured by the low-level S.O. agents who notify the HL-SHS. This system learns from the user action flow 11 and the

previous user action flows 1,2,4-10. Based on this learning it is able to predict when to set the home to hibernate state and it offers this as a service to the user - flow 12. From the scenario it is noted that flow 3 is an action offered directly from the S.O. agent locally. Hence, Alice's behavior pattern consists of a sequence of actions that Alice performed as a function of time. This action sequence is fed to the HL-SHS that runs a higher level of learning and prediction algorithms.

From a logical point of view the prediction probability must be higher if all the information like time, sequence and activity type is used in the prediction process. Actually, Shannon's information theorem supports this point of view by stating that the entropy (quantifies the uncertainty in predicting the value of a random variable) will be lower, if more is known about this pattern sequence. So it can be stated that using all the available uncorrelated information provides a better prediction probability. Hence, this fact and the use of temporal behavior enables the HL-SHS to predict complicated patterns and sequences relative to the more simple LL-SHS.

To reflect on this statement, imagine that only low-level prediction systems (LL-SHS) are used in the just discussed Alice scenario. This means that one of the S.O.s needs to predict that Alice left the home. Assuming that the Hall S.O. would be the one in charge, it needs to base its decision on available information, i.e., its contained agents must perform the prediction, but these agents work locally for the hall and only get information from sensors positioned in the hall. So, it is possible to detect that Alice puts her coat on (coat rack sensor) and leaves her home (door sensor). However, Alice could leave her home for many other reasons than going to work, maybe she went out to empty the trashcan or empty her mailbox etc. Letting the home enter hibernating state in these cases would be annoying for the user and wasteful from an energy and pollution point of view. So, anchoring the hibernating state decision on the sequence presented in the Alice scenario and add factors like time of day and today's date, etc. provides a more reliable decision foundation.

9.1.3 OTHER APPLICATION EXAMPLES

Other application examples are listed in Table 9.1. They are provided in a simplified overview form. However, the possible usability areas are much broader.

Task	S.O. activity
Home appliances	E.g., washing machine / dish washer. Monitors when electricity is cheap and the machine is ready. Then automatically starts it. Additionally, it offers remote control services.
Green technologies	Monitor and learn user's behavior which is autonomously replayed. An example could be the HVAC system.
AAL supervisory	Learn the status of home appliances as a function of time. Thus, abnormal behavior can be reported to the user or to the relatives.
Telemedicine	Add context status to expert systems. An example is a heart-rate monitor which can get information from the patient's environments such as room sensors and other activity sensors.
Security	Learn the user's habits and inform the user (e.g., a smart phone message) if abnormal sensor activity is detected.

Table 9.1. Examples of other smart object applications.

In general, it is noted that the S.O.s in combination with the HL-SHS offer a general framework which can be integrated into many of the devices found in a home today. This concept offers the benefits from the home automation area in the form of timer / calendar triggered actions and remote control (Section 3.2). In addition, it offers the smart homes features in the form of AI and real-time learning (Section 3.3).

9.2 A DETAILED OVERVIEW

This section provides a detailed discussion of each element included in the SHS as illustrated in Figure 9.4.

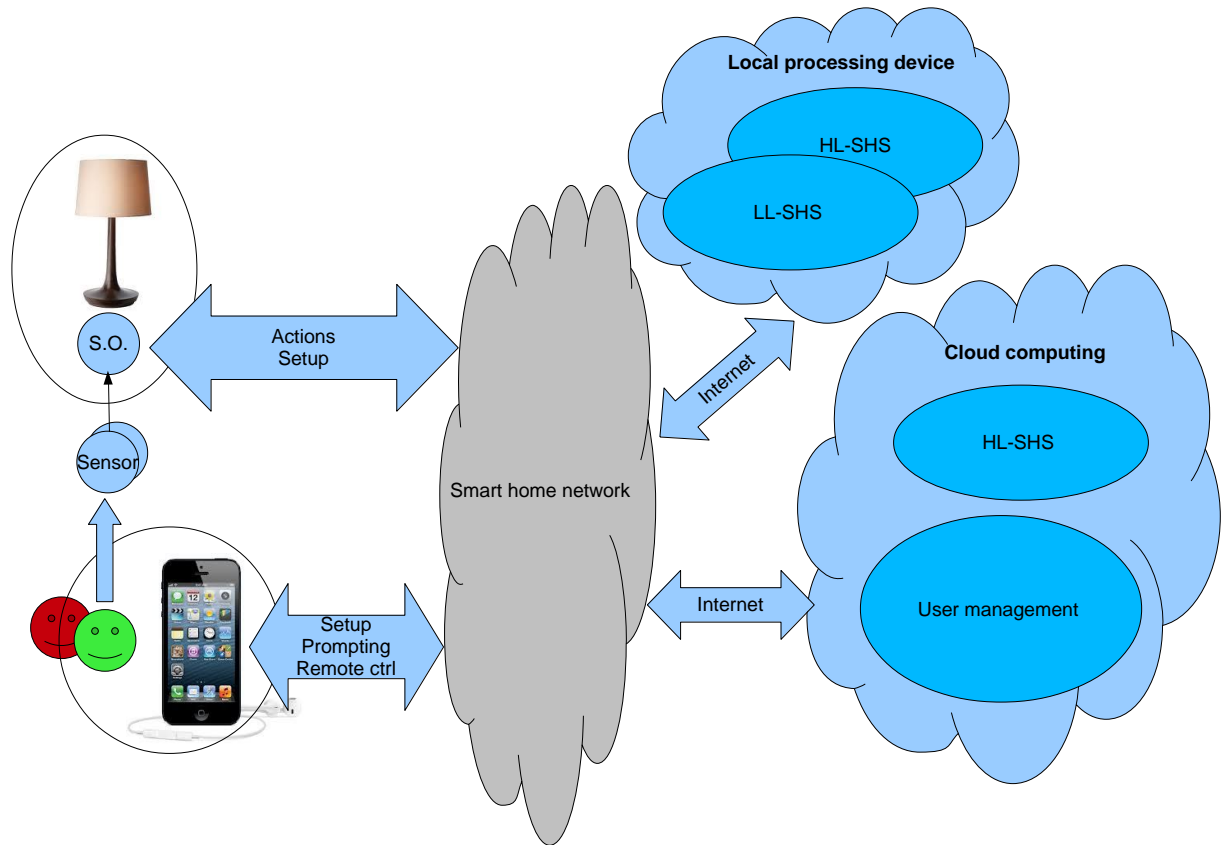


Figure 9.4. The presented smart home system at a detailed level.

9.2.1 USER INTERACTION

Some user interface ideas and an overview are presented in this section; however, it is noted that the user interface is outside scope for the work (Section 1.3).

Starting with the user (green/red icon) a portable user interface is added to the SHS exemplified by the smart phone in Figure 9.4. This enables the user to interact with the system in an easy and comfortable way. Most people are familiar with mobile phones and many of them are also familiar with the more advanced smart phone concept. From a usability point of view the smart phone is a good choice for a smart home user interface platform (Section 3.2). However, the presented SHS is not limited to interact with a smart phone only it is also possible to use other devices such as Ipads and PC's. This flexibility is provided by using the Internet as a backbone. Thus, connecting the Internet to the SHN means that the user can access it from a smart phone. Additionally, many home entertainment devices such as smart

TV, music players, and NAS devices are connected to the Internet, which means it is possible to control these from the same smart phone device (Subsection 4.2).

Whenever an S.O. (LL-SHS) or HL-SHS emits an action it can be presented to the user in the form of a prompt on the smart phone or put into a smart home calendar located in the HL-SHS. The user interface could be based on user profile information so the suggested action can be either performed automatically without user intervention or it can be held back until the user accepts or rejects it. It should also be possible to interact with the calendar system where actions can be either accepted or rejected. Other principles for user interaction with the SHS are described in subsection 3.3.1.

Functionality for doing home control based timed actions should be provided. It can be implemented by schedule timed action in the smart home calendar and let the user interface reflect these to the S.O.s. In addition, the user interface must provide functionality to administrate and setup the different smart home devices.

Other user related items that need to be addressed are: Handling and implementing of user identification; user profiles and user policies; system security and user access rights; and control based smart homes.

9.2.2 SMART OBJECTS AND SENSORS

As illustrated in Figure 9.4 the S.O.s interact with sensors. Based on sensor inputs the S.O.s are able to emit actions that indicate a particular user activity has been performed. Descriptive example is presented in section 9.1.

Compatibility with existing devices, which do not have a smart home interface, can be obtained by using the flexibility of the S.O. implementation. It is possible to connect a non-smart device (e.g., a conventional lamp) without a smart home interface to an intelligent S.O. based wall power outlet. The artificial part together with the sensor input can then be ignored, i.e., the S.O. behaves like a home automation device. Thus, it is possible to remote control the non-smart device and provide timer based control to it.

9.2.3 LL-SHS

The LL-SHS contains the S.O.s (Section 8.5) which captures the particular events from the sensors (Figure 9.4). Based on these the S.O.s learn and predict the user behavior (Section 8.5). Thus, an example could be the behavior of the lamp illustrated in Figure 9.4. Whenever an action is predicted by an S.O. it sends an action command to the HL-SHS, which uses this to predict advanced temporal activities (Section 8.3).

The S.O.s (LL-SHS) are generic in the way that they can be used in different points and positions in the SHS. Hence, it is possible to run the S.O.s as a standalone intelligent platform. In general the S.O. based system can exist in many instances and it is implementable on many different local platforms such as PCs, hbbTVs, IPads, home network routers and NAS devices, smart phones, etc. This is illustrated (Figure 9.4) as a blue cloud named “local processing device”, which contains an LL-SHS system.

The LL-SHS offers a simple robust probabilistic framework that is capable of learning and predicting user activities. Its main goal is to keep simplicity as a core value, but also achieve good reliable results. This simplicity is achieved by using a simplified naïve Bayes probabilistic classifier framework (Section 8.2) as AI element. Actually good results are achieved compared to related frameworks that do not offer these simplicities as explained in section 8.4.

9.2.4 HL-SHS

As illustrated in Figure 9.4 another component in the SHS is the HL-SHS. It can be located at a local server or on a cloud based service as discussed in section 8.3. This system is used to learn and predict more advanced and composite temporal user activities. Thus, most user activities are composed as a sequence of temporal actions which leads to a final action that needs to be detected.

The HL-SHS is a probabilistic flexible system that is able to learn and predict complex user activities in real time, i.e., real-time learning and predicting (Section 8.3). It uses actions detected by other smart home components, such as the S.O.s, as input in combination with today’s time.

The AI element that has been chosen to model the HL-SHS in this work is the well-known Hidden Markov Model (HMM) (Section 5.6). It has been modified in a mathematical way to be adapted to a smart home environment. A more detailed and in-depth exploration of this component can be found in section 8.3.

9.3 CONTRIBUTIONS

This section collects and lists my original contributions derived throughout the research period to the different research areas. In the following section an italic notation is used to highlight these original contributions.

As stated in research question one (Section 1.2) the challenge is to find a system architecture that includes S.O.s and supports a distributed AI framework. The contribution of this work to answer this question is discussed in the following.

Initially, the state-of-the-art is found in the system architecture areas and analyzed for its advantages and disadvantages (Chapter 6). To analyze and be able to concretize these areas into manageable subjects *two generic Centralized Smart Home (CSH) and Agent-based Smart Home (ASH) models have been invented* (Section 6.1). These models are a new unique contribution to the research area of smart home system architectures which is dominated by a diversity of different model types (subsections 6.1.1 and 6.1.2). Additionally, it has been shown (Subsection 6.1.3) that these two models support most of the available smart homes and provide the complexity needed to capture vital details, but at the same time they keep the complexity at a manageable level.

By analyzing the CSH and ASH models (Chapter 6) these problem areas are explored at a detailed level. The combined outcome in the form of gained knowledge has been used to construct a new model (Subsection 6.1.3), which is usable for answering research question one. Thus the model provides the needed elements: Distributed system architecture, distributed AI, and an S.O. concept. This *smart home system (SHS) model* is illustrated in Figure 7.3 and discussed in (Section 7.1). Its distributed behavior is new and removes many of the problems found in the CSH research area such as the bottleneck and single point of failure. In addition, it takes the agent-based ASH system some steps further into a real

distributed system with autonomous, parallel processing, context-aware, and communicating nodes (S.O.s) (Section 7.2).

So the derived CSH, ASH and SHS models contribute to the centralized and distributed (agent-based) SHS research areas with generalizable models that are useable to describe most smart home systems in a uniform manner. These research areas have been introduced in chapter 4.

As discussed, the state-of-the-art smart home concepts can be modeled by mapping them into either the CSH or the ASH model. This means that these research areas do not provide the freedom to combine these models. However, such a model offers advantages (Subsection 6.1.3) and provides the distributed architecture required by research question one. Thus, the *SHS model offers two components*. The first high-level component *reuses existing processing power* that is available in many home devices or processing power that exists on dedicated servers such as cloud based servers. This processing power is used to run advanced AI processing (Sections 6.2 and 8.3). The second low-level component constitutes the *distributed part* of the SHS model (Section 7.1). The possibility to allocate the high-level SHS part on existing devices and distribute the low-level part is new and contributes to the smart home research area (Sections 7.1 and 7.2).

Dividing the model into these two parts is beneficial from a technical perspective with respect to sensor, actuators, power, and used bandwidth (Section 7.1). From a user point of view each S.O. is uniquely tagged with a RFID tag which means that a server based user interface is able to identify it, e.g., when the user touches it with a smart phone or similar devices.

The low-level distributed part in the suggested SHS needs to be able to provide a distributed agent-based S.O. concept in terms of: Network load, resource consumption, and ability to support a distributed AI framework (Section 1.2). A candidate for this is the *suggested S.O. model* (Section 7.2). This model is different from the known ASH based models and constitutes a unique concept. The derived S.O. model is bound to the sensor context and is loosely coupled to the SHS, which means that it behaves autonomously and does not require a complex system setup procedure. They are uniquely identifiable, able to communicate, store events, and process applications in a distributed manner on embedded low-power devices.

Another outcome is that most of the S.O. framework is related to IoT concepts and that the *presented smart home concept is compatible with upcoming IoT technologies* (Section 7.4).

One of the main challenges in deriving an S.O. model is designing its behavior in a structured way and to provide a platform for distributed intelligence. Thus, its software architecture has been defined. These challenges and their different solutions have been modeled by a *UML based context model* (Subsection 7.2.2). Moreover, this work invents a *four level software layered model* where the main contributions are the *middleware layer*, the *application layer* (Subsection 7.2.2) and the *communication architecture* (Subsection 7.2.3). These contributions provide a new general framework that contributes to the ASH as well as the IoT research areas (Sections 7.2 and 7.4).

The application layer runs parallel applications which provide: A timer unit; calendar and activity scheduler unit; sensor input filtering and processing unit; and an output processing unit. These units provide a high-level of generality in the application areas. The developed middleware layer offers general access from external devices by standardizing the interface functionality and the *communication architecture* (Subsection 7.2.3).

A smart home S.O. device is part of the distributed SHS architecture stated in research question one. Further stated is that it must be based on existing technology (Sections 1.2 and 1.3).

The software layered model (Subsection 7.2.2) has been mapped to an *embedded hardware implementation model* (Subsection 7.2.4) by adding interface and communication electronic circuits. Such a model has not been derived for smart homes and this contributes to the ASH, S.O., and IoT research areas (Subsection 7.2.4), (Section 7.4). The final derived S.O. offers a *high-level of generalities* such as *programmable timers, filtered event sampling, programmable calendar functions, low-power communication, and support for AI applications* (Section 7.2). In addition, this model offers: Low battery power consumption; low cost; support for both wired and wireless sensors; and a low form factor which makes it easy to install.

The S.O. software and implementation models suggested in this work add features for *filtering* and *processing sensor events* in an advantageous way so that noise and sporadic sensor events are avoided (Subsection 7.2.4). This saves processing power in the S.O.s because only necessary processing is performed. Moreover, it adds knowledge about general concepts to the agent area because S.O.s are derived from this area and thereby related to them. Thus, this S.O. model adds knowledge to the sensor and actuator research areas, which is described in section 4.4.

To support the integration of S.O.s in an SHS, inter-object communication is needed. This communication must facilitate S.O. low resource consumption, support wired and wireless connections and provide support for multiple devices types and protocols. Such a *communication model has been derived* (Section 7.3). This model is unique and provides many benefits compared to related research areas (Section 7.3). Additionally, it adds new technology and concepts in form of clustering and transmit-only sensors to the smart home communication research area.

The model contains a mains powered core that handles communication with external devices such as a cloud based user interfaces, routes actions predicted by the battery powered S.O.s, and processes both wireless and wired sensor events (Subsection 7.3.3). The model is general in terms of choice of the communication device types and protocols because it supports the use of multiple gateways.

The suggested SHS needs to handle events from the battery powered sensors as discussed earlier (Subsection 4.5.5). These events must be distributed to all the S.O.s without flooding the cluster core. Thus, using multiple gateways raises the problem of multi event flooding, *and a method has been derived which is able to handle this* (Subsection 7.3.3). This method is not new and it has been used in other networks; however, the usage in a distributed smart home context is new and it contributes to this area as well as the IoT research area. Basically, this method is able to suppress the event copies found in common flooding such as the Gossiping algorithm (Subsection 7.3.3).

The communication model defines a concept of a mains powered core (Subsection 7.3.3) that handles power consuming tasks on behalf of the battery powered nodes. Moreover, it handles

wireless and wired sensors and gateways by defining a protocol concept that is able to deal with their challenges. Thus, this work adds to the research area of network technologies described in section 4.5.

To lower the interference level and thereby the S.O. resource consumption a clustering model has been employed. This model must support the limited resources available on some devices. *The clustering model has been derived.* It is new and contributes to the low power smart home research area.

The clustering model optimizes the power consumption and lowers the interference level by grouping the S.O.s into selected spatial clusters (Subsection 7.3.4). These clusters are allocated so that S.O.s in the same room belong to the same cluster and the *cluster header is chosen among these.* This concept lowers the interference and power consumption because the spatial distance is small and no obstacles such as walls are present. Locking the choice of cluster header ensures it is chosen between the mains powered ones in the available pool of S.O.s.

A method to lower the power consumption and component cost is to eliminate the receiver part in the sensors. Thus, *battery-power savings are achieved by allowing transmit-only sensors to participate in the smart home network* (Subsection 7.3.5). This work suggests a method that enables such a concept. In the smart home context this research is a new contribution to the sensors (Section 4.4) and S.O. power saving research area (Subsection 7.3.5).

Normally, using transmit-only sensors are impossible because they cannot participate in the framing used in wireless networks. However, using a CSMA-CD scheme ZigBee devices and retransmitting transmit-only sensors are able to co-exist in the same network.

Another method to enable the use of transmit-only sensors is using SDR. Hence, *a future perspective about using SDR in combination with transmits-only nodes is discussed and a solution is presented* (Subsection 7.3.5). This concept is a new research contribution to the smart home, the WSN and the IoT research areas.

A way to implement this is by using an S.O. to implement an SDR based dual-band transceiver which is able to handle a ZigBee solution overlaid by a transmit-only-sensor signal (Subsection 7.3.5).

As stated in research question two a distributed artificial intelligence framework is needed which complies with the smart home system architecture. Such a system must support distributed AI embedded on small controllers with limited resources and other existing devices. Additionally, it must consider technologies that deal with the battery, bandwidth and processing challenges (Section 1.2).

A distributed AI system has been designed (Chapter 8), which fits into the derived architecture. This system provides a new unique contribution to the smart homes, the IoT, and the hierarchical AI research areas (Chapter 8).

The system divides the processing load into *two subsystems* (Subsection 8.1.2), which require different processing capabilities. *The first subsystem supports devices with low processing capability* (Section 8.5) such as *S.O. agents implemented on embedded devices*, i.e., it fits the LL-SHS. *The second subsystem performs a more complex and resource demanding processing* (Section 8.3) task and is expected to be run on either a smart home server or a cloud based service, i.e., it fits the HL-SHS.

To provide a distributed AI system in a smart home context a mathematical framework is needed, which is based on existing technologies (Section 1.3). Hence, a distributed AI system has been designed by delivering a *general mathematical model* (Section 8.2), which describes a *combined hidden Markov and naïve Bayes model*. By using the combination of models it is possible to *distribute each model element to different processing contexts*. This framework contributes to the distributed smart home, distributed AI, and the IoT research areas (Sections 8.1 and 8.2).

This framework facilitates a distributed SHS by *offering a “standalone” NB agent-based subsystem* (Section 8.5) and a *temporal model, which can be allocated on more resourceful devices* (Section 8.3). Thus, the standalone NB based agent subsystem offers S.O.s, which can be integrated into common home devices, and it supports individual manufacturing.

Employing a complex AI framework on small embedded S.O.s requires a simplification of the used implementation algorithms (Sections 8.2 and 8.3). Thus, the NB agent subsystem is simplified to reduce its resource needs as much as possible without degrading its learned and predicted abilities. These simplifications are suggested and validated in this work, i.e., the *NB agent learning algorithm has been reduced to a simple integer adding operation* (Subsection 8.5.5). Similarly, *the prediction algorithm uses an approach where integers are added and compared to an adaptive threshold* (Subsection 8.5.6). Using these simplifications in smart homes without suffering from a predictability loss is a contribution to the smart home research area (Sections 8.4 and 8.6). Additionally, it contributes to the IoT research area where the employed technologies could be used. Generally, it is expected that some losses are seen when algorithms are simplified, but the used approaches do not reduce predictability considerably compared to similar works (Sections 8.4 and 8.6). However, the benefits achieved from these simplifications have a huge impact on the S.O.s performance and resource use, as described in sections 8.6 and 8.8.

In addition, the sensor interface to the S.O.s has been simplified (Subsection 8.5.3) by using a *fixed sampling time* of one second; a hardware based capture hold circuit; and a *binary (on /off) concept* for the *sensor events, where the off events are ignored*. These techniques save battery power and network resources for the wireless sensors (Section 8.7). Using these simplification techniques in a distributed smart home context is a contribution to the smart home AI research area (Sections 5.7 and 8.8).

As stated in section 1.2 resources such as power consumption, used bandwidth and used processing power are the main challenges. In this light it is relevant to investigate the achieved savings. Some of the savings are provided by using the distributed NB agent-based concept (i.e., the S.O's). It provides considerable savings in the transmitter power consumption and the network load, because *local events are consumed locally inside its cluster by the connected S.O.*(Section 8.7). In addition, this provides benefits such as low network utilization, no single point of failure, low prediction delay (Section 8.7). Employing distributed AI locally in S.O.s with ability to do local processing is a new contribution to the smart home research area and to the IoT research area (Section 8.8).

Additionally, the S.O. agent-based concept has been investigated by using the publicly available dataset from Kasteren et al. (2008a) where it is found that this system scores comparably to similar systems (Section 8.6).

To perform a validation of the obtained savings *the NB agent code has been developed in C-code and simulated* in a state-of-the-art simulator (Section 8.8) from the company Microchip. By optimizing this C-code and implementing the discussed S.O. agent simplifications extensive savings in processing resource, network load and battery power have been found. The optimizations used in the code are a contribution to the smart home research areas that deal with implementation techniques and algorithms. In addition, these techniques are useable in the IoT research area.

As discussed, the AI framework developed in this work is hierarchically structured to reflect the derived system architecture. Hence, to enable the HL-SHS to interface the LL-SHS (i.e., the S.O.s) and similarly increase its efficiency, reliability, and resource savings, some techniques and modifications have been performed as discussed in the next paragraph.

The *hidden Markov model based HL-SHS has been modified* so that it uses binary inputs from the S.O. agents (Section 8.3), i.e., they do not provide any emission probabilities. It also employs a concept of only *allowing one state in the latent variable*, i.e., it is binary (Subsection 8.3.3). To provide these binary predictions an *adaptive threshold has been employed* (Subsection 8.3.4). To add temporal behavior the previous actions must be included in the prediction process. This is performed by using a *simplified Viterbi algorithm* (Subsection 8.3.4) where the final state is known which limits the trellis search. In addition, a *time dimension is added to the emission probabilities* (Section 8.3), which increases the prediction probability. These transformations of a general probabilistic model into a specialized HL-SHS framework have not been seen in a smart home context and are a new contribution to this research area (Sections 5.7 and 8.4).

It is noted that these changes do not reduce the system performance compared to comparable systems (Section 8.4).

An important element in the distributed AI framework is real-time learning (Section 1.2). Thus both systems are designed as *real-time learning circuits* (Subsections 8.3.3 and 8.5.5). This means that no priori training is necessary, i.e., the training happens on the fly when the system runs. In addition, this *adds the ability to update or override old learning with new ones* (Subsections 8.3.3 and 8.5.5). Real-time learning contributes to the smart home and the IoT research areas (Section 5.3).

9.4 INTERPRETATION OF CONTRIBUTION AND RESULTS

This section provides an interpretation of selected contribution parts (Section 9.3) in relation to the state-of-the-art research. Firstly, a walk-through of this work is presented where some of the achieved results are highlighted. Secondly, three selected questions are answered, they are:

1. Model selection for the smart home analysis in relation to the state-of-the-art.
2. Interpretation of the simulated results in relation to the state-of-the-art.
3. The general system architecture in respect to the user-support in smart homes, including state-of-the-art considerations.

9.4.1 OVERVIEW OF THIS WORK AND SELECTED CONTRIBUTIONS

The basic research performed to support this work has been conducted on an exploratory basis to establish the state-of-the-art in smart home models. It has been found that most researchers and performed scientific experiments have used the centralized smart homes architecture. However, research has also been performed in the area of the distributed smart home architecture, where it has been found that this architecture is very promising for future smart homes (Alam et al., 2012), (Reinisch et al., 2010), (Cook & Das, 2006), (Del-Hyo et al., 2012), (Hannon & Brunell, 2005), (Mocanu et al., 2013). Alam et al. (2012) and Silva et al. (2012) point out that this architecture is the future candidate for smart homes. In addition, this architecture has many similarities with the IoT research area, which is beneficial for the future integration of these areas.

This work takes offset in this knowledge and gradually transforms the centralized smart home architecture into a distributed one with focus on its high-level and low-level parts. The high-

level part covers elements such as enabling the use of cloud processing technology (Section 7.1), centralization of composite events (combining sensors information), and providing a central access-point for the system. The low-level part focuses on the composition of S.O. components, the smart home network aspects, and the used S.O. resources. Especially the restricted amount of resources was found to be the limiting factor in providing working S.O.s with today's state-of-the-art technology (Section 8.7). Actually, the most resource demanding part in the S.O.s is the implementation of an embedded AI real-time framework, why different candidates were compared and analyzed (Section 5.5).

From this analysis in combination with the research performed by Cook et al. (2012) it has been found that the probabilistic models are the most beneficial ones for smart home environments (Section 5.7). However, the distributed probabilistic framework for use in smart home context, based on the well-proven NB and HMM probabilistic classifiers, is a new approach. This framework is divided into two parts, a HL-SHS and a LL-SHS (i.e. S.O.s). Both systems have been simulated in a real world context by using qualitative data from independent researchers. The HL-SHS has been simulated by using a designed and implemented model (Section 8.4.1) on datasets available from the WSU CASAS project (Cook, 2012). It has been found that the HL-SHS performs similar to other centralized-based classifiers (Cheng et al., 2009). Similarly, the S.O. has been modeled (Section 8.5) and validated by using datasets from independent researchers. It has been found that its learning and predicting capabilities are comparable to similar works (Section 8.6). Its suitability to be implemented on an embedded microcontroller has been explored by using a simulator, which was provided by the microcontroller manufacturer. It has been found that using the developed simplified and optimized NB framework, considerable savings were provided in processor resources, smart home network interference level, and battery lifetime (Section 8.8).

9.4.2 QUESTION 1: MODEL SELECTION

Model selection for the smart home analysis has been based on different criteria. Firstly, the model needs the ability to support a distributed framework in order to go beyond the current state-of-the-art (Alam et al., 2012), (Reinisch et al., 2010). Secondly, the model must be robust in noisy environments and be able to learn and predict in real-time (Cook, 2012).

Fundamentally, pattern recognition methods and models can be divided into three groups as presented in Table 9.2 and discussed in section 5.2.

Model	Pros and cons	References
Temporal data mining and temporal pattern models	<p>Advantages: these methods find recurrent patterns automatically without user feedback.</p> <p>Disadvantages: the unknown detected patterns can be useless because it is not directly linked to the user behavior. These techniques are closer to action discovery rather than action detection.</p>	<p>(Jakkula & Cook, 2008)</p> <p>(Jakkula et al., 2007)</p>
Logic based methods	<p>Advantages: build behavior rules as a function of time that is hardcoded into the system. Using these rules it is possible to check if some specific behavior takes place in a defined time window.</p> <p>Disadvantages: ambiguity exists in which action that has been performed. I.e., it detects action sequences, not specific actions.</p>	<p>(Cheng et al., 2009)</p>
Temporal probabilistic models.	<p>Advantages: they model sequential data and efficient algorithms exist that perform inference in an optimal manner. In addition they are robust against noise (i.e. uncorrelated sensor events).</p> <p>Disadvantages: annotated training data are needed to learn the model parameters.</p>	<p>(Cook, 2012)</p> <p>(Kasteren et al., 2008b)</p> <p>(Fang et al., 2012)</p> <p>(Kasteren et al., 2011)</p> <p>(Chua et al., 2011)</p>

Table 9.2. The three groups of fundamental pattern recognition methods.

As stated in Table 9.2 the “temporal data mining models” do not relate the user action directly to the sensor events. This means that the user needs to be involved in sorting out usable actions and events, which is not a user-friendly approach. In addition, temporal logics is essentially based on finding sequences of related actions; however, such an approach requires centralized processing where all actions are available, i.e. it is not suitable for distributed processing.

The “logic based methods” are based on a priori knowledge in form of pre-coded rules that are used to detect the most likely action sequence. However, as discussed in relation to the

“temporal data mining models” action sequences are not suitable in a distributed system context. Furthermore, pre-coded rules do not support real-time learning.

The “temporal probabilistic models” have been used by many researchers in a smart home context. Their main disadvantage is that they need training in advance which loads the user. However, in this work the training is performed on the fly (i.e. real-time learning), so the trained S.O. learns the user habits over time.

Conclusively, it has been found that probabilistic models are the most suited ones for distributed systems because they offer the ability to classify single actions, they are robust against uncorrelated sensor events (noise), and they are able to learn a priori sensor patterns when triggered by a single user activity. Regarding the detectability of action sequences, probabilistic models handle this by offering temporal behavior. Statements from other researchers support the choice of probabilistic models in smart homes, e.g. Cook (2012):

Some of the most commonly used approaches are naïve Bayes classifiers, decision trees, Markov models, and conditional random fields [5][6][7][8]. In our approach we initially test three models: a naïve Bayes classifier (NBC), a hidden Markov model (HMM), and a conditional random field (CRF) model. These three approaches are considered because they traditionally are robust in the presence of a moderate amount of noise, are designed to handle sequential data, and generate probability distributions over the class labels.

Their statement is supported by Das et al. (2012), who compared different AI algorithms for real-time use on a resource-constrained mobile phone:

...require a substantial amount of computation, which is difficult to achieve with an Android phone that has a 1 GHz processor. Among the less computationally expensive classifiers, Naïve Bayes performs better and therefore it is chosen to run on the phone. Figure 4 (right) shows the performance accuracy of the five different activities separately with Naïve Bayes classifier. We also tested the performance of Naïve Bayes for real-time activity recognition and the average accuracy is more than 85%.

Similarly, Kasteren et al. (2008a) state:

Our objective is to recognize activities from sensor readings in a house. This is a formidable task, since the label in question (activity performed) cannot be estimated directly from our data (sensor readings). Furthermore, the temporal patterns of our data contain valuable information regarding the performed activity. A suitable framework for this task is temporal probabilistic models.

Looking into the state-of-the-art research in the distributed smart home area, only a limited amount of research has been performed (Subsection 3.3.4). Thus, the “ThinkHome” project (Reinisch et al., 2010), the “Multi-Agent system” (Hamzi et al., 2013), and the ”gold seekers” distributed multi-agent framework for intelligent environments (Hannon & Brunell, 2005) are representative examples. However, these systems use a centralized approach where each agent is allocated to a specific task.

A detailed overview of the discussed multi-agent systems is provided together with a comparative matrix (Table 9.3) which focuses on the important issues discussed in this work and which is supported by state-of-the-art research (Alam et al., 2012), (Silva et al., 2012). Thus, Table 9.3 compares this work in relation to the following three state-of-the-art systems:

1. The “ThinkHome” project is based on a centralized knowledge base which provides the reasoning and data storage facility to the agents. These agents are allocated to handle dedicated tasks such as: user-agent, global-goals-agent, and context inference agents.
2. The “Multi-Agent system” consists of a hierarchical based collection of specialized agents. At the top level the planning and management agents controls the lower level agents such as: service-agents, update-agents, mission-agents, etc.
3. The ”gold seekers” distributed multi-agent framework for intelligent environments project offers a collection of agents where these are allocated to different layers and tasks such as: physical sensor layer, information layer, decision layer, communication layer, etc. Hence, the agents are specialized and allocated in advance.

Challenges in S.O. / agent based systems	System 1	System 2	System 3	This work
Sensitive to single point of failure. (Section 6.1.3.).	Yes	Yes	Yes	No
Requires central smart home server / cloud server (Section 6.1.3.).	Yes	Yes	Yes	(Yes)
Reduced network load and channel blocking (Section 6.1.3.).	No	No	No	Yes
Possible to route data in devices (Section 6.1.3.).	Yes	Yes	No	Yes
Simulated vital elements in embedded form	No	No	No	(Yes)
Offer AI capable of running in multiple instances on small low power processors (Section 8.5)	No	No	No	Yes
Offer support for high level temporal actions (Section 8.3)	Yes	Yes	No	Yes
Offer the possibility to save power by hibernating agents	No	Yes	Yes	Yes
Bridge the gap between controlled homes and smart homes.	No	No	No	Yes
Supports integration with IoT devices.	No	No	(No)	Yes
Offer real-time parallel processing	Yes	Yes	Yes	Yes
Optimized S.O. clustering strategy and model prolonging battery lifetime (Subsection 7.3.4)	No	Yes	No	Yes
Optimized embedded AI in S.O.s for saving battery lifetime (Section 8.7)	No	No	No	Yes
S.O. (agent) offers standardized interfaces (middleware layer, etc.) (Subsection 7.2.2).	No	No	Yes	Yes
Wired / non wired communication architecture (Subsection 7.2.3)	No	Yes	No	Yes
General hardware model with timers, pulse-width-modulators, power outputs, etc. (Subsection 7.2.4)	No	No	Yes	Yes
Support for low power transmit-only-sensors (Subsection 7.3.5)	No	No	No	Yes
Offer “standalone” functionality (enable common home devices and support individual manufacturing) (Section 8.3), (Section 8.5)	No	No	No	Yes
S.O./agent offer real-time learning and prediction (Subsections 8.3.3 and 8.5.5).	No	No	No	Yes

Table 9.3. Comparative matrix for multi-agent systems.

In conclusion, this work is research-wise positioned so it moves the research frontiers in the distributed smart home area beyond the current state-of-the-art with respect to the discussed challenges.

9.4.3 QUESTION 2: RESULTS OF THE SIMULATIONS

Simulations are important tools for validating and extracting knowledge from invented models and architectures and identifying challenges that need more research to reach their “satisfactory states”. For this work two simulators have modeled, designed and developed as discussed in section 8.4 in order to simulate and validate the performance of the derived HL-SHS and the distributed S.O.s.

Usually, smart homes that are based on HMM have their AI implementation directly connected to the smart home sensors. However, in this work the HMM is simplified and only connected to the S.O.s, i.e. it receives predicted actions instead of sensor events. The S.O.s implement a simplified AI system based on the well-known NB classifier, which in turn is connected to the sensors. In addition, the algorithms used for the smart home AI system have been optimized with respect to their processing and network resource consumptions. Most of the algorithms content and structural changes have been performed by using non-linear mathematics such as quantization, truncation, integer and Boolean types. Performing such dramatic changes has influence on the algorithm performance, which have been considered and compensated throughout the design of these, however, from a scientific point of view this performance needs to be validated and compared to the non-modified use of these.

The validation starts by looking into the learning achieved by the HL-SHS algorithm by simulating it on a public available dataset and analyzes its learning ability, which is captured and stored in its “weights” (Section 8.4). This analysis reveals that the learning has taken place (Subsection 8.4.2). In addition, the used dataset contains user annotations, why it was possible to compare the predicted output (an activity) from the HL-SHS algorithm with the user provided ground truth. Running this simulation twice on different datasets showed that the HL-SHS algorithm performed similar to the work performed by others (Subsection 8.4.2), (Subsection 8.4.3).

A representative selection of the state-of-the-art AI systems that learn and predict user activities in smart homes is presented in Table 9.4 as the percentage of true positive, where the user annotation and the predicted value agrees.

Algorithm	Score	Ref.
Modified HMM (this work, example 1)	86%	Section 8.4.2
Modified HMM (this work, example 2)	94%	Section 8.4.3
ASBR	80%	(Cheng et al., 2009)
CBR	75%	(Cheng et al., 2009)
HMM	73%	(Fang et al., 2012)
HMM	83%	(Kasteren et al., 2011)
HMM (Cairo)	82%	(Cook, 2012)

Table 9.4. A representative selection of the state-of-the-art AI systems that learn and predict user activities in smart homes, compared to this work.

As noted from Table 9.4 the HL-SHS algorithm developed in this work performs similarly to related works, even though it has been optimized as previously discussed. Especially, the HMM (Cairo) experiment performed by Cook et al. (2012) is noteworthy because it uses the same dataset as this work, but with a fully equipped HMM based classifier that was connected directly to the sensors.

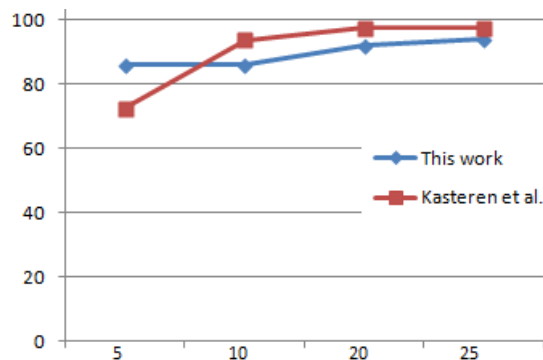


Figure 9.5. HL-SHS real-time learning performance compared to the HMM model used by (Kasteren et al., 2008a). X-axis is learning times and y-axis is true-positive outputs in percentage.

Comparing the real-time learning for the HL-SHS algorithm with the HMM based system presented by Kasteren et al. (2008a) reveals that this system offers similar accuracy and its learning rate is approximately the same (Figure 9.5). As illustrated, the x-axis is the number of training times the algorithms have been trained and the y-axis is the predicting score in true-positive percent.

The S.O.s have been optimized with a focus on being implemented on a resource restricted embedded microcontroller. Thus, non-linear approximations and quantizations have been used in form of binary sensor events, using integer mathematics, to avoid expensive multiplications, simplify the probability functions, use binary decisions, adding a dynamic threshold, and approximating the logarithm functions, etc. (Section 8.5). This optimized S.O. algorithm must be validated with respect to its performance. Firstly, it needs to be validated in terms of its learning and predicting abilities compared to similar systems. However, a direct comparison with similar systems is not possible, because to the author's knowledge they do not exist. So, as an alternative it is compared to the existing centralized systems, where the AI algorithm is allocated on a server. Secondly, the S.O.s are implemented on resource restricted microcontroller devices, which often are placed at hard-to-reach positions in the smart homes. Thus, its resource consumption and performance in terms of battery lifetime need to be found by using simulation.

Validating the S.O. in terms of its learning and predicting abilities has been performed by using a designed and implemented simulator (Subsection 8.4) and a publicly available dataset and the performance has been compared to the results presented by Kasteren et al. (2008a) (Figure 9.6).



Figure 9.6. The SO-NB algorithm developed in this work compared to the results achieved by Kasteren et al. (2008a) on a centralized server.

As illustrated (Figure 9.6), the modified and implemented S.O. algorithm (modified NB) performs similar to the centralized server implementation of the traditional NB algorithm with

full resolution, no quantization, and using floating-point calculations, i.e. all the needed resources were provided.

Research of the S.O. resource consumption and performance in terms of battery lifetime has been performed by using a simulator from Microchip and the public available dataset from the Milan CASAS experiment (Cook, 2012), (Section 8.7). The simulated code was implemented using the discussed optimizations and has been compared with a non-optimized implementation in Figure 9.7 (Section 8.2), (Section 8.5), (Section 8.8).

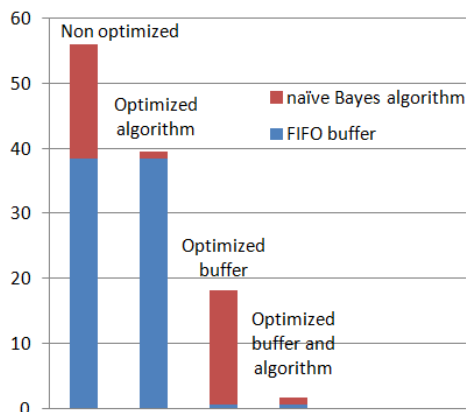


Figure 9.7. Saving in process-time (ms) achieved by using optimized FIFO buffer and naïve Bayes implementations.

As illustrated considerable savings have been achieved (Figure 9.7) for processing one event in an optimized S.O. compared to a non-optimized S.O.

From Milan CASAS dataset (Cook, 2012) 5 kitchen sensors have been selected and their energy consumption of transmitting events in a traditionally centralized system has been compared to the energy consumption used in the same system based on S.O.s (Section 8.8). Because the S.O.s only transmit predicted actions and not all the sensor events, large savings in consumed energy and smart home network load have been found (Figure 9.8) (Section 8.8).

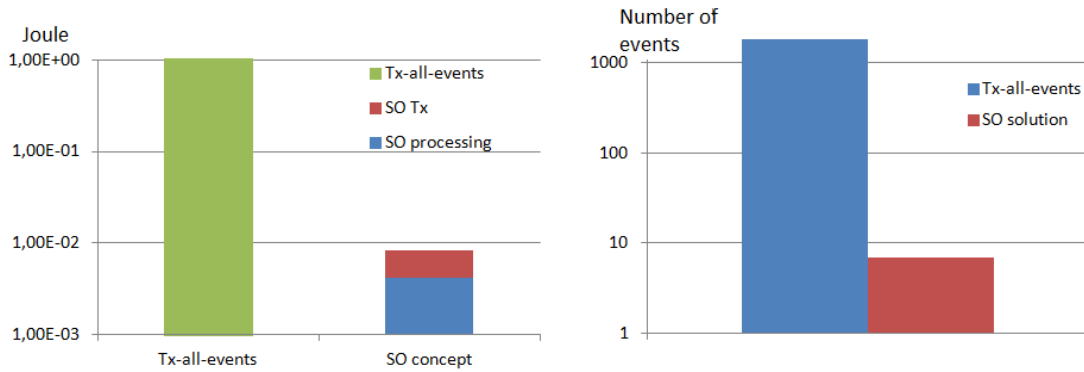


Figure 9.8. Left-hand side: A traditional sensor framework (Transmit-all-events) is compared with the presented S.O. concept. Five kitchen sensors in a CASAS smart home have been used. Right-hand side: Saving in transmitted events.

Thus, a common centralized state-of-the-art system like “a smart home in a box” (Cook et al., 2012) or the smart home experiments performed by (Kasteren et al., 2008a) transmit all sensor events to a centralized server, which performs the AI processing; such a system is named “tx-all-events” in Figure 9.8. In contrast, the S.O.-based concept “consumes” the events locally and only transmits the predicted actions (Lynggaard, 2013c). This means that the energy is used to process the event locally and to transmit the predicted actions. However, as illustrated in Figure 9.8 this amount of energy is much lower. In addition, the amount of predicted actions is much lower than the amount of sensor events, why the network load, the interference level, and the power consumption are reduced.

In conclusion, the S.O. concept presented in this work moves the state-of-the-art research frontiers beyond the existing agent and centralized based smart home systems, by contributing with new algorithms that cover both temporal and non-temporal AI algorithms. These algorithms are integrated into the S.O.s, i.e. they are implemented into small embedded microcontrollers. By using simulation based on third party datasets the performance of these S.O. algorithms compared to agent and centralized systems has been found to be equally good with respect to prediction performance, but superior with respect to energy consumption, resource usage, and network load.

9.4.4 QUESTION 3: GENERAL SYSTEM ARCHITECTURE

This subsection looks into how the presented HL-SHS and the S.O. architectures support the smart home users by offering new intelligent services. These services are beyond what can be offered by the state-of-the-art smart homes because they are either centralized or agent based, which means that they miss the advantages in combining these systems shown in the present work. Furthermore, this work provides a “true” distributed S.O. based system based on IoT in contrast to the state-of-the-art research of agent-based systems and a centralized approach (Subsection 4.3.1). The following paragraph discusses these services.

A smart home must comprise services for initial setup, management, and user interaction with the smart home devices. The initial setup procedure used in the centralized smart homes includes a smart home management system, which often is based on a GUI (Silva et al., 2012), (Rathnayaka et al., 2012) (Bregman & Korman, 2009). To be able to add new devices or communicate with existing devices they need to be identified by the system, which often means that the user must traverse through a menu hierarchy and possess the needed identity information. However, the SHS architecture presented in this work offer an alternative way. Thus, the user can identify a device (S.O.) by physically approximating it with a portable device (e.g. a smart phone) that reads the device RFID information. An app on the portable device is then able to perform most of the setup procedure and open the dedicated menu for the S.O. device, which is located on the centralized system part, i.e. the HL-SHS. From a user experience point of view this architecture relates the identity of a physical S.O. device to its virtual identity that exists in the centralized management system. Hence, this activity is similar to the user experience of remotely controlling a common TV, etc.

Another advantage in the presented SHS architecture is the standalone ability of the S.O. devices. Decoupling the S.O. devices from the smart homes means that they are not part of a predefined system where the components depend on each other, such as the “smart home in a box”(Cook et al., 2012). Using decoupling means that different manufacturers are able to manufacture S.O. devices as long as the middleware interfaces are coherent (this requires some standardization, which is currently missing). Thus, the free market mechanisms are supported, which enables an evolution in the smart home area. From a user perspective this means that the S.O. devices like e.g. a table lamp can be manually operated without using its

smart features, or it can easily be integrated into the smart home AI-based framework, which enables its full functionality. Additionally, by using the Internet, the user is able to remotely control S.O. devices and perform programming of these remotely. This means that the user can turn S.O. device functionality on/off or program simple built-in timer functionality remotely. Thus, users are in control and are able to manage the S.O. device the way they prefer. The discussed benefits moves the research frontiers in smart homes beyond the state-of-the-art and probably will be part of the future smart homes (Alam et al., 2012), (Silva et al., 2012).

An architecture that includes an S.O. part offers advantages as previously discussed. However, combining this with a HL-SHS part provides advantages in form of synergetic services. One of these services is the smart home calendar introduced by Yu et al. (2010), which keeps track of the users and makes updates based on previous decisions taken by the users. By extending their framework the smart home calendar can capture user activities that have been detected by the S.O.s. This approach offers HL-SHS-agent based services (Section 9.1), offers an overview of the performed user actions, and it offers the possibility to manage these, e.g. in relation to defined user profiles. In a future perspective this system can be expanded to include coordination with the individual smart home user calendars, which enables social or health related services, etc. In addition, smart home calendars can be used in a larger perspective such as smart cities, intelligent power grids, and other big-data related services.

The implicit context-based filtering constituted by the S.O.-devices grouping (Section 7.3) makes the HL-SHS user management more transparent. Thus, on a common centralized system an agent can either use all raw events, or some pre-filtering can be used to sort out the relevant ones. In the first case the sensor signal-to-noise ratio (i.e. they “misfires”) will affect the prediction and demand huge processing resources. In the second case pre-filtering requires complex structures and settings that interact with the huge amount of sensor data.

The similarity between IoT and the S.O.s presented in this work enables new services in the areas of ambient assisted living (AAL) and telemedicine. Dohr et al. (2010) state that IoT will be an enabler for healthcare services and telemedicine where elderly stay in their homes and still are accessible for different groups of care providers. Their vision is shared by the EU-

commission (2009). However, their works do not deal with the large amount of processing resources that are needed in the IoT devices for handling AAL scenarios, etc. However, in relation to the framework of HL-SHS and S.O.s this problem is manageable, because the S.O.s only perform part of the AI processing and hand over their results to the HL-SHS, which finishes the resource demanding part of the processing.

In conclusion, the framework developed in this work moves the IoT research frontiers by offering the benefits of “standalone” IoT devices alias S.O.s, but at the same time provides a high level framework (HL-SHS), which offers the beneficial elements found in the centralized smart home system research area. These contributions have been positioned research-wise by the provided simulations and examples.

10 CONCLUSION AND OUTLOOK

This chapter starts with a discussion of the answer to the research question (Section 10.1). It is followed by a reflection of the used process (Section 10.2) and finally, the future challenges are explored in section 10.3.

10.1 ANSWERING THE RESEARCH QUESTION

This thesis has studied a smart home distributed system architecture and a distributed AI framework which can be integrated into it. Hence, the significance of this research is its contributions that move the state-of-the-art research forward in the relevant areas which constitute today's (2013) smart homes. The derived distributed system architecture contributes with spatial flexibility and in particular with an S.O. framework which moves the smart home devices into the coming era of Internet of things. The derived distributed AI framework contributes with a hierarchical concept that offers the possibility to provide AI with different complexity levels at different spatial locations. In addition, the system offers progressive learning on the fly, i.e., the system improves gradually without priori training. Similarly, services are offered on the fly.

To revisit the aim of this thesis the research questions are consulted. These are restated for convenience:

Question 1: *How should the distributed system architecture for smart homes be designed in order to incorporate AI and the diversity of S.O's?*

The research objectives are to explore and derive a distributed system architecture, including S.O.s, use existing technology such as small embedded controllers, and provide support for the derived distributed AI framework. These areas have been researched and a model has been derived. Vital parts of the S.O.s, included in this model, have been implemented and simulated on a modern microprocessor platform.

Question 2: *How should the AI be distributed to comply with the smart home system architecture?*

The objectives of this research question are to explore the possibility to distribute smart home AI systems, add real-time learning, support distributed AI embedded on small controllers with limited resources, and consider technologies that deal with the battery, bandwidth and processing challenges. These issues have been discussed and researched in this thesis. In addition, models have been derived for simulation of critical parts. Simulation outputs from these models have shown that the derived systems perform comparable to other systems.

Additionally, this research provides significance in the form of its contribution to the smart home area where it provides context aware services based on AI and facilitates remote controls. However, some barriers exist in the form of high cost, inflexibility, and poor manageability. By using the derived thesis elements, i.e., the generality of the S.O.s in combination with the high-level AI framework these barriers are lowered. This is mainly because it facilitates a graduate process where the smart homes can be created on an evolutionary basis. This means that the manufacturers can provide AI based smart devices (S.O.s), which are based on their own technologies as long as the middleware is compatible. In addition, the simplifications provided by the AI framework, the optimized battery performance, and the network structures enable smart devices to be implemented on small cheap embedded platforms. Using these platforms facilitate lower cost and an increased flexibility. Regarding the poor manageability AI will automate part of the smart home functionality; however, this does not necessarily increase the manageability.

From an economic, political and strategic perspective this research provides a significant difference. Hence, the European Commission highlights the challenge in developing semantics and software, which can emulate human reasoning and identify smart devices as a potential growth and research area. Looking into the controlling and monitoring feature market in North America, a research survey indicates that 50 percent or more of the customers would like to control, set scenes and remotely monitor their homes. The forecasted value of this market today (2013) is 2.4 billion dollars.

Thus conclusively, state-of-the-art solutions have been researched and derived. These moves the research frontiers forward and contribute to deploy the smart home concepts into real world products.

10.2 REFLECTIONS

Basically the research period has progressed as planned, however, minor changes have been made to the plan on the fly. This has not been a problem because the used iteratively based process model was able to deal with these changes and adapt to the way to go. Thus, using an iterative experimenting approach with literature and online search, mathematical derivation, implementation, and verification by simulation has worked well, even though it did not always converge into a solution in the first few rounds.

Students, media, friends and guests have often asked when smart homes become commonly available. Five years ago, the answer would probably have been five years from now; however, today we know this is not the case. Thus, researching the smart home area has revealed that it is slow moving and results come on a wide evolutionary basis. Nevertheless, it has been found that smart home research has made considerable progress during the author's research period, but it has not been commercialized yet.

From a summary point of view the performed smart home research has followed the planned activities, achieved the expected outcomes, and it has provided interesting learning.

10.3 DISCUSSION AND FUTURE CHALLENGES

This thesis has used an explorative approach to investigate distributed smart home models and their integrated AI framework. In these focus areas the subjects are explored, researched and modeled at a detailed level. However, solving some of the smart home challenges still leaves multiple challenges explored at a holistic level only. This section is dedicated to discuss some of these.

10.3.1 SMART HOMES AND USERS

To obtain user acceptance for adopting smart home technologies solutions with good usability are needed. Actually, this is one of the main barriers (Brush et al., 2011). Thus, a question that needs to be explored is how to design a distributed management system, which is able to handle complex context information and user preferences in a user friendly way. To answer this question a multitude of factors needs attention such as: Who are the users? Which smart home elements should be managed? How should this management be performed? In addition,

research is needed to explore the usability issues where the users are involved. Today (2013) experiments are mostly performed in living labs, but using an evolutionary approach instead, where the S.O.s gradually enter the homes, will most likely increase the user acceptance ratio as seen from the evolution of the Internet. This work supports an evolutionary approach by offering S.O.s, which can be integrated into common things and which offer generic support for existing smart home network technologies.

Another challenge is adding multi-user support to the smart homes. This requires that the S.O.s have an identity, which is addressable from the users interface. Thus, using RFID is one possibility, another is using a picture recognition based scheme. So, by using a smart phone camera detection and identification of S.O.s are possible (Lynggaard, 2013d). Alternatively, RFID tags could trigger an application on the user's smart phone which downloads and runs a dedicated management interface for the triggering device. Other alternatives are by using an intuitive GUI-based 3D virtual reality world (IBM Sales & Distribution, 2010) or finger gesture in front of the camera (Silva et al., 2012), etc.

Issues as personalization, privacy and security also need future exploration. Personalization and privacy in smart homes could be based on user profiles which are widely used for personalization of services, device settings, etc., in our daily life. However, only limited information is available regarding systems that offer portable user profile management with support for human reasoning and AI. A portable system has the possibility to carry the user's identity and profile required in different contexts. Designing such a system in the context of smart homes is challenging, because only few papers deal with this. Thus, Li et al. (2010) deal with a customizing knowledge-based recommender system based on user behavior analysis. Silva et al. (2009) explore conflict solving between context-aware applications. Jacob et al. (2009) model dynamic service behavior using context functions. None of these papers address a portable user profile management system with human reasoning. The security perspective also needs to be addressed because S.O.s can be remote controlled and personal information can be revealed from these. Some low-level security is present in this work because it uses today's technologies with built-in security systems (Subsection 4.5.6). However, a security framework for the high-level functionalities needs to be researched.

Brush et al. (2011) suggest using the unique nature of the smart home domain to design simple primitives, e.g., proximity indicates a level of trust.

10.3.2 PUSHING THE SMART HOMES FORWARD

As discussed, Brush et al. (2011) state four barriers that prevent smart homes from moving forward, they are *high cost of ownership*, *inflexibility*, *poor manageability* and *difficulty achieving security*. These have been discussed in subsection 10.3.1 and 3.3.2. So another question is if the market is available, but as discussed earlier (Subsection 3.3.2) the area of energy management, home monitoring, home control and home healthcare technology have a turnover of billions of dollars in the U.S. market only. Thus, the market is available, but to bring down the cost research is needed in the areas of multi manufacturing, i.e., the product must be attractive for the consumers, produced by different manufacturers, plug-and-play enabled, and the product must offer a level of autonomous behavior based on AI. However, the system presented in this work addresses most of these needs.

The coming era of IoT will promote and broaden the market for smart home technologies. Hence, the presented S.O. concept and technologies can be used in other areas such as telemedicine, helping disabled persons, ambient assisted livings, reducing pollution, intelligent traffic systems, and facilitate green technologies. Further research is needed to achieve this.

10.3.3 FUTURE SMART OBJECT CHALLENGES

The IoT area shares many common elements a characteristics with the S.O. area as stated in section 3.4. However, there are differences because the S.O.s offer a general framework for supporting sensors and actuators with the needed resources such as AI, communication, local processing, operating system, middleware interface, etc. A future challenge is to what extent it is advantageous to integrate the S.O.s approach with the IoT area.

It is reasonable to assume that the technology will progress according to Moore's law, which means that the amount of processing power will increase and the power consumption will drop. Thus, low-power embedded devices together with their peripheral devices will iterate against the concept of SoC. This kind of integration provides possibilities for using other technologies such as energy harvesting, integrated sensors with processing and transceiver

facilities. So, the future S.O.s have a possibility to be integrated with all its external devices and sensors which will reduce the price, the power consumption and the form factor. This area needs future research.

Another area that needs attention is developing a standardized SHN protocol which supports simple low power data transfer in combination with a more power consuming high rate data transfer. Such a network will support the S.O.s daily *action* exchange and it will support a high-level exchange of setup information. This network needs to support very low-power sensor devices that use the transmit-only approach. A beneficial approach would be to use an ip-based topology because it will simplify the gateway devices. In addition this network must support wired and wireless devices, which are battery and mains powered.

To handle the routing of different vendors system and protocol choices the S.O.s gateways and routers could benefit from the SDR and Cognitive Radio research areas. The aspects of SDR's have been discussed in subsection 4.5.1. Cognitive Radio provides the possibility to use shared and unused frequency bands in one device. So, using these technologies in the S.O. area could be beneficial; however, research is needed in this area.

10.3.4 FUTURE ARTIFICIAL INTELLIGENCE CHALLENGES

With the future presence of embedded devices and SoC that offer more processing power for less power consumption it will be possible to allocate more advanced AI algorithms directly on the devices. So, it will be possible to use distributing processing where all the devices work together to form a powerful single virtual processing device which will be able to handle the HL-SHS and the user interface. This needs to be researched. However, the presented work has shown that using more advanced intelligence directly on the S.O.s will not improve the recognition probabilities (Subsection 5.6.2).

Another challenge is to research the non-probabilistic algorithms in a distributed smart home context. Especially, the real-time learning feature provided by this work constitutes an interesting research area.

Another AI area that could benefit from future research is the noise level embedded in sensor data. Thus, as stated (Subsection 8.4.3) the prediction probability normally is in the interval of

80 to 90 percent. Future research in this area could provide the driving mechanisms behind this limit and maybe suggest ways to improve it.

11 BIBLIOGRAPHY

AAL_Association, 2012. *Catalogue of Projects 2012*. Brussels: Ambient Assisted Living Association aal-europe.eu.

Abdelgawad, A., Abdelhak, S., Ghosh, S. & Bayoumi, M., 2009. A low-power multiplication algorithm for signal processing in wireless sensor networks. In *IEEE International Midwest Symposium on Circuits and Systems, 2009. MWSCAS '09. 52nd.*, 2009.

Alam, S. & Noll, J., 2010. Enabling Sensor as Virtual Services through Lightweight Sensor Description. In *Fourth International Conference on Sensor Technologies and Applications.*, 2010. IEEE Computer Society.

Alam, M., Reaz, M. & Ali, M., 2012. A Review of Smart Homes—Past, Present, and Future. *IEEE Transactions on Systems, Man, and Cybernetics 42(6)*, pp.1190-203.

Alkazami, B., Felemban, E., Abid, A. & Zahrani, F., 2012. Middleware Model for Wireless Sensor Networks. In *International Conference on Multimedia Computing and Systems (ICMCS).*, 2012. IEEE Conference Publications.

Alliance-IPSO, 2013. *Alliance, IPSO*. [Online] Available at: <http://www.ipso-alliance.org> [Accessed 2013].

Ambler, S.W., 2006. *Why Agile Software Development Techniques Work: Improved Feedback*. [Online] Available at: <http://www.ambysoft.com/essays/whyAgileWorksFeedback.html> [Accessed 2013].

Arabnia, H.R., Fang, W.-C., Lee, C. & Zhang, Y., 2010. Context-Aware Middleware and Intelligent Agents for Smart Environments. *Intelligent Systems, IEEE, 25(2)*.

Araki, K. & Morishita, Y., 2012. Past, current, and future trend in reconfigurable transceiver research and development. In *IEEE International Symposium on Radio-Frequency Integration Technology (RFIT).*, 2012. IEEE Conference Publications.

Artemio, S. & Leonardo, B., 2012. Evaluation of CORBA and Web Services in distributed applications. *22nd International Conference on Electrical Communications and Computers (CONIELECOMP)*, pp.97-100.

Azim, A. & Islam, M., 2009. Hybrid LEACH: a relay node based low energy adaptive clustering hierarchy for wireless sensor networks. In *Conference on Communications (MICC), 2009 IEEE 9th Malaysia International.*, 2009. IEEE Conference Publications.

Bandyopadhyay, D. & Sen, J., 2011. Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Personal Communications 58(1)*, pp.49-69.

- Basu, D., Moretti, G., G., S. & Marsland, S., 2013. Wireless sensor network based smart home: Sensor selection, deployment and monitoring. In *IEEE Sensors Applications Symposium (SAS)*., 2013.
- Bhardwaj, S., Ozcelebi, T., Lukkien, J. & Uysal, C., 2012. Resource and service management architecture of a low capacity network for smart spaces. *IEEE Transactions on Consumer Electronics* 58(2), pp.389-96.
- Bishop, C., 2006. *Pattern recognition and machine learning*. Springer business and science.
- Blaszczynszyn, B. & Radunovic, B., 2008. Using transmit-only sensors to reduce deployment cost of wireless sensor networks. In *IEEE INFOCOM. The 27th Conference on Computer Communications*., 2008. IEEE Conference Publications.
- Bleda, A. et al., 2012. Evaluation of the impact of furniture on communications performance for ubiquitous deployment of Wireless Sensor Networks in smart homes. *Sensors* 12(5), pp.6463-96.
- Boegel, G., 2012. Batteryless sensors in building automation by use of wireless Energy Harvesting. In *IEEE 1st International Symposium on Wireless Systems (IDAACS-SWS)*., 2012.
- Bregman, D. & Korman, A., 2009. A Universal Implementation Model, for the Smart Home. *International Journal of Smart Home*. 3(3) , pp.15-30.
- Brgulja, N., Kusber, R. & David, K., 2010. Validating Context Information in Context Aware Systems. In *Sixth International Conference on Intelligent Environments*., 2010. IEEE Conference Publications.
- Brodersen, L., 2008. *Geo-communication and information design*. 1st ed. Tankegang a/s.
- Brush, A. et al., 2011. Home automation in the wild: challenges and opportunities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*., 2011. ACM.
- Burns, A. & Wellings, A., 2001. *Teal-Time systems and programming languages*. 1st ed. Pearson education limited.
- Casilari, E., Cano-Garcia, J.M. & Campos-Garrido, C., 2010. Modeling of Current Consumption in 802.15.4/ZigBee Sensor. *Sensors* 10(6), June. pp.5443-68.
- Castellani, A.P. et al., 2010. Architecture and Protocols for the Internet of Things: A Case Study. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 8th., 2010.
- CERP-IoT et al., 2010. *Vision and Challenges for Realising the Internet of Things*. EC - Information Society and Media DG /CERP-IoT.
- Chan, M., Estève, D., Escriba, C. & Campo, E., 2008. A review of smart homes- present state and future challenges. *Computer methods and programs in biomedicine* 91(1), pp.55-81.

- Chen, M., 2013. Towards smart city: M2M communications with software agent intelligence. *Multimedia Tools and Applications* 67(1).
- Chen, C., Das, B. & Cook, D.J., 2010. A Data Mining Framework for Activity Recognition In Smart Environments. In *Intelligent Environments (IE), Sixth International Conference.*, 2010.
- Cheng, S., Wang, C. & Chen, C., 2009. An Adaptive Scenario Based Reasoning System cross smart houses. In *International Symposium on Communications and Information Technology, ISCIT, 9th.*, 2009. IEEE Conference Publications.
- Cheng, S., Wnag, C. & Hornng, G., 2008. OSGi-based smart home architecture for heterogeneous network. In *International Conference on Sensing Technology, ICST, 3rd.*, 2008. IEEE Conference Publications.
- Chen, Z. & Liu, G., 2010. Application of Artificial Intelligence Technology in Water Resouces Planning of River Basin. *International Conference of Information Science and Management Engineering 1(1)*, pp.322-25.
- Chen, J., Wan, J. & Li, F., 2011. Machine-to-Machine Communications: Architectures, Standards and Applications. *KSII Transactions on Internet and Information Systems* 6(1), pp.134-50.
- Chen, Z. et al., 2013. A localization method for the Internet of Things. *The Journal of Supercomputing* 63(3), pp.657-74.
- Choi, J. & Shin, D., 2005. Research on design and implementation of the artificial intelligence agent for smart home based on support vector machine. *Advances in Natural Computation Lecture Notes in Computer Science* 3610(1), pp.1185-88.
- Chong, G., Zhihao, L. & Yifeng, Y., 2011. The research and implement of smart home system based on Internet of Things. In *International Conference on Electronics, Communications and Control (ICECC).*, 2011. IEEE Conference Publications.
- Cho, H., Yamazaki, T. & Hahn, M., 2010. AERO: extraction of user's activities from electric power consumption data. *IEEE Transactions on Consumer Electronics* 56(3).
- Christensen, S.H., Jamison, A. & Botin, L., 2011. *A hybrid imagination, Science and Technology in Cultural Perspective*. 1st ed. Morgan & Claypool Publisher.
- Chua, S., Marsland, S. & Guesgen, H., 2011. Behaviour recognition in smart homes. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence.*, 2011. IJCAL.
- Cockburn, A., 2008. *Using both incremental and iterative development*. Software Engineering Technology.
- Cook, D.J., 2012. Learning Setting-Generalized Activity Models for Smart Spaces. *IEEE Intellligent Systems* 27(1), pp.32-38.

- Cook, D., Crandall, A., Thomas, B. & Krishnan, N., 2012. CASAS: A Smart Home in a Box. *IEEE Computer pp(99)*, p.1.
- Cook, D. & Das, S., 2006. A Multi-agent Approach to Controlling a Smart Environment. *Designing Smart Homes Lecture Notes in Computer Science 4008(1)*, pp.165-82.
- Cook, D. et al., 2003. MavHome: An agent-based smart home. In *IEEE International Conference on Pervasive Computing and Communications, (PerCom)*, 2003. IEEE Conference Publications.
- Coulouris, G., Dollimore, J., Kindberg, T. & Blair, G., 2011. *Distributed systems concepts and design*. 2nd ed. Addison-Wesley.
- Darianian, M. & Michael, M.P., 2008. Smart Home Mobile RFID-based Internet-Of-Things Systems and Services. In *International Conference on Advanced Computer Theory and Engineering, 2008. ICACTE '08.*, 2008.
- Das, B. et al., 2012. Using Smart Phones for Context-Aware Prompting in Smart Environments. In *IEEE Consumer Communications and Networking Conference.*, 2012. IEEE Conference Publications.
- Del-Hyo, R. et al., 2012. Virtual Agents in Next Generation Interactive Homes. *Trends in Practical Applications of Agents and Multiagent Systems, Advances in Intelligent and Soft Computing 157(1)*, pp.9-17.
- Dohr, A. et al., 2010. The Internet of Things for Ambient Assisted Living. In *Proc. of Information Technology: New Generations (ITNG'10)*, 2010. Information Technology: New Generations (ITNG).
- Dominici, M., Zecca, G., Weis, F. & Banatre, M., 2010. Physical Approach in Smart Homes A Proposition and a Prototype. In *Proceedings of the Third conference on Smart Spaces and next generation wired, and 10th international conference on Wireless networking.*, 2010. Springer-Verlag Berlin Heidelberg.
- Dongmei, Y. & Zhiguang, D., 2010. zigbee-based smart home system design. In *International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010. IEEE Conference Publications.
- Donninger, C. & Lorenz, U., 2010. *The hydra project*. Xcell Journal.
- Dori, D., 2002. *Object-process methodology: a holistic systems paradigm*. 1st ed. Haifa, Israel: Springer-Verlag.
- Drew, J., 2012. Energy Harvester Produces Power from Local Environment, Eliminating Batteries in Wireless Sensors. *ElectronicDesign*, 9 2012. pp.1-2.
- Dunkels, A., 2008. *IP for Smart Objects*. Swedish Institute of Computer Science.

EC, C.o.t., 2009. *Internet of Things — An action plan for Europe*. Brussels: Commission of the European Communities EU.

Eisenhauer, M., Resengren, P. & Antolin, P., 2009. A development platform for integrating wireless devices and sensors into Ambient Intelligence systems. *IEEE conference on SECON*, pp.1-3.

Ericsson, 2011. *More than 50 billion connected devices – taking connected devices to mass market and profitability*. Ericsson.

EU Commission, 2009. *Internet of Things - An action plan for Europe*. [Online] Available at: http://ec.europa.eu/information_society/policy/rfid/library/index_en.htm [Accessed 2013].

EU_FP7, C., 2013. *Cordis*. [Online] Available at: http://cordis.europa.eu/fp7/ict/programme/home_en.html [Accessed 2013].

European Commission, 2013. *clima policies roadmap*. [Online] Available at: http://ec.europa.eu/clima/policies/roadmap/index_en.htm [Accessed 2013].

Fahim, M., Hameed, M. & Lee, S., 2010. A multi-strategy bayesian model for sensor fusion in smart environments. In *5th International Conference on Digital Object Identifiers.*, 2010. IEEE Conference Publications.

Fang, H. & He, L., 2012. BP Neural Network for Human Activity Recognition in Smart Home. *International Conference on Computer Science and Service System*, pp.1034-37.

Fang, H., Srinivasan, R. & Cook, D.J., 2012. Feature Selections for Human Activity Recognition in Smart Home Environments. *International Journal of Innovative Computing, Information and Control* 8(5), pp.3525-35.

Fedosseev, A. et al., 2011. *Broadband forum value proposition for connected homes, issue 1*. Broadband forum.

Gershenfeld, N., Krikorian, R. & Cohen, D., 2004. The Internet of things. *Scientific American*, 27 September. pp.1-5.

Gomez, C. & Paradells, J., 2010. Wireless Home Automation Networks: A Survey of Architectures and Technologies. *IEEE Communications Magazine* 48(6), pp.92-101.

Google-I/O, A., 2011. *YouTube*. [Online] Available at: <http://www.youtube.com/watch?v=zD3Q4kJhD5w> [Accessed 2013].

Graham, B., Tracey, D., Timmons, N. & Morrison, J., 2011. Design and performance analysis of a low cost light energy harvester for Wireless Sensors. In *2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies.*, 2011. IEEE Conference Publications.

- Granzer, W., Reinisch, C. & Kastner, W., 2010. Future challenges for building automation: Wireless and security. In *IEEE International Symposium on Industrial Electronics (ISIE)*., 2010. IEEE Conference Publications.
- Gratton, D., 2007. *Developing Practical Wireless Applications*. 1st ed. Elsevier digital press.
- Greenja, ?, 2013. *Texas Instruments E2E Community*. [Online] Available at: http://e2e.ti.com/support/low_power_rf/f/538/t/263195.aspx [Accessed 2013].
- Grønbaek, I., 2008. Architecture for the Internet of Things (IoT): API and interconnect. In *Second International Conference on Sensor Technologies and Applications, SENSORCOMM '08.*, 2008. IEEE Conference Publications.
- Gugliotta, A. & Guarise, A., 2009. *hydramiddleware*. [Online] Available at: http://www.hydramiddleware.eu/downloads.php?cat_id=1&download_id=108 [Accessed 2013].
- Gupta, R., Talwar, S. & Agrawal, D., 2002. Jini home networking: a step toward pervasive computing. *Computer* 35(8), pp.34-40.
- Hamzi, A., Koudil, M., Jamont, J. & Ocelllo, M., 2013. Multi-Agent Architecture for the Design of WSN Applications. *Wireless Sensor Network* 5(2), pp.14-15.
- Hannon, C. & Brunell, L., 2005. A Distributed Multi-Agent Framework for Intelligent Environments. *Journal on Systemics, Cybernetics and Informatics* 3(1).
- Hay, D., 2011. *UML and Data Modeling: A Reconciliation*. 2nd ed. Technics Publications.
- Hedetniemi, S. & Liestman, A., 1988. A Survey of Gossiping and Broadcasting in Communication Networks. *IEEE networks* 18(4), pp.319-49.
- Heinzelman, W., Kulik, J. & Balakrishnan, H., 1999. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom.*, 1999. ACM New York.
- Helmke, R.J., 2013. *Homeseer*. [Online] Available at: <http://www.homeseer.com/android/> [Accessed 2013].
- Hui, J., 2008. Extending IP to Low-Power, Wireless Personal Area Networks. *IEEE Internet Computing* 12(4), pp.37-45.
- Huo, H., Xu, Y., Bilen, C. & Zhang, H., 2009. Coexistence Issues of 2.4GHz Sensor Networks with Other RF Devices at Home. In *Third International Conference on Sensor Technologies and Applications.*, 2009. IEEE computer society.
- IBM Sales & Distribution, W.P., 2010. *The IBM vision of a smarter home*. NY, USA: IBM Corporation IBM.

- Jacob, C. & Steglich, S., 2009. Modeling dynamic service behavior using context functions. In *First International Conference on Networked Digital Technologies, NDT '09.*, 2009. IEEE Conference Publications.
- Jain, P., Noor, A. & Sharma, V., 2011. Internet of Things An Introduction. In *Proceedings of ASCNT.*, 2011. CDCC.
- Jakkula, V. & Cook, D., 2008. Anomaly detection using temporal data mining in a smart home environment. *Methods Inf Med.* 47(1), pp.70-75.
- Jakkula, V., Cook, D. & Crandall, A., 2007. Temporal pattern discovery for anomaly detection in a smart home. In *International Conference on Intelligent Environments, IE 07. 3rd IET.*, 2007. IET Conference Publications.
- Jin, C. & Kunz, T., 2011. Smart home networking: Combining wireless and powerline networking. In *International conference on Wireless Communications and Mobile Computing.*, 2011. IEEE Conference Publications.
- Johnson, 2012. *What Is a Smart Building.* [Online] Available at: <https://whatspossible.johnsoncontrols.com/docs/DOC-1131> [Accessed 2013].
- JohnsonControl, 2012. *Putting Building Information To Work: Smart Buildings.* [Online] Available at: <https://whatspossible.johnsoncontrols.com/docs/DOC-1134/version/1> [Accessed 2013].
- Kaila, L. & Mikkonen, J., 2008. *The eHome—a Practical Smart Home Implementation.* Whitepaper. www.pervasive2008.org/Papers/Workshop/w3-02.pdf.
- Kamath, S. & Lindh, J., 2013. *Application note ANO92.* Texas instruments.
- Kasteren, T., Engleblenne, G. & Kröse, B., 2010. An activity monitoring system for elderly care using generative and discriminative models. *Personal and Ubiquitous Computing* 14(6), pp.489-98.
- Kasteren, T., Kröse, B. & Englebienne, G., 2011. Human activity recognition from wireless sensor network data: Benchmark and software. In Kasteren, T., Kröse, B. & Englebienne, G. *Activity Recognition in pervasive intelligent environments.* 2nd ed. Springer link.
- Kasteren, T.V., Noulas, A., Englebienne, G. & Kröse, B., 2008a. Accurate Activity Recognition in a Home Setting. In *Proceedings of the 10th international conference on Ubiquitous computing.* Seoul, Korea., 2008a. ACM New York.
- Kasteren, T., Noulas, A. & Kröse, B., 2008b. Conditional random fields versus hidden markov models for activity recognition in temporal sensor data. In *Proceedings of the 14th Annual Conference of the Advanced School for Computing and Imaging.* Delft, 2008b. ASCI'08.

Kawsar, F., Nakajima, T., Park, J. & Yeo, S., 2009. Design and implementation of a framework for building distributed smart object systems. *The Journal of Supercomputing* 54(1), pp.4-28.

Kevin, R., Wadleigh, R. & Crawford, I., 2000. *Software Optimization for High-Performance Computers*. 1st ed. Hewlett-Packard professional books.

Khan, S. & Aziz, H., 2009. Clustered home area network: A beacon enabled IEEE 802.15. 4 approach. In *IEEE 9th Malaysia International Conference on Communications (MICC)*., 2009. IEEE Conference Publications.

Khoo, L.Q., 2013. *comp2013 group7 Wiki*. [Online] Available at: http://comp2013group7.wikia.com/wiki/User_blog:Changtau2005/Literature_review [Accessed 2013].

Kim, J. et al., 2012. Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes. In *IEEE International Conference on Intelligent Environments (IE)*., 2012. IEEE Conference Publications.

Kim, D. & Kim, D., 2006. An Intelligent Smart Home Control Using Body Gestures. In *International Conference on Hybrid Information Technology*., 2006. IEEE Conference Publications.

Kim, H., Song, W. & Kim, S., 2008. Light-weighted Internet protocol version 6 for low-power wireless personal area networks. In *IEEE International Symposium on Consumer Electronics, ISCE*., 2008. IEEE Conference Publications.

Kondo, T., 2011. Revised GMDH-type neural network using artificial intelligence and its application to medical image diagnosis. In *IEEE Workshop On Hybrid Intelligent Models And Applications (HIMA)*., 2011. IEEE Conference Publications.

Kulik, J., Heinzelman, W. & Balakrishnan, H., 2002. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *Wireless networks* 8(2/3), pp.169-85.

Lafferty, J., McCallum, A. & Pereira, F., 2001. Conditional Random Fields: Probabilistic Models. In *Proceedings of the 18th International Conference on Machine Learning*., 2001. ACM.

Lameski, P., Zdravevski, E., Kulakov, A. & Davcev, D., 2011. Architecture for Wireless Sensor and Actor Networks Control and Data Acquisition. In *International Conference on Distributed Computing in Sensor Systems and Workshops*., 2011. IEEE Conference Publications.

Latour, B., 1998. From the World of Science to that of Research. *Science* 280(5361), pp.208-09.

- Lee, P., Eu, Z., Han, M. & Tan, H., 2011. Empirical modeling of a solar-powered energy harvesting wireless sensor node for time-slotted operation. In *IEEE Wireless Communications and Networking Conference.*, 2011. IEEE Conference Publications.
- Lee, H. & Kim, J., 2007. An approach for content sharing among UPnP devices in different home networks. *IEEE Transactions on consumer Electronics* 53(4), pp.1419-26.
- Liang, N., Fu, L. & Wu, C., 2002. An integrated, flexible, and Internet-based control architecture for home automation system in the Internet era. In *IEEE International Conference on Robotics and Automation. Proceedings. ICRA '02.*, 2002. IEEE Conference publications.
- Li, X., Lu, R., Liang, X. & Shen, X., 2011b. Smart community: an internet of things application. *IEEE Communications Magazine* 49(11), pp.68-75.
- Li, X. & Murata, T., 2010. Customizing Knowledge-based Recommender System by Tracking Analysis of User Behavior. In *IEEE 17Th International Conference on Industrial Engineering and Engineering Management (IE&EM).*, 2010. IEEE Conference Publications.
- Liu, Q., Diao, L. & Tu, G., 2010. The Application of Artificial Intelligence in Mobile Learning. In *8th Workshop on Intelligent Solutions in Embedded Systems.*, 2010. IEEE Conference Publications.
- Liutkevičius, A., Vrubliauskas, A., Kazanavičius, E. & Imbrasas, D., 2011. Smart home services development, provisioning, and management framework. *Information Technology and Control* 40(1), pp.163-69.
- Liu, J. & Tong, W., 2010. Dynamic Services Model Based on Context Resources in the Internet of Things. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on.*, 2010. IEEE Conference Publications.
- Li, L., Xiaoguang, H., Ke, C. & Ketai, H., 2011a. The Applications Of WiFi-based Wireless Sensor Network In Internet Of Things And Smart Grid. In *IEEE conference on Industrial Electronics and Applications (ICIEA).*, 2011a. IEEE Conference Publications.
- Li, B. & Yu, j., 2011. Research and Application on the Smart Home Based on Component Technologies and Internet of Things. *Procedia Engineering* 15(1), pp.2087-92.
- López, T., Ranasinghe, D., Harrison, M. & MaFarlane, D., 2012. Adding sense to the Internet of Things. *Pers. Ubiquit Comput* 16(3), pp.291-308.
- Luger, G., 2008. *Artificial intelligence, Structures and Strategies for Complex Problem Solving.* 3rd ed. Pearson, Addison Wesley.
- Lutz, K. & Konig, A., 2010. Minimizing power consumption in wireless sensor networks by duty-cycled reconfigurable sensor electronics. In *2010 8th Workshop on Intelligent Solutions in Embedded Systems.*, 2010. IEEE Conference Publications.

- Lu, C. & Wu, Q., 2011. Interconnecting ZigBee and 6LoWPAN wireless sensor networks for smart grid applications. In *Fifth International Conference on Sensing Technology.*, 2011. IEEE Conference Publications.
- Lynggaard, P., 2011. A Smart Home Development Tool combining Simulation, Emulation and Real-World IoT. In *Electronic proceeding (extended abstract).*, 2011. AAU / CMI Conference:Internet of Things –Our Environment Becomes Intelligent.
- Lynggaard, P., 2012a. A Distributed Smart Home Artificial Intelligence System. *Scientific Journal of the University of Szczecin no. 763 TOM II*, pp.521-31.
- Lynggaard, P., 2012b. Distributed Smart Home Activity Recommender System Using Hidden Markov Model Principles. *Scientific Journal of the University of Szczecin no. 762 TOM I*, pp.359-69.
- Lynggaard, P., 2013a. A Distributed Smart Home System Based on Smart Objects. In *Electronic proceeding (submitted).*, 2013a. AAU / CMI conference: Developing the Future ICT Infrastructure.
- Lynggaard, P., 2013b. Smart Home Learning and Activity Prediction using a Simplified Naïve Bayes Classifier and Simple Sensors. *IEEE Internet of Things Journal (submitted).*
- Lynggaard, P., 2013c. Smart Home Wireless Sensor Nodes, Addressing the Challenges using Smart Objects and Artificial Intelligence. In *meeting proceedings / WWRF Outlook /IEEE Vehicular Technology Magazine (awaiting publication).* Vancouver, 2013c. WWRF.
- Lynggaard, P., 2013d. Using ORB in a Smart Home Portable User Interface. *IEEE Transactions on visualization and computer graphics (submitted).*
- Mao, M. et al., 2010. Solution to Intelligent Management and Control of Digital Home. In *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on.*, 2010. IEEE Conference Publications.
- Ma, Y., Wei, N. & Lv, M., 2012. Performance Analysis of Wireless Network Based on IEEE 802.15.4 in Smart Home Environment. In *Fourth International Conference on Multimedia Information Networking and Security.*, 2012. IEEE Conference Publications.
- Medjahed, H., Istrate, D., Boudy, J. & Dorizzi, B., 2009. Human Activities of Daily Living Recognition Using Fuzzy Logic For Elderly Home Monitoring. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE.*, 2009. IEEE Conference Publications.
- Mehairi, S.O., Barada, H. & Qutayri, M., 2007. Integration of Technologies for Smart Home Application. In *IEEE/ACS International Conference on Computer Systems and Applications.*, 2007. IEEE Conference Publications.
- Mingyi, M., Qian, M.Q., Jianjun, L. & Zhicheng, C., 2010. Solution to Intelligent Management and Control of Digital Home. In *International Conference on Biomedical Engineering and Informatics (BMEI).*, 2010. IEEE Conference Publications.

- Mocanu, I., Negreanu, L. & Flora, A., 2013. A Multi-Agent System for Service Acquiring in Smart Environments. *Intelligent Distributed Computing VI Studies in Computational Intelligence* 446(1), pp.297-306.
- Naeem, U. & Bigham, J., 2009. Activity recognition in the home using a hierarchal framework with object usage dat. *Journal of Ambient Intelligence and Smart Environments* 1(4), pp.335-50.
- Nazerfard, E., Rashidi, P. & Cook, D., 2010. Discovering Temporal Features and Relations of Activity Patterns. In *IEEE International Conference on Data Mining Workshops.*, 2010. IEEE Conference Publications.
- Normann, A. & Jamison, A., 2011. *Knowledge Making in Transition. On the Changing Contexts of Science and Technology.* University of Pittsburgh Press.
- Orr, R. & Abowd, G., 2000. The smart floor: a mechanism for natural user identification and tracking. In *CHI'00 extended abstracts on Human factors in computing systems.*, 2000. ACM.
- Park, V. & Corson, M., 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE.*, 1997. IEEE Conference Publications.
- Park, S., Kim, J. & Kim, S., 2013. An object-based middleware supporting efficient interoperability on a smart home network. *Multimedia Tools and Applications* 63(1), pp.227-46.
- Pensas, H. & Vanhala, J., 2010. WSN Middleware for Existing Smart Homes. In *Fourth International Conference on Sensor Technologies and Applications.*, 2010. IEEE Computer Society.
- Perumal, T., Rmali, A.R. & Leong, C.Y., 2011. Interoperability Framework for Smart Home Systems. *IEEE Transactions on Consumer Electronics* 57(4), pp.1607-16011.
- Pole, D. & Mackworth, A., 2010. *Artificial Intelligence, Foundations of Computational Agents.* Cambridge university press.
- Pucker, L. & Kaiser, D., 2013. *Software Defined Radio Forum.* [Online] Available at: www.sdrforum.org [Accessed 2013].
- Rabiner, L., 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2), pp.257-86.
- Radi, K., Mahdi, S., Sakhaei, R. & Hussein, M., 2011. A utilization of wireless sensor network in smart homes, reviewing its combination with modern cellular networks. In *International Conference on Communication and Industrial Application (ICCIA).*, 2011. IEEE Conference Publications.

- Rashid, J., Broxvall, M. & Saffiotti, A., 2012. A middleware to integrate robots, simple devices and everyday objects into an ambient ecology. *Pervasive and Mobile Computing* 8(4), pp.522-41.
- Rashidi, P., Cook, D., Holder, L. & Edgecombe, M., 2011. Discovering Activities to Recognize and Track in a Smart Environment. *IEEE Transactions on Knowledge and Data Engineering* 23(4), pp.527-39.
- Rathnayaka, A., Potdar, V. & Kuruppu, S., 2012. Energy Resource Management in Smart Home: State of the Art and Challenges Ahead. In M'Sirdi, N., Namaane, A., Howlett, R. & Jain, L. *Sustainability in Energy and Buildings*. Springer Berlin Heidelberg. pp.403-11.
- Reinisch, C., Kofler, M. & Kastner, W., 2010. ThinkHome: A Smart Home as Digital Ecosystem. In *IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST)*., 2010. IEEE Conference Publications.
- Reis, G. & Luiz, A., 2012. Introduction to the Software-defined Radio Approach. *Latin America Transactions, IEEE (Revista IEEE America Latina)* 10(1), pp.1156-61.
- Rish, I., 2001. An empirical study of the naive bayes classifier. In *In IJCAI 2001 workshop on empirical methods in artificial intelligence.*, 2001. IJCAI.
- Russell, S. & Norvig, P., 2009. *Artificial intelligence, a modern approach*. 2nd ed. Prentice Hall; 3 edition.
- Sagi, M. & Mijic, D., 2012. Smart home automation. In *Telecommunication Forum (TELFOR)*., 2012. IEEE Conference Publications.
- Satoh, I., 2009. Location-aware communications in smart environments. *Information Systems Frontiers* 11(5), pp.501-12.
- Sauter, T. & Soucek, S., 2011. The Evolution of Factory and Building Automation. *IEEE Industrial Electronics Magazine* 5(3), pp.35-48.
- Shah, M., Bashir, A. & Chauhdary, S., 2009. Mobile Ad Hoc Computational Grid for Low Constraint Devices. In *IEEE International Conference on Future Computer and Communication.*, 2009. IEEE Conference Publications.
- Shi, Q., 2007. Power Management in Networked Sensor Radios A Network Energy Model. In *Sensors Applications Symposium.*, 2007. IEEE Conference Publications.
- Silva, T.R.M.B., Loureiro, A. & Ruiz, L.B., 2009. A Conflict Resolution Methodology for Collective Ubiquitous Context-Aware Applications. In *Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on.*, 2009. IEEE Conference Publications.
- Silva, C., Morikawa, C. & Petra, I., 2012. State of the art of smart homes. *Engineering Applications of Artificial Intelligenc* 25(7), pp.1313-21.

- Sohrabi, K. & Pottie, G., 1999. Performance of a novel self-organization protocol for wireless ad-hoc sensor networks. In *IEEE Vehicular Technology Conference.*, 1999. IEEE Conference Publications.
- Sohraby, K., Minoli, D. & Znati, T., 2007. *Wireless Sensor Networks*. 2nd ed. Wiley Interscience.
- Song, Y., Keeney, J. & Conlan, O., 2011. An ontology-driven approach to support wireless network monitoring for home area networks. In *International Conference on Network and Service Management (CNSM), 7th.*, 2011. IEEE Conference Publications.
- Sørensen, L. & Skouby, K., 2009. *User scenarios 2020 -a worldwide wireless future*. WWRF No. 4.
- Starsinic, M., 2010. System Architecture Challenges in the Home M2M network. In *IEEE Applications and Technology Conference (LISAT).*, 2010. IEEE Conference Publications.
- Sudhaakar, S., Qiao, C., Yoon, S. & Zhao, J., 2009. A MAC protocol for real-time sensing applications using asymmetric transceivers. In *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, 2009. MASS '09.*, 2009. IEEE Conference Publications.
- Sundmaeker, H., Guillemin, P., Friess, P. & Woelfflé, S., 2010. Vision and Challenges for Realising the Internet of Things. European Commission - Information Society and Media DG.
- Sutton, C. & McCallum, A., 2007. An Introduction to Conditional Random Fields for Relational Learning. In Getoor, L. & Taskar, B. *Introduction to statistical relational learning*. 3rd ed. MIT.
- Takizawa, S., Komuro, N. & Sakata, S., 2012. Routing Control Scheme Prolonging Network Lifetime in a 6LoWPAN WSN with Power-supplied and Battery-powered Nodes. In *2012 IEEE CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE (CCNC).*, 2012. IEEE Conference publications.
- Tanasiev, V. & Badea, A., 2012. Energy Efficiency through Intelligent Management of Building Equipment. *U.P.B. sci. Bull, series C 74(1)*, pp.49-57.
- Tapia, E.M., Intille, S.S. & Larson, K., 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. *Pervasive Computing In Pervasive Computing 3001(1)*, pp.158-75.
- Top-ten-reviews, 2013. *Top-ten-reviews*. [Online] Available at: <http://home-automation-systems-review.toptenreviews.com/> [Accessed 2013].
- Trevennor, A., 2012. Smart Home Enablers. In *Practical AVR Microcontrollers*. 2nd ed. Springer Link. pp.305-31.
- Tribble, A., 2008. The software defined radio: Fact and fiction. In *IEEE Radio and Wireless Symposium.*, 2008. IEEE Conference Publications.

Turner, K., 2011. Flexible management of smart homes. *Journal of Ambient Intelligence and Smart Environments* 3(2), pp.83-109.

Turner, B., 2013. *Discovery curiosity*. [Online] Available at: <http://dsc.discovery.com/tv-shows/curiosity/topics/ways-artificial-intelligence-will-affect-our-lives.htm> [Accessed june 2013].

Tu, M., Shin, D. & Shin, D., 2009. Fundamentals and Design of Smart Home Middleware. In *International Joint Conference on Computational Sciences and Optimization.*, 2009. IEEE Computer Society.

Uckelmann, D. & Harrison, M., 2011. An Architectural Approach Towards the Future Internet of Things. In F. Michahelles, ed. *Architecting the Internet of Things*. 1st ed. Springer. pp.1-24.

UniversityOfEssex, 2013. *iSpace*. [Online] Available at: <http://cswww.essex.ac.uk/ieeg/idorm.htm> [Accessed 2013].

Valhouli, C.A., 2010. *The Internet of Things Networked objects and smart devices*. The Hammersmith group.

Wang, S., 2010. *Intelligent Buildings and Building Automation*. 2nd ed. New York: Spon press.

Wu, J., Huang, L., Wang, D. & Shen, F., 2008. R-OSGi-based Architecture of Distributed Smart Home System. *IEEE Transactions on Consumer Electronics* 54(3), pp.1166-72.

Wu, Y. & Shao, P., 2012. Middleware-Based Distributed Data Acquisition and Control in Smart Home Networks. In *Proceedings of the 2nd International Congress on Computer Applications and Computational Science.*, 2012. Springer Link.

Xiaojuan, Z., 2010. he Strategy of Smart Home Control System Design based on Wireless Network. *International Conference on Computer Engineering and Technology (ICCET)*, pp.4-37 to 4-40.

Xuenei, L., Gang, L. & Li, L., 2008. RFID based Smart Home architecture for improving lives. In *International Conference on Anti-counterfeiting, Security and Identification ASID.*, 2008. IEEE Conference Publications.

Xu, Y. et al., 2013. Architecture and Routing Protocols for Smart Wireless Home Sensor Networks. *International journal of distributed sensor networks* 2013(1), p.14.

Yan, D. & Dan, Z., 2010. ZigBee-based Smart Home system design. In *International Conference on Advanced Computer Theory and Engineering.*, 2010. IEEE Conference Publications.

- Yao, F., Yang, S. & Zheng, W., 2010. Mitigating interference caused by IEEE 802.11b in the IEEE 802.15.4 WSN within the environment of smart house. In *IEEE International Conference on Systems Man and Cybernetics (SMC)*., 2010. IEEE Conference Publications.
- Ye, X. & Huang, J., 2011. A Framework for Cloud-based Smart Home. In *International Conference on Computer Science and Network Technology*., 2011. IEEE Conference Publications.
- Yoon, S. et al., 2007. QoMoR: AQoSaware MAC Protocol Using Optimal Retransmission for Intra-Vehicular Wireless Sensor Networks. In *2007 Mobile Networking for Vehicular Environments*., 2007. IEEE Conference Publications.
- Yu, Y.-C., You, S.-c.D. & Tsai, D.-R., 2010. A Calendar Oriented Service for Smart Home. In *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*., 2010. IEEE Conference Publications.
- Zhang, Q., Chen, Z. & Zhang, P., 2012. *Internet of Things Applied in the Home-Based Caring System for the Aged*. 2nd ed. Springer link.
- Zhao, X., 2010. The strategy of smart home control system design based on wireless network. In *International Conference on Computer Engineering and Technology (ICCET)*., 2010. IEEE Conference Publications.
- Zhao, J., Qiao, C., Sudhaakar, R. & Yoon, S., 2013. Improve Efficiency and Reliability in Single-Hop WSNs with Transmit-Only Nodes. *IEEE Transactions on Parallel and Distributed Systems* 24(3), pp.520-34.
- Zheng, H., Wang, H. & Black, N., 2008. Human activity detection in smart home environment with self-adaptive neural networks. In *IEEE International Conference on Networking, Sensing and Control, ICNSC*., 2008. IEEE Conference Publications.
- Zoref, L., Bregman, D. & Dori, D., 2009. Networking Mobile Devices and Computers in an Intelligent Home. *International Journal of Smart Home* 3(4), pp.15-23.