journal homepage: www.ijmijournal.com

Development and evaluation of SOA-based AAL services in real-life environments: A case study and lessons learned

Erlend Stav[a,*], *Ståle Walderhaug*[a,d], *Marius Mikalsen*[a], *Sten Hanke*[b], *Ivan Benc*[c]

[a] *SINTEF ICT, P.O. Box 4760 Sluppen, NO-7465 Trondheim, Norway*
[b] *Austrian Institute of Technology GmbH, Austria*
[c] *Ericsson Nikola Tesla D.D., Croatia*
[d] *Department of Computer Science, University in Tromsø, Norway*

ARTICLE INFO

Article history:
Received 28 February 2010
Received in revised form
17 August 2010
Accepted 10 March 2011

Keywords:
Ambient Assisted Living
Service-Oriented Architecture
Information systems
Model-driven architecture
Model-driven development
Case study

ABSTRACT

Intro: The proper use of ICT services can support seniors in living independently longer. While such services are starting to emerge, current proprietary solutions are often expensive, covering only isolated parts of seniors' needs, and lack support for sharing information between services and between users. For developers, the challenge is that it is complex and time consuming to develop high quality, interoperable services, and new techniques are needed to simplify the development and reduce the development costs.

This paper provides the complete view of the experiences gained in the MPOWER project with respect to using model-driven development (MDD) techniques for Service Oriented Architecture (SOA) system development in the Ambient Assisted Living (AAL) domain.

Method: To address this challenge, the approach of the European research project MPOWER (2006–2009) was to investigate and record the user needs, define a set of reusable software services based on these needs, and then implement pilot systems using these services. Further, a model-driven toolchain covering key development phases was developed to support software developers through this process. Evaluations were conducted both on the technical artefacts (methodology and tools), and on end user experience from using the pilot systems in trial sites.

Results: The outcome of the work on the user needs is a knowledge base recorded as a Unified Modeling Language (UML) model. This comprehensive model describes actors, use cases, and features derived from these. The model further includes the design of a set of software services, including full trace information back to the features and use cases motivating their design. Based on the model, the services were implemented for use in Service Oriented Architecture (SOA) systems, and are publicly available as open source software. The services were successfully used in the realization of two pilot applications. There is therefore a direct and traceable link from the user needs of the elderly, through the service design knowledge base, to the service and pilot implementations.

* *Corresponding author*. Tel.: +47 7359 0371; mobile: +47 9341 1597.
E-mail address: erlend.stav@sintef.no (E. Stav).
1386-5056/$ – see front matter
doi:10.1016/j.ijmedinf.2011.03.007

The evaluation of the SOA approach on the developers in the project revealed that SOA is useful with respect to job performance and quality. Furthermore, they think SOA is easy to use and support development of AAL applications. An important finding is that the developers clearly report that they intend to use SOA in the future, but not for all type of projects. With respect to using model-driven development in web services design and implementation, the developers reported that it was useful. However, it is important that the code generated from the models is correct if the full potential of MDD should be achieved.
The pilots and their evaluation in the trial sites showed that the services of the platform are sufficient to create suitable systems for end users in the domain.
Conclusions: A SOA platform with a set of reusable domain services is a suitable foundation for more rapid development and tailoring of assisted living systems covering reoccurring needs among elderly users. It is feasible to realize a tool-chain for model-driven development of SOA applications in the AAL domain, and such a tool-chain can be accepted and found useful by software developers.

1. Introduction

The proper use of ICT services can support seniors in living independently longer [1,2]. While such services are starting to emerge, current proprietary solutions are often expensive, covering only isolated parts of seniors' needs, and lack support for sharing information between services and between different stakeholders. It is complex and time consuming to develop high quality services, and support is needed to simplify the development and reduce the development costs. By providing a knowledge base of elderly user needs, software development tools and out of the box software components, we can reduce the costs of delivering ICT services to seniors and as such increase the number and quality of services delivered [2].

In the seminal paper about health information systems from 2006 [3], Haux argues that it is necessary to explore new architectural styles that focus on trans-institutional access to patient data. Finally he concludes that: "informatics methodology and technology is expected to facilitate continuous quality of care in aging societies". The Service-Oriented Architecture (SOA) [4] architectural style is designed upon the ideas of cross-organizational information sharing and reuse.

This paper reports on the design, development and evaluation of a software platform for building assistive services for elderly users, with special focus on users with cognitive challenges such as mild dementia. The platform builds upon the concepts of SOA, and was developed in the context of the European research project MPOWER[1] [5] (October 2006–June 2009). The SOA approach facilitates reuse of domain services between disparate systems and incorporation of domain knowledge into software artefacts, and its flexibility facilitates tailoring and personalization of the systems to the needs of individual users or user groups. The main contributions presented in the paper are:

1. *A knowledge base of user needs*: The user needs knowledge base was developed through an extensive requirement phase including 143 persons including seniors professional care providers, family, and physicians.
2. *Model driven toolchain development*: Using a model-driven development toolchain, an architectural description including UML [6] use cases and information models for the target domain was specified.
3. *A set of reusable domain specific software services*: Based on the identified requirements in the domain, requirements for tool support and reusable services were derived. A set of 25 SOA services were designed and implemented according to the SOA for HL7 methodology [7] and task-centric approach described by Erl [8].
4. *Two pilot systems developed using domain software services*: To validate the usefulness of the SOA services, one information-sharing and one sensor-centric pilot system were implemented and deployed to real-life settings. The two pilot systems were built on top of the 25 domain services in order to reduce the development effort and time.

Evaluations have been conducted on both technical artefacts (methodology and tools), and on end user experience. Two evaluations were conducted on the developers: the use of model-driven development tools and the use of SOA in application development. The evaluation of end user experience was conducted in the period from February 2008 to June 2009. Five elderly (age 65–92) and their carers (formal and informal) in Norway have used the information-sharing system. The evaluation was conducted in three phases, as reported in [9]. The sensor-centric system was installed in a Polish nursing home and a usability evaluation was done in a three-month period from April to June 2009.

The main technical findings are that a SOA facilitates the development of domain services that may improve the overall system design and performance. However, the reusability of services varies between the functional areas, and this must be taken into account. From an elderly/patient and carer point of view, the pilot systems provided appropriate functionality and reasonable quality of service. For one user of the information-sharing system the activities of daily living were significantly improved.

This paper provides the complete view of the experiences gained in the EU IST project MPOWER with respect to using

[1] http://www.mpower-project.eu/.

model-driven development techniques for SOA service and system development in the AAL domain. In so doing, references to already published works are given, and some quotes from their conclusions are presented. In addition to providing this holistic view, important new results on developers' perceptions of using SOA is provided in Section 4.1. The main referenced works are:

- On user needs investigation and service modelling: [9–11]
- On toolchain design and evaluation: [11–15]
- On pilot system evaluation [9,16]

We conclude that SOA in development of support systems for elderly and their carers (formal and informal) can be beneficial when the services are designed in line with the domain needs and technical environment factors such as network stability are considered carefully. The SOA services are provided as open source in the FREE MPOWER project [17] and are developed further in the EU IST project universAAL.[2] Software companies and system integrators could benefit from both our reported experiences and the reusable software services.

The paper is organized as follows: First we present the context in which the work was carried out, along with the overall research questions and the methods applied. Some parts of the work have been published elsewhere and references along with a short summary of the method and results are included to get the complete picture. Then the results are presented in two sections: a technical and an evaluation section. The technical section focuses on the reusable software artefacts whereas the evaluation part reports from both the developer and elderly/patient/carer evaluations. Then a discussion section follows, addressing the core research questions before some concluding remarks are given.

2. Research questions, context and methods

A core objective of the MPOWER project was to define and implement a platform of reusable software services that could improve the development of novel assistive care applications with respect to time, quality and standards adherence. This paper presents the results from the project, focusing on three questions:

(1) How do the developers perceive the use of model-driven development and SOA for application development in the AAL domain?
(2) What are the set of reusable SOA software services covering reoccurring user needs in the target domain?
(3) Are the reusable SOA software services sufficient to build sustainable AAL systems to be used in real life environments?

[2] universAAL project website: http://www.universaal.org.

2.1. Context

2.1.1. AAL and services

European countries are facing a great challenge in dealing with a steadily aging population. The 2009 Aging Report projections for 2060 state that "Population structures will become increasingly dominated by old people rather than young" [18]. In the period 2010–2060 within the EU, the working population (age 15–64) is forecast to drop by 15 percent, and the number of elderly people aged 65 or more is expected to double. The total population will have only a slight increase. Despite an upward trend in the employment rate of women, there will be a shortage of labour to provide care for the elderly. In 2000, the prevalence of dementia was 7.1 million people in Europe, and this number is expected to rise to 16.2 million in 2050 [19].

Ambient Assisted Living (AAL) is emerging as technological support that can help elderly in their activities of daily living, enabling them to live more independently in their preferred environment. Examples of assistance can include giving reminders as a remedy to failing memory, facilitating social interaction with family and friends, monitoring doors, windows, water flow, and stove to improve security and safety in the home, tracking patients that wander off and detecting falls, and sharing of information between subject of care, healthcare personnel and next of kin. To provide the best assistance, these services need to be selected and personalized to each user, and furthermore this personalization may need updates as cognitive and physical condition tend to deteriorate over time.

Even though the needs for assistance and care vary from person to person, there are similarities at a service level. This opens for reuse of a high percentage of the services, given that the services allows for personalization and flexible composition. In the next subsection, Service Oriented Architectures are presented as a technological solution that provides reuse and flexibility to the AAL domain.

2.1.2. Technological context

Service Oriented Architectures (SOA) is an architectural software style that seeks to divide an information system into a set of interconnected stand-alone components that expose their functionality through a clearly defined interface. These components can be discovered and used by one or many applications through standard communication protocols such as the HTTP. The most common implementation of SOA is Web Services. Web Services are now being used by most organizations to share information across systems and platforms, using the Internet/Intranet as a transport medium. Gartner research estimates that Service-Oriented Architecture (SOA) "*will be used in more than 50 percent of new mission-critical operational applications and business processes designed in 2007 and in more than 80 percent by 2010*" [20].

The benefits of SOA are often reported based on key properties of the technology. Kawamoto and Lobach [21] highlights that SOA focuses on encapsulating core business capabilities within independent software services, and leveraging these services by using various business front-ends to fulfil business requirements. The SOA properties they list are; use of business oriented services, message-based interactions with "black-box" implementations; communication over

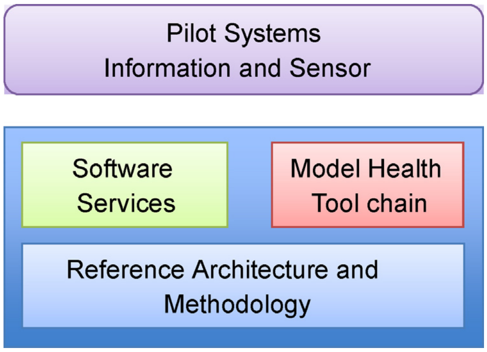

Fig. 1 – Overview of MPOWER platform.

a network; platform neutrality; service description and discovery; and loose coupling between system components. They argue that this provides the following benefits:

- Simpler software design and implementation, by decomposing complex problems into smaller, more manageable ones.
- Improves software reusability through enhanced reuse of existing IT resources.
- Improved adaptability to changing business requirements.
- Cost savings consequent to the above benefits.

Mykkänen et al. argues that the main challenges in health information systems (HIS) include redundant data and functionality, heterogeneous technologies, lack of reuse and many new requirements and local variations healthcare provision and medical practice [22].

The information system architectures in the domain of assistive services for elderly are even more distributed than traditional hospital information systems, and also target a broader set of stakeholders such as home care services and informal (family) carers. Handling the information system complexity is of utmost importance for successful diffusion of information systems that support teamwork treatment and management of patients in their home. In this respect, the SOA promises are highly relevant for assistive care, and the MPOWER project set out to investigate the feasibility of applying SOA to develop an ICT solution to be used in two typical, but completely different assistive services settings: one information-sharing system and one sensor-centric system.

2.2. *Overview and reference architecture for the SOA-based MPOWER platform*

As illustrated in Fig. 1, the MPOWER platform is comprised of four main parts:

1. A reference architecture that conforms to the SOA model [4]. Associated with the reference architecture is a methodology for application development and service lifecycle management.
2. Domain specific software services that consist of medical and social information services, communication services, sensor services, interoperability services, and security services.
3. A model-driven development toolchain supporting the development process, which was applied to the development of the software services. The toolchain was evaluated on the 16 service developers in the project.
4. Two pilot systems: one information-sharing and one sensor-centric. The pilot systems were developed using the domain specific software services, and validated in trails by end-users.

The reference architecture (part 1) is discussed below. The domain specific services are discussed in Section 3.3, the toolchain in Section 3.2 and the pilots in Section 3.4.

The reference architecture and hence the target platform was decided on from a set of criteria that was defined by the project consortium. The overall platform criterion was that it must be robust and available. Furthermore, the platform:

- had to be free and preferably open source
- had to be operating system independent
- should have an active community and a significant market share

An investigation of the SOA reference implementations led us to the IBM SOA Reference Architecture [23]. At the time of investigation, IBM's reference architecture was the main instantiation of OASIS' SOA reference model [4], and IBM did also provide developer support in the form of UML profiles and code examples [24].

The MPOWER adaptation of the IBM SOA Reference Architecture is shown in Fig. 2. It consists of five layers – each layer comprising a set of "components" that conforms to the rules and requirements specified for the layer. The figure also indicates how the different groups of MPOWER services map to the reference architecture. A short description of each layer and its application in MPOWER is shown in Table 1.

2.3. *Design and development method of the platform*

This section describes the methods used during design and development of the MPOWER platform. The methods ranges from how user needs was elicited and used to find the features to support, via how domain specific services were derived based on the user needs and then implemented, and to how the pilot systems were designed and implemented based on the platform. The overall flow of this process is illustrated in Fig. 3. Along with the model-driven tool chain which supports the process illustrated in the figure, the main parts of the figure represent the four main contributions listed in the introduction chapter. The methods for each of these are presented in the following sub-sections.

2.3.1. *User needs*

One of the core ideas of Service-Oriented Architectures, or service-oriented computing in general, is to create software artefacts (services) that have a tight coupling to business needs in the target domain, bridging the gap between IT and business. A well-designed SOA service is an implementation-independent reusable business function that can be utilized by

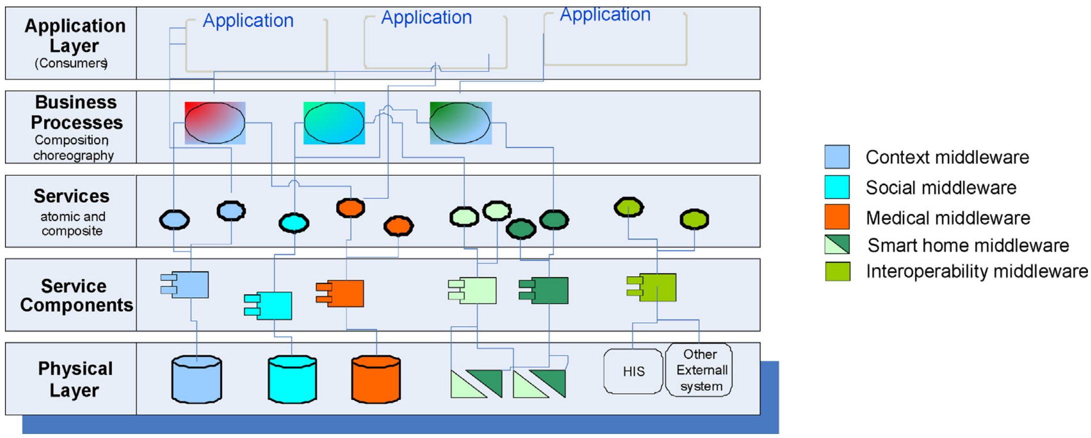

Fig. 2 – MPOWER reference architecture.

Table 1 – Layers of the reference architecture and their use.

Layer	Description	MPOWER use
Application	Provides user interface and application specific components, decoupling these from the underlying (business) services on which they build.	Applications built using MPOWER, including the pilots, belong to this layer. These provide the access point through which the users of health-care applications access the health-care services.
Business processes	Defines the business rules and process of the applications. Services are bundled into a flow through orchestration or choreography, and thus act together in supporting use cases and business process of the application.	Used to define business rules of the MPOWER pilot applications. An example of an assistive care business process is management of a shared calendar where calendar, patient and caregiver information, and medical plans are accessed through a set of services and service components.
Services	Provides services available for invocation. Service implementations may use service components in their realization, and expose their functionality through service interface descriptions. Services can be made available for service discovery through a registry.	The main functionality of the MPOWER platform is provided as services. These are described in Section 3.3 of this paper.
Service components	Exposes the functionality of the components and databases in the resource layer. The service components provide a high-level access to their information and control functions.	A typical service component in MPOWER is a Smart House sensor driver that encapsulates and implements the sensor communication logic for the higher layer services.
Physical	Consists of databases, existing custom built applications, and low level resource such as physical sensors and actuators.	In MPOWER, examples are databases storing medication and administrative information, and (smart) sensors for e.g. physiological monitoring and door control.

applications using open standard protocols across networks. There is a plethora of methodologies that describe how to properly design a SOA service, and common for all of them is to understand the target (business domain).

The AAL domain is large and involves a wide range of stakeholders that each has responsibilities and information needs. After an initial literature review, the project initiated a thorough user needs investigation in Norway, the Netherlands, Austria and Poland. The goal of the investigation was to capture the needs of the most relevant stakeholders, harmonize and prioritize them and finally map them to functional or non-functional system features (high-level requirements).

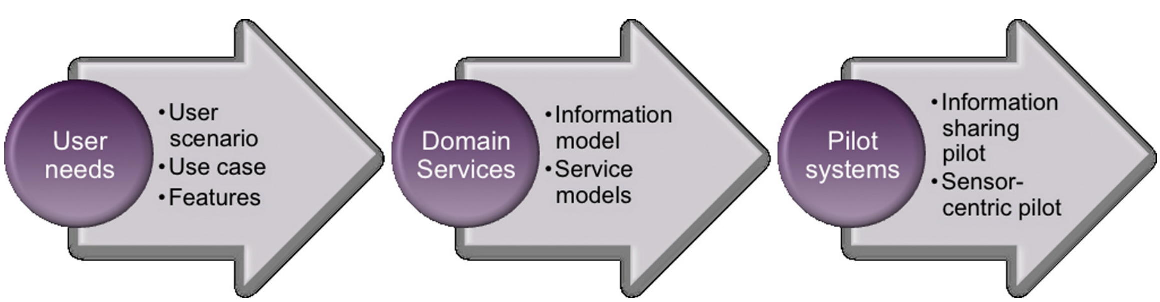

Fig. 3 – The overall design and development process.

The main steps in this process were:

1. Recruit domain stakeholders in the four European countries. Care organizations in each country used their networks.
2. Use qualitative methods to capture the activities and information needs of the main stakeholders.
3. Describe the needs in terms of activities of daily living (ADL) – an everyday scenario that illustrates the problems.
4. In a joint domain (care) expert and technical expert effort, describe scenario where a future IT system assists in solving ADL problems described in the scenarios.
5. Using UML Use case modelling, identify stakeholders, use cases and relationships. Model these in a common model using best practice modelling principles and naming/concept conventions from domain standards (e.g. EN 13940 – Continuity of Care standard from CEN TC 251 [25]) and relevant taxonomies [26].
6. From the use case models, derive system features into functional groups.

A total of 143 persons (family carers, dementia experts, patients and care providers) participated in the investigation. Table 2 shows a summary of the method for data collection and, respondent characteristics and relationship type.

In the modelling process for the use cases and the features traceability were maintained between the modelling elements and back to the scenarios. Traceability is described further in [12,27].

2.3.2. Model-driven toolchain development

To assist the use case and feature modelling, as well as the service design process, a model-driven development toolchain was specified and shared between the researchers. The toolchain is a UML [6] modelling tool customized with a domain specific UML profile and transformation scripts for web services development. The toolchain supported the complete development process from user scenarios through service design and service/application development. Table 3 shows the tools that were used and the artefacts that were created in each step.

The model driven development tool and the accompanying documentation enforce the developers to create explicit links between the services models, features, use cases and scenarios, creating a coherent and fully navigable model of models. More details on the development methodology and toolchain are provided in [13,15].

2.3.3. Service modelling and development

To specify the services modelling process was specified which combined the SOA for HL7 methodology and the IBM Software Service profile methodology. The main steps of the service modelling process were:

1. Identify the service candidate. Create explicit trace links from the service candidate to the (user needs) features it realizes. Use a UML class model element stereotyped with «ServiceProvider» to represent the service candidate application, and a UML port element stereotyped with «Service» on the UML class to represent the service itself.
2. Identify the interface(s) that the service provides. A rule of thumb is to separate read and write operations into different interfaces. Due to some minor bugs in the Java IDE, the services were designed to only provide one interface each. A service interface is a UML Interface stereotyped with «ServiceSpecification».
3. Identify the interface operations. To support the identification of relevant operations, the features specified in the user needs provided valuable input. In many cases, there was a one to one mapping between operations and features, e.g., feature "add Calendar event" is an operation "addNewCalendarEvent" on the calendar service interface.
4. Design the messages (documents passed as request and response) for each operation. Use information model elements from the domain information model do design message-classes. Update the operations to use the messages for input and output parameters.
5. Generate web service description files (WSDL) and documentation (RTF file and html).
6. Use the WSDL files to generate web service framework in a Java development tool (e.g. Netbeans). Manually program the operation logic using the feature and use case relationships as domain knowledge input.
7. Create the web service deployment file and test it on the Glassfish application server.

More details on the service modelling and development process are provided in [13,15].

2.3.4. Pilot systems

Two pilot systems were developed to validate the usefulness of the domain software services of the platform in developing assistive systems. The pilot systems had different characteristics with respect to complexity and user/system interaction requirements.

The approach followed to define, design and implement the pilot systems was based on the following main steps:

- Firstly, user scenarios were prioritized, to decide on which scenarios was of most importance for the target user groups.
- Secondly, pilot system designs were created, supporting high-priority scenarios.
- Lastly, the designs were mapped to the reusable services of the platform, and the pilot applications were implemented.

The two pilots systems defined were:

- *Information-sharing system*. The objective of this pilot system was to create a common plan solution which allowed sharing of information between the elderly, their professional caregivers and the informal caregivers (family/relatives).
- *Sensor-centric system*. This pilot is a Smart House system concentrating on users who need to keep or improve their quality of life, e.g. older or handicapped people; including people with dementia, who's physical and/or cognitive capabilities to perform daily tasks are reduced. From an engineering perspective a "Smart House" is a blending of wired and wireless technologies, sensors and processing

Table 2 – Overview of user needs investigation participation.

Country/target group	Method for data collection	Respondent characteristics	Relation
Norway Target group 1: Family carers	KJ-method	Three men and three women.	Spouse 1, Son/daughter 3, Sibling 1 Friend 1
Norway Target group 2: Expert panel	Questionnaire on mail	Sent to experts on dementia and technology	Alzheimer's Society Research workers Nurses and Occupational Therapists from memory clinics
Austria Target group 1: Family carers	KJ-method	Two men and three women	Spouses 3, Child 1 Sibling 1
Austria Target group 2: Experts	Interview using questionnaire as interview guide	One psychologist and three experts on dementia	
The Netherlands: Elderly people	Over Wonen van Ouderen Gesproken" ("Speaking about living for elderly") [28]	Five men and seventeen women	Age 57–93
Poland Target group 1 Experts, professionals, carers	Domain screening tool (questionnaire) exposing awareness and main problems	49 professionals involved in the care of elderly persons and experts	
Poland Target group 2 Elderly people	Interview delivered by the experts guided by detailed questionnaire	40 elderly people	Average age: 69
Poland Target group 3 Professional carers	Detailed questionnaire to be answered anonymously	49 professional carers (nurse, physiotherapists)	

Table 3 – Overview of the tools, artefacts and development phases.

Tools/phase	User needs	Model-driven development toolchain	Software services	Pilot systems
Microsoft Word	User problem and activity scenarios	Service modelling guidelines	Documentation generated from Enterprise Architect as RTF or HTML documents	User guidelines
Sparx Enterprise Architect 7.x		Project template UML Profile for Homecare SOA Model library of Actors		
Eclipse 3.4 with MOFScript plug-in		Transformation Script for WSDL creation	Generated WSDL files	
Netbeans 6.x			Web Service implementation (source code) and SOAPUI tests	Pilot systems implemented as web application (source code)
Glassfish v2.x with OpenESB			Deployment of Web Services (WAR files)	Web applications

units, with the aim of improving the quality of life of the user. In some cases, these technologies, e.g. wireless sensor networks for daily life, can provide possibilities for the person to remain at home and live independently.

2.4. Evaluation method

2.4.1. Evaluation method of model-driven toolchain and platform

An important target user groups for the SOA platform and services are developers of AAL systems. Therefore it was necessary to plan and conduct extensive evaluation in order to validate the proposed tools, services and methodologies. The first research questions presented at the start of Section 2 can be divided into the following sub-questions:

- (SD-1) Does the model-driven toolchain assist in developing the system services?
- (SD-2) Does SOA assist in developing domain-specific applications?

An evaluation addressing the first question, SD-1, is presented in the article from the European Conference on Model Driven Architecture in 2008 [9]. It concludes that the model-driven toolchain supports the development, although some errors in the code generation process made the developers

Table 4 – Overview of subjects (developers) in the evaluation of toolchain and SOA approach.

Work duties	Highest completed education	Years experience with programming/systems engineering
81% are developers	31% Bachelor's degree	From 1 to 20 years
30% involved in architecture	63% Master's degree	Mean: 5.63
31% are doing research	6% Other education	
6% project management		
6% product support		

spend much time on debugging code that should be flawless. Inherent aspects of model-driven development such as traceability and generation of system documentation were found useful.

The evaluation of SD-2, targeting the use of Service-Oriented Architectures was done on the same cohort as SD-1, namely 16 developers in the project. A summary of developers' education and experience is shown in Table 4 ($N = 16$).

A questionnaire was sent to each developer and all developers responded. The questionnaire was structured according to TAM [29,30] and complemented with a set of claims from the SOA literature about the promises of SOA.

The survey was published as an online web survey using LimeSurvey.[3] The 16 developers responded to within one week. The results were exported from LimeSurvey to SPSS[4] and analysed statistically.

2.4.2. *Evaluation of pilot systems built on SOA*

The two fully functional pilot systems were installed in real environments to evaluate the feasibility of the domain software services in application development. The pilot systems were:

- *Information-sharing system* concentrated on the management of the individual plan and aimed at demonstrating the feasibility of the platform in relation to a dynamic sharing of plans and information between elderly and caregivers. From February 2008 to June 2009, six elderly and their caregivers evaluated the information sharing pilot system in Norway. Evaluation was carried out in three phases using interviews and observation as described in [9].
- *Sensor-centric system* aimed at demonstrating the applicability of the platform in relation to the interconnectivity of created services and sensors in the creation of smart home environments. The sensor centric pilot system was evaluated in a residential house for elderly in Poland in the period from February to June 2009.

[3] http://www.limesurvey.org/.
[4] http://www.spss.com/.

In this paper, only excerpts of the pilot system evaluations are presented to demonstrate the feasibility of building sustainable SOA-based systems from the domain services developed. An extensive user evaluation of the pilot systems is outside the scope of this paper.

3. Technical results

3.1. *User needs*

The results from the domain user needs investigation are presented according to the main steps in the investigation process.

The recorded material from the questionnaires, workshops and interviews was used to describe the problem and activity scenarios. The developed artefacts from this process are:

- 18 user scenarios, each with three to four sub-processes were described. These were then subject for analysis by system architects using UML use case modelling.
- 113 UML use cases were described, separated in 10 logical groups. 82 actors (stakeholders and system actors) were identified and modelled. The organization and person actors were harmonized according to the CEN CONTSYS standard [25] whereas services and devices were structured according to the taxonomy by Stefanov [26] (e.g., "Automation and control", "Assistive services", "Assistive devices").
- From each use case, one or more features were described. A total of 145 features were identified and structured into the logical groups.

Figs. 4 and 5 are two examples from the user needs modelling that illustrate the relationships between actors, use cases and features. The full report from the user needs investigation can be found on the MPOWER website (http://www.mpower-project.eu) and a summary is presented in [10].

Fig. 4 shows use cases and associated features for calendar events, which during the pilot development became central element in the information-sharing pilot. The diagram shows that:

- The "Add calendar event" use case is the source for four features.
- It includes another use case – "remind about calendar event".
- Six actors can be involved in the use case, including one system, the "Calendar System".

Modelling use cases with relationships (dependency and association) to scenario and features allows for a quick and informative overview of the properties of the system under design. The UML tool provides functionality for presenting a relationship matrix and navigate along these relationships. For the designer of a system this means that one can easily check if all use cases are based on a real user scenario and if all features are related to a scenario (via use case).

Fig. 5 shows central use cases related to alarming. The features are derived from the use cases and the user scenario

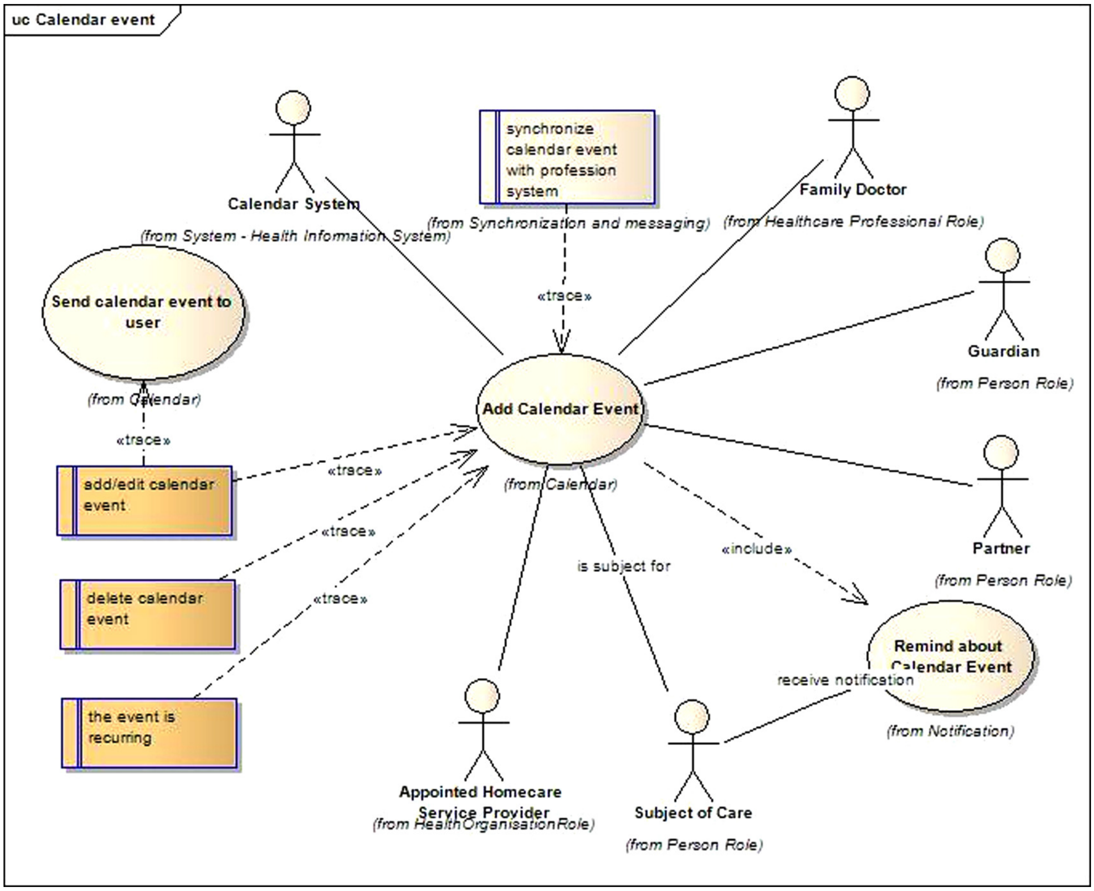

Fig. 4 – Example of the information-sharing use cases and features.

descriptions that are properties of the use case elements. The diagram indicates that reminder functionality will require careful design and implementation. The alarming use cases and features were used during the pilot development for the sensor-centric pilot.

3.2. *Model-driven toolchain development*

The MPOWER toolchain consists of a set of standard development tools (e.g. UML modelling tool and Java IDE) and a set of customizations (UML profiles and transformation scripts) that are used by the developers in all development phases to produce the software. The functionality provided by the MPOWER toolchain is:

- Full support for the Object Management Group's Model Driven Architecture® for developing software services,
  - Computation Independent Models (CIM): using Sparx Enterprise Architect, the domain experts and system architects can model the business (domain specific) aspects of the system. This is typically done using UML Use Case and Activity diagrams. The toolchain also supports creating models using the Business Process Modelling Notation (BPMN).
  - Platform Independent Models (PIM): also using Sparx Enterprise Architect, system architects and developers will identify and design software service models in UML. Using the UML extensions provided by the toolchain, the architects and developers could add domain specific details to the models, making the models more specific and suitable for more targeted model transformation and code generation.
  - Platform Specific Models (PSM): the toolchain supports generation of Web Service Description Language (WSDL) files from the service design in two ways: either directly to WSDL code using Eclipse with the MOFScript plug-in, or via a WSDL model in Enterprise Architect. The results are a WSDL file that has a complete description of the services' semantics (messages and types), behaviour (operations) and protocols (ports and bindings). The WSDL file is used by Java IDEs such as Netbeans to generate most of the code needed to implement an executable Web Service component.
- Provides domain specific modelling support for the assistive care domain: using UML Profiles as a mechanism for defining a Domain Specific Modelling Language (DSML), the MPOWER toolchain provides SOA and Homecare specific modelling elements. These elements, called stereotypes, tagged values and constraints, enable the system architects and developers to model services using a common semantic library (e.g. stereotyping patients with «SubjectOfCare»).
- Provides model transformation and code generation functionality: the toolchain provides two mechanisms for generating WSDL code: (1) using Enterprise Architect model

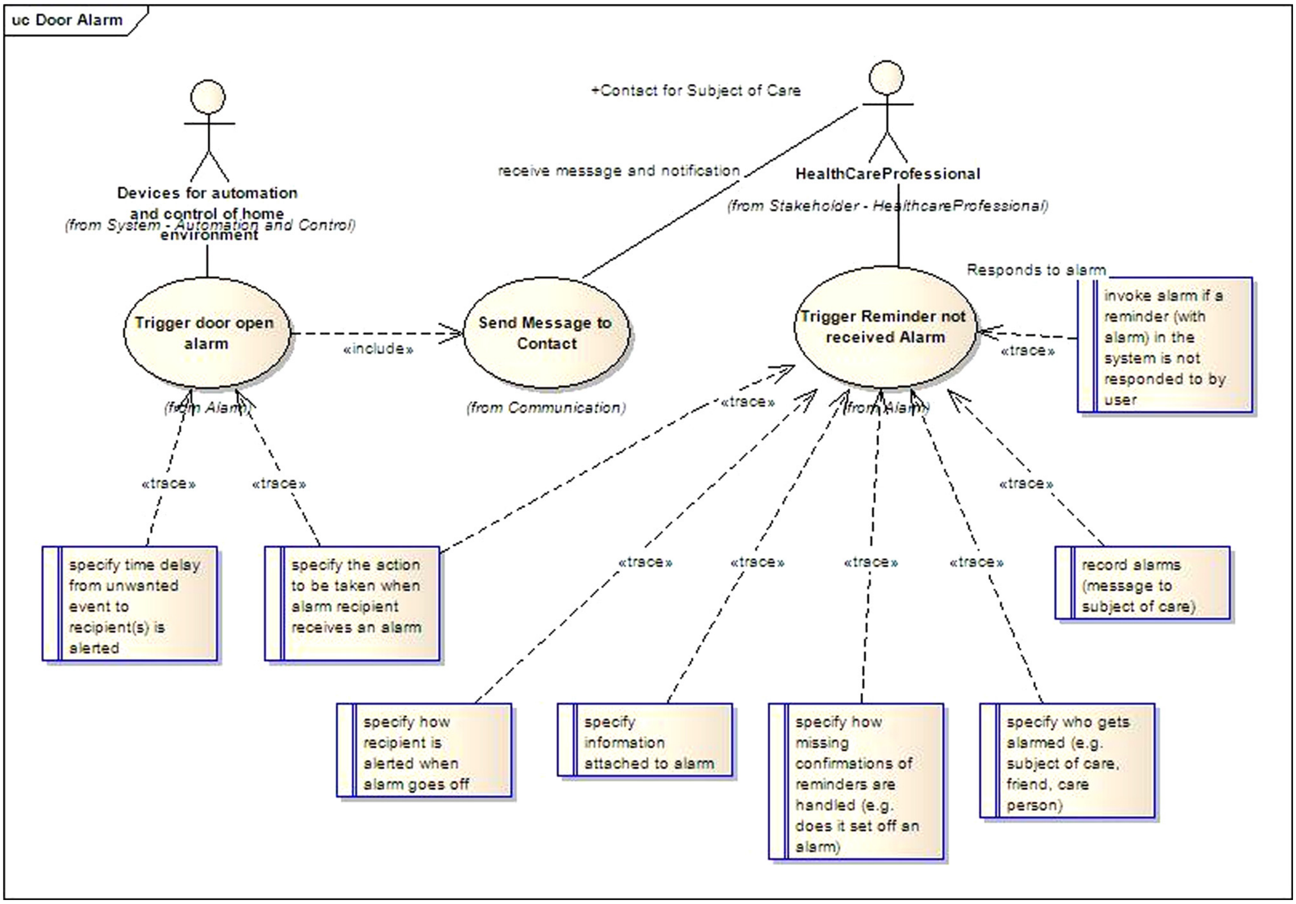

Fig. 5 – Use cases and features related to alarming in sensor-intensive pilot system.

transformation for WSDL and then WSDL code generation, or (2) exporting the service design models to XMI to be used by Eclipse and MOFScript to flexibly generate high quality WSDL code. The MOFScript can be extended to generate more code such as hibernate mappings.

Fig. 6 shows a schematic of the software engineering process proposed with the MPOWER Toolchain.

3.3. *Services modelling and development*

This section presents the results from the service modelling and development activity. Firstly the results of the design methodology applied are presented. Secondly, we present an overview of the complete set of services designed and implemented in the MPOWER platform. The descriptions of the services are organized in a set of categories of related services, each of which are described in a sub-section.

As described in SOA4HL7 methodology [7] and the task-centric service design described by Erl [8] (page 539) a service should represent a logical functionality in the domain. User scenarios were used to develop domain-specific system features. The features were grouped into logical categories making a rigorous knowledge base from which concrete services could be defined.

Service definitions can include one or more interfaces. In general, different interaction styles can be split into different interfaces e.g. Query (read-only) vs. Manage vs. Notification (subscription based). In case there is only one interface it can be called the same as a service. The features will also give some input to the process of defining the interfaces and their operations. Experience from the development of the MPOWER services indicate that the tools support works best if there is only one interface for each service.

The service definitions are described using UML class diagrams. As shown in Fig. 7, the service is defined by a UML class representing the service provider, a UML interface describing each interface and its operations, and with the interface associated with a port of the service provider.

Each of the services described in this section was designed based on the use cases and features developed in the project, as described in Section 2.3.3. Some of the services were implemented using the full tool-chain, while for others the WSDL description and implementation was done by manually based on the service model because the full toolchain was not available during their development.

With the exception of some of the services described under Section 3.3.3, all the services were realized as SOAP based web services available over HTTP. The services were developed in Java based, with their interfaces described in WSDL (either generated or hand-coded).

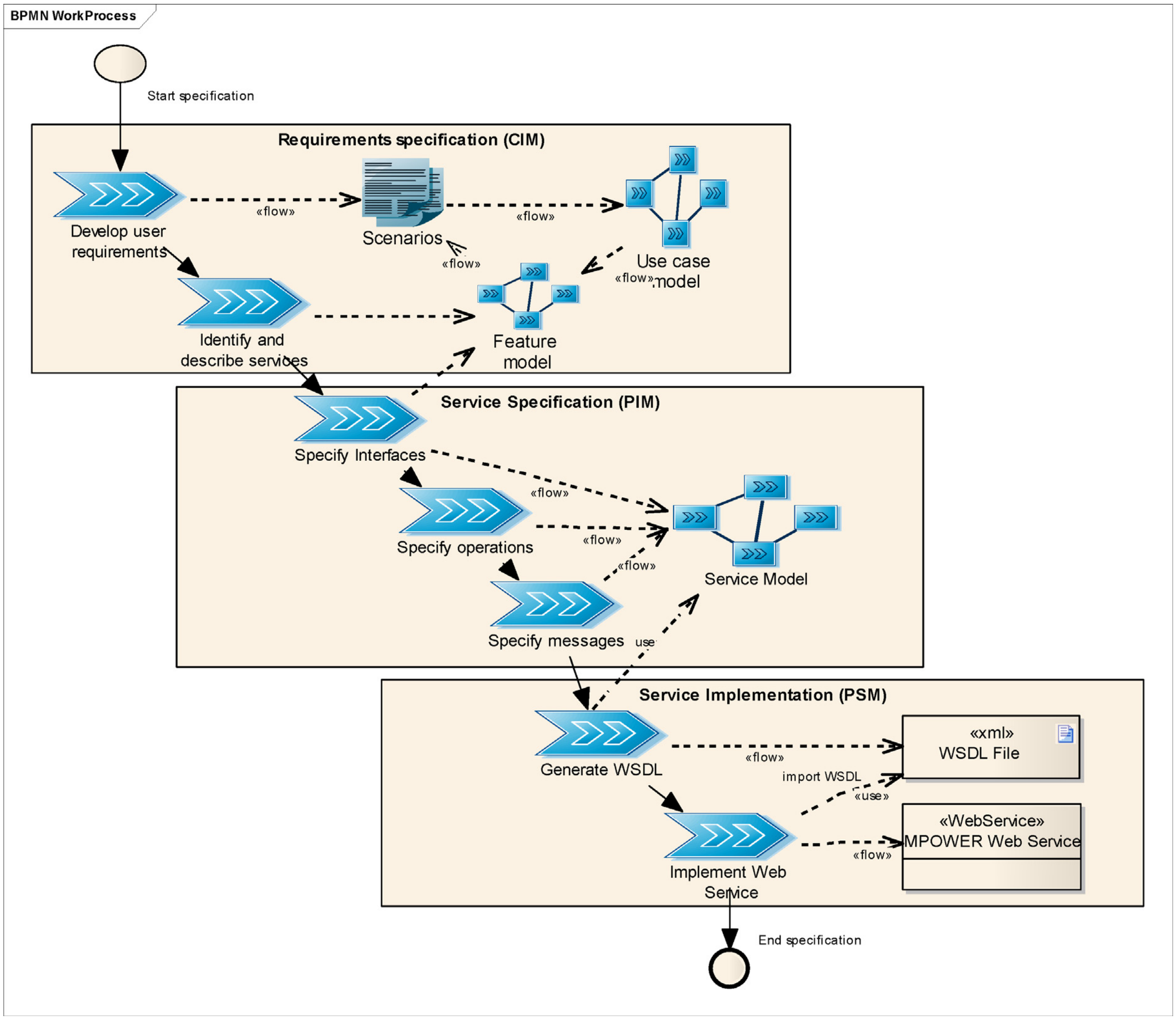

Fig. 6 – The MPOWER toolchain engineering process.

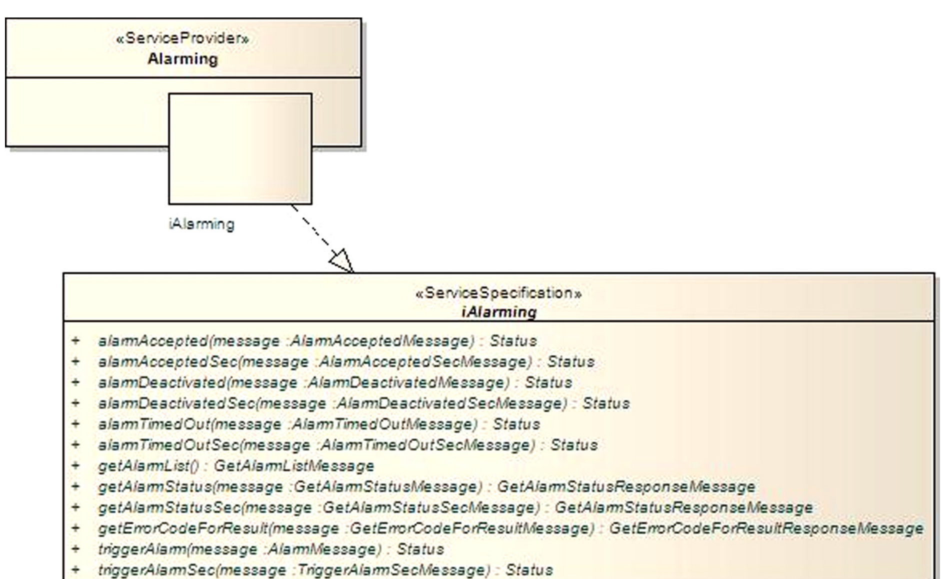

Fig. 7 – The service model with UML stereotypes.

The final set of services is described in the next section.

3.3.1. *Medical and social information services*

The medical and social middleware services design represents the cornerstone of the Subject of Care individual plan support provided by MPOWER. Among the provided services are the Calendar and Medication management features, which are central in this context. Most of the medical and social services provided by the platform uses input an output messages, which conform to the HL7 version 3 standard. Health Level Seven (HL7) is as American National Standards Institute (ANSI) accredited Standards Developing Organizations operating in the healthcare arena, and with its domain of clinical and administrative data it maps well to the MPOWER domain scope.

The following services are offered:

- *Medication management* provides functions for managing and retrieving medication information for a subject of care. HL7 supported.
- *Calendar management* provides functions for scheduling all kinds of social and medical activities for subjects of care, caregivers, family, and friends. It includes operations for booking, confirming, rejecting and cancelling activities, as well as for finding and retrieving existing activities and finding available slots. HL7 supported.
- *Message board* provides functions allowing caregivers or family members to exchange messages that could contain any patient's related information that needs to be shared. HL7 supported.
- *Reminder* provides set of operations for creating and managing various types of reminders for upcoming medical and social activities and events, and can be used for people with cognitive disorders to enable options of reminding them about upcoming social events, medical treatments or other scheduled activities. HL7 supported.
- *Patient management* provides information about the patients through a common and standardized interface, and enables the developers to add, update and delete stakeholders from the system. In addition, the interface allows for querying for relationships between stakeholders such as patient-provider relationships or patient-relative relationships. HL7 not supported.

3.3.2. *Communication services*

The communication services support different kinds of communication between users and systems, including alarm handling, sending messages and notifications, and calls with voice and video.

The following component and services are offered:

- *Alarming* service is designed to manage alarms in the MPOWER system, and provides operations to trigger new alarms, accepting and deactivating alarms, as well as querying for current alarms and their status.
- *Notification* mechanism consists of two main parts: (a) the MPOWER notification web application with the Notification Service and (b) a virtual service represented by a business service. To receive the notification message from the MPOWER environment the application has to subscribe to the notification mechanism using the notification service. A BPEL is sending notifications to of the alarming statuses to the alarming service as well as all subscribed applications. Therefore the BPEL is the business controlling to handle the notification and alarming procedure of different applications and different alarming statuses.
- *External notification* implements web-methods to interface external service for sending emails and SMS messages, by connecting to email servers and a public HTTP2SMS service.
- *Voice-/video communication* services provide the possibility to call other users (audio live stream) and watch them (IP-camera live stream). The service includes methods for managing incoming, outgoing and active calls, and for managing accounts and contacts. Existing SIP server technology is used as part of the realization of this service. The service is web service based and therefore independent from any GUIs. As the service is using SIP it is possible to establish also voice connections to external phones from the application.

3.3.3. *Sensor framework and services*

The sensor services provide functionality for configuring (add, remove, adjust) devices and retrieving sensor information. The services expose both management mechanism and data access through an easy to use and standardized interface. The idea is to have components in the middleware which are enabling a plug-and-play of sensors independent from the protocol and vendor.

The following component and services are offered:

- Therefore the Frame Sensor Adapter (FSA) framework service has been implemented which provides unified access to sensors and actuators that use different communication channels and different data formats through the use of the ISO/IEEE 11073 [31–33] data protocol standard. The main goal of the FSA model is to hide complexity and details inherent to the sensor communication to the upper middleware services. FSA receives heterogeneous information from many kinds of devices and transforms it to a standardized message format. The messages are sent through the Enterprise Service Bus (ESB), and component and services can subscribe to this bus in order to capture the data when they are published at the bus. All sensors to the FSA have been connected over a KNX[5] network.
- *FSA-J2ME* is an adapted version of the FSA component for use of the FSA on a mobile device. This component is written following the J2ME specifications.
- *Door control* provides a service for accessing and operating a door lock. The service contains operations for opening the door lock for an authorized person, requesting the current status of the lock (open/closed), and retrieving a log of people that have opened the door.
- *Door control management* provides functionality for manage access to different areas of a house. The service contains operations for adding and removing access to doors for users, for specifying the time intervals in which access is allowed, and for associating RFID-tags with users.

[5] KNX website: http://www.knx.org, accessed August 5, 2010.

- *Camera Access* is used for controlling and providing access to a camera stream via a HTTP network protocol. This service is in charge of controlling access to different cameras that can be registered in the MPOWER environment. The service deals with the task of connecting with the camera and redirecting the information coming from the camera to the client that asks for the camera stream.
- *Device manager* is used to register several types of devices that are to be installed in a system, including device types and protocols. As an example a medical SpO2 Sensor has been connected to the system using Bluetooth, providing the heart rate information of a user to the system. As another example a thermometer device can be connected using KNX standard [34].

3.3.4. *Interoperability services*

Interoperability services are providing the interface for external systems. This is important as medical and social relevant data have to be transferred to legacy systems, etc. The interoperability of the platform and the integration in overall environment frameworks must be guaranteed. To this end, international standards with open interfaces have been applied.

The following services are offered (see also Fig. 8):

- The *Export to Google Health* service is providing all medical and social relevant data in a standardized data record format of Continuity of Care Record (CCR) [35]. The CCR standard is a health record standard specification developed jointly by ASTM International, the Massachusetts Medical Society (MMS), the HIMSS, the American Academy of Family Physicians (AAFP), the American Academy of Pediatrics (AAP) and other health informatics vendors. As the CCR is not common in Europe but is the standardized format that Google Health is able to read, there is also the possibility of a Continuity of Care Document (CCD) export. CCD is the combined and agreed clinical document architecture of ASTMs CCR and HL7s CDA data format. Google Health is a service offered by Google where individual users are allowed to create their own profile and manage their own health information. The overall idea is to allow centralized management of personal health information. Google Health also provides application programming interfaces (API) to allow third party applications to retrieve profile data, and to add new elements to personal profiles.
  Through an XSLT transformation the data can also be simply transferred to a HTML view of the data.
- The *Medication Plan Synchronizer Service* offers with the functionality described above the possibility to synchronize data of the internal MPOWER system with data records of any legacy system (hospital information system or nursing system) so far they are also providing the data information in the internationally standardized format of CCD or CCR. The purpose of using these standards is to create flexible documents that contain the most relevant and timely core health information about a patient and transfer and synchronize them with external data about the patient.
- The *iCal and Google Calendar Export* service provides the possibility to export and synchronize the calendar information from the internal Calendar Management service with external calendar systems. Therefore the export in the common iCal format is used which can be imported with a lot of calendar tools like Microsoft Outlook, Apples iCal, Mozilla Sunbird, Thunderbird and many more.
- With the *Calendar Synchronizer* a subscription to the iCal format is possible. With such a subscription the mentioned applications can manage to get the latest updates of events and reminders by downloading and parsing the iCal file intervallic. Therefore all scheduling information of generic appointments for healthcare services and/or resource slots as well as appointments with relatives, friends and other members of patient social network as well as for any kind of activity including social activities or medication activities, can be synchronized and also subscribed by a lot of individual calendars. This technique allows the user to easily link dates from one calendar application to another.
  The iCal format is also used by the Google Calendar.
- The UDDI *Service Registry* service provides a platform independent way of describing and discovering Web services and Web service providers. The UDDI data structures provide a framework for the description of basic service information, and an extensible mechanism to specify detailed service access information using any standard description language. Many such languages exist in specific industry domains and at different levels of the protocol stack.

3.3.5. *Security services*

The objective of the MPOWER security middleware is to ensure sufficient protection (i.e. security level) for any of the MPOWER enabled services when they are used. This implies that security middleware is orthogonal to the other services in the way that it is implicit a part of each service, ensuring a satisfying security level of any combination of services in the MPOWER platform.

The authentication mechanism in MPOWER is included to allow only legitimate system users access to MPOWER resources. Before access to the system can be granted, users have to prove their identity. This procedure is performed through an authentication process. Authorization in MPOWER is based on a Role-Based Access Control (RBAC) scheme. In this scheme, a set of permissions is associated with each defined role, and users get permissions indirectly through the roles they are assigned. Roles in healthcare systems are often clearly defined, and the number of roles is often significantly lower than the number of users.

The following services are offered:

- *Authorization* service determines what operations and which data an authenticated user can access, allowing access to resources only to legitimate, authorized users.
- *Authentication* service verifies a user's credentials and allows access to the system only to users with valid credentials.
- *Token management* service is used by the authentication and authorization services to manage the login sessions.
- *Role management* service enables the Administrator to manage the roles of the system. The Administrator may add/delete roles, assign users to roles, get the role information, and get the user's assigned to a role.

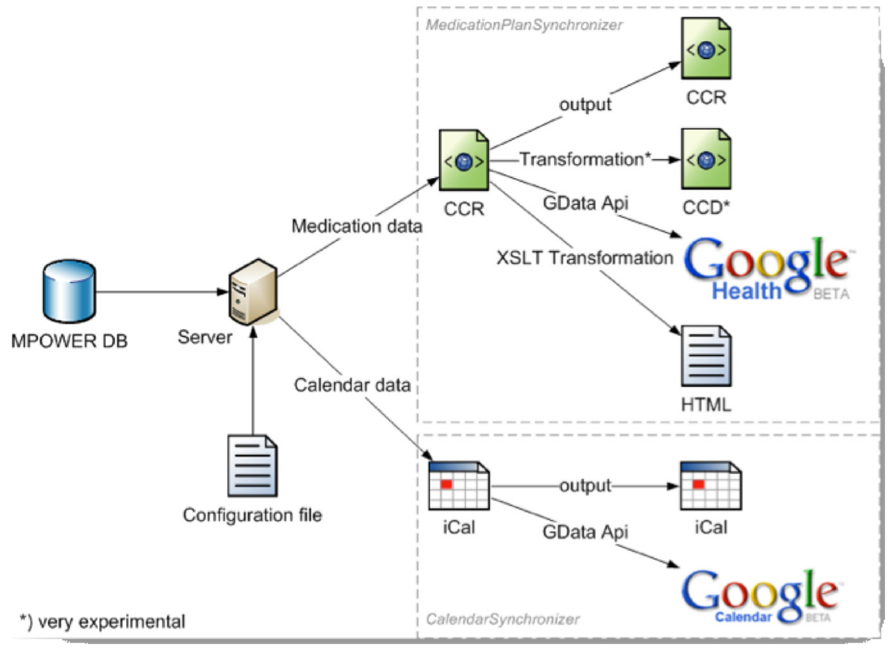

Fig. 8 – MPOWER data export services.

- *Access management* service manages the permissions and access profiles associated with the access control system.
- *User management*: The User management service enables the Administrator to manage the users of the system. The Administrator may: add/delete users, update the user's roles, and get the user information.

3.4. *Pilot systems*

3.4.1. *Information-sharing pilot*

The goal of the information-sharing system is to organize and share information among the informal and formal caregivers and the subjects of care. The shared information pertains to the subject's care process and his or hers everyday living. The information-sharing system is designed to help the people suffering from the initial stages of the dementia to remain in their homes for as long as possible.

The concept of the information-sharing system is presented in the figure below (Fig. 9). The information-sharing system creates a care space within which the information on the care processes is shared among the caregivers and the subject of care. For each subject of care a single care space is created that relates the subject of care and all the caregivers that care for that particular subject. Furthermore, that care space contains all the information that is relevant for provision of quality care for the subject of care. For instance, an elderly person named Jane might have a care space. All the formal and informal caregivers of Jane, such as doctors, nurses, visiting nurses and family members might be connected to that care space. Jane's caregivers and Jane share the information that is important for the care provisioning to the Jane. For instance, they share calendar information on the everyday tasks of the subject of care, can send and share messages related to the Jane, or initiate calls among the Jane's caregivers and Jane. The MPOWER services are plugged in into the care space in order to provide the necessary support for information sharing. Furthermore, the external systems are also related to the care space in order to provide the communication facilities that are external to the MPOWER services. For instance, external systems are clinical systems for drug management that provide the information on medications used by subjects of care and telecommunication networks that provide the SMS and voice call capabilities.

The information-sharing system has been designed in two variations. The first variation is developed as a web-based application and was tested by seven respondents in Trondheim region of Norway. The second variation of the system is a slightly improved version developed as a thick client application and tested by two respondents in the Grimstad region

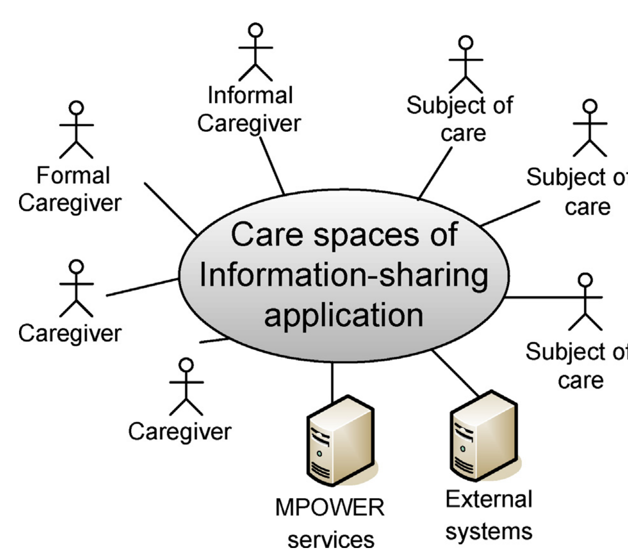

Fig. 9 – The conceptual organization of the information-sharing system.

Fig. 10 – Screenshot from information-sharing pilot systems, the web-based variant.

of Norway. The two variations differentiate in the stability, applied technology and appearance but essentially provide the same functionality with only slight differences.

3.4.1.1. *Features*. The information-sharing application runs as a standard PC application and connects to the care space provided by the MPOWER services through a mobile or DSL broadband Internet connection. For interaction with elderly subjects' of care a 17″ touch screen was installed in an easily accessible area in the home of the elderly.

Figs. 10 and 11 present the layout of the application as seen by the elderly users. Fig. 10 represents the Web-based variation of the application, while Fig. 11 represents the thick client variation of the application. The graphical layout of both variations is designed inline with the input from the staff from home-based services in collaboration with the Norwegian Centre for Dementia Research. Both the subjects of care and the caregivers see the same layout. However, the caregiver interface has a few extra features that are explained below.

The *layout* uses clear big fonts and large contrast between colours wherever possible to enhance the readability to the elderly. The interface provides the time and day information in various formats as it was judged that the people suffering from cognitive decline may have different tendencies toward comprehension of time represented in digital vs. analogue clock. The time and day information is located in the left, right and upper part of the screen. The central section of the screen is reserved for presentation of the information pertaining to the care process. The lower part of the screen is reserved for large buttons that enable access to the various functionalities provided by the application.

The possibility to use some of the features of the system depends on the level of the privileges the particular user has. Thus, the system also provides the *security features* that enables secure login to the system, recognition of the user, and potential to control access to some features based on the privileges of the user.

The most important feature of the system is the *basic calendar feature* that represents the daily activities to the elderly. Example of such information is seen in Fig. 11. This information should be displayed to the elderly throughout the day, thus the applications automatically switch to this view after a period of inactivity from the elderly. The information that is displayed in the calendar is entered by the caregivers of the elderly. Thus, all caregivers of the elderly can see the activities and edit them. For instance, nurse from home-based services, could enter information that he will come to the home of elderly to give him a bath. Besides just displaying the calendar information to the user, the caregiver has the option to assign a reminder to an activity. The reminders are activated automatically and use the visual and audio indication in the interface to indicate to the elderly that some activity is about to start. The *advanced calendar feature* provides the information similar to the basic calendar but not for the current day, but for any provisional day. This advanced calendar feature is accessed by pressing the appropriate button in the interface.

By pressing the Medicine button the *list of medication feature* is activated. The subjects of care and informal caregivers can use this feature to observe the list of medication the elderly should be using. The formal caregiver with sufficient privileges can modify the contents of the list. One important data related to each medication is the instructions for use. Thus, the caregivers and elderly always have access to the proper instructions.

The *messaging feature* enables exchange of the textual messages among the caregivers and the elderly. While submitting

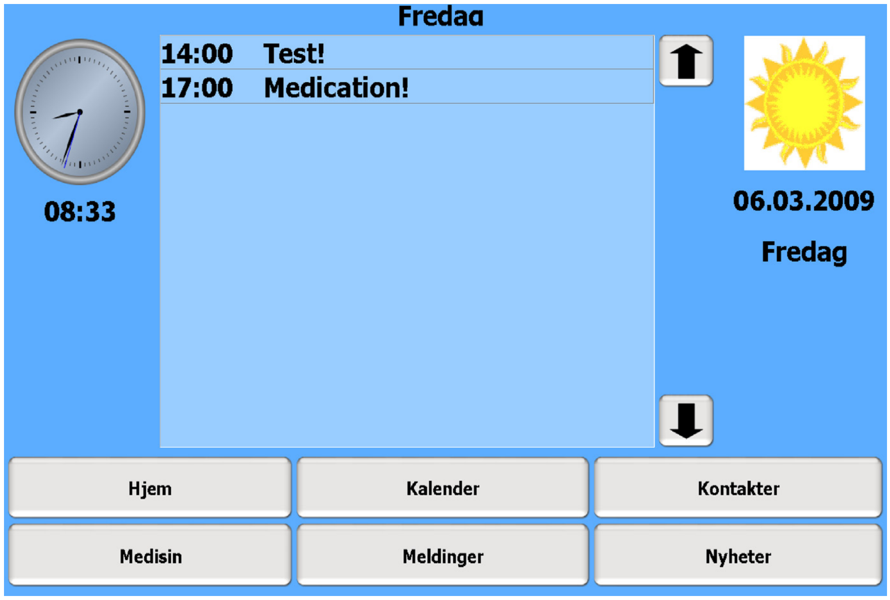

Fig. 11 – Screenshot from the information-sharing pilot system, the thick client variant.

the message the sender can specify which members of the care space should be able to read the message. This feature was frequently used as a message board to exchange information about the state of the elderly among the caregivers.

Contacts and call initiation feature displays the information about the persons related to the care space of the elderly. The participants of the care space are listed in a form that shows their name and picture so that elderly can easily recognize and remember the persons he/she is related to. By pressing the small picture or small button next to the picture of person an SMS message is send to the person. The message indicates that the person should call the elderly. The care space participants that are displayed to the elderly can be edited, in a sense that some participants need not be listed (e.g. doctors), while some others may be moved upper on the list (e.g. alpha daughters).

The elderly has an option to access the *news feed feature*. By accessing the news feed feature the feed for local news is accessed over the Internet and the news are represented in a clear format to the elderly. The exact location of the news feed can be edited.

The thick client version of the application has the internationalization option, which allows for simple translation to different languages. For instance, Fig. 11 represents the English version of the application.

3.4.1.2. Applied MPOWER services. Table 5 summarizes the MPOWER services used in the creation of the information-sharing applications. The first column of the table designates the feature of the application, while the second column designates the MPOWER services that were applied in provisioning of the feature.

3.4.2. Sensor-centric pilot

The goal of the sensor-centric system is creation of a smart-home environment adjusted for application in residential houses for elderly where it should simplify the care process to the personnel and increase the quality of care for the elderly residents. The sensor-centric system is intended for people having cognitive decline and as such are not capable of living independently and thus require continuous surveillance and help. The system was tested in a Polish nursing home.

Table 5 – The MPOWER services applied in the information-sharing system.

Feature	Applied MPOWER services
Security features	To execute the Security feature the system uses the *Authentication*, *Authorization*, *Token management* and *Access management services* of the MPOWER platform.
Basic and advanced calendar feature	The web-based variation of the application uses the *Calendar management service* based on the *HL7 standard*, while the thick client variation of the application relies on the *Calendar management service* based on *the iCal standard*. For the functionalities related to the reminders, *Reminder service* is used.
Medication list feature	The management of the list of medications is done using the *Medication management* service.
Messaging feature	The management of messages is done by application of the *Message Board service*.
Contact list and call initiation feature	The information represented in the contact list is managed using the *Patient management service* of the MPOWER platform. The SMS messages are activated by using the *External messaging* service.
News feed feature	The news feed functionality was developed without using the MPOWER services. It is implemented as an RSS feed reader built into the information-sharing application.

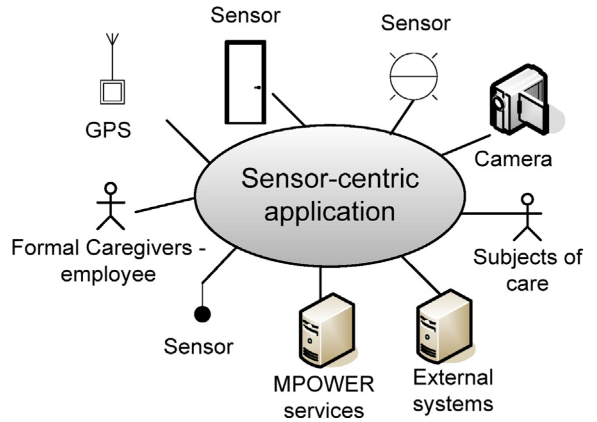

Fig. 12 – The conceptual organization of the sensor-centric system.

The concept of the sensor-centric application is presented in the figure above (Fig. 12). The sensor-centric application collects the information about the state of the nursing home using the set of sensors installed in different rooms of the house. Primarily, the sensors are installed in the rooms of the elderly residents where they are used to check environment variables of the room, such as room temperature and if the doors and windows are opened or closed. The information collected by the sensors is presented to the formal caregivers – personnel of the nursing home through a nursing cockpit that they can use to monitor the state of multiple subjects of care and their rooms simultaneously. Based on the readings displayed in the nursing cockpit they can initiate different actions in order to help the elderly residents do they everyday activities.

Besides providing the observation and monitoring station, the nursing cockpit also serves as a means of communication between the personnel and elderly residents. Specifically, rooms of elderly are equipped with PCs and monitors through which the audio, video and messaging communication with the caregivers can be done. The sensor-centric application provides these functionalities by using the MPOWER services and communicating with external systems for additional support.

The sensor-centric system has been installed in an elderly house in Lubomierz in Poland and has been tested by the elderly house personnel and six residents from the April 2009 till the completion of the project activities.

3.4.2.1. *Features*. The sensor-centric system runs as a web-based application that has a nursing cockpit for caregivers and a simple user interface for the elderly. The caregivers use the system through a standard PC installed at a reception office, while the elderly residents have a PC installed in their rooms. Additionally, some of the features of the nursing cockpit view are also available through web interface customized for application on smart phones. Thus, personnel had an option to access relevant information anywhere and anytime.

Figs. 13 and 14 present the layout of the sensor-centric application. The (a) part of the figure represents the application as seen by the elderly residents of the house, while the (b) part represents the nursing cockpit that is displayed to the personnel of the elderly house. As can be seen in the figure, the interface for the elderly is very simple featuring only three options. On the other hand the nursing cockpit displays a myriad of options that are enabled for the house personnel.

The system enables *environment information collection* through which the personnel of the elderly house can access and read the information collected by the sensors in the house. Primary this feature is used the check the state of the doors and windows in the rooms of the elderly (opened/closed) and the room temperature of the elderly rooms.

Besides collecting information about the state of the rooms in the house, the system enables *collection of the elderly health state information*. For instance, if an elderly has an Oximeter

Fig. 13 – Screenshot from sensor-centric pilot system, the elderly resident's interface.

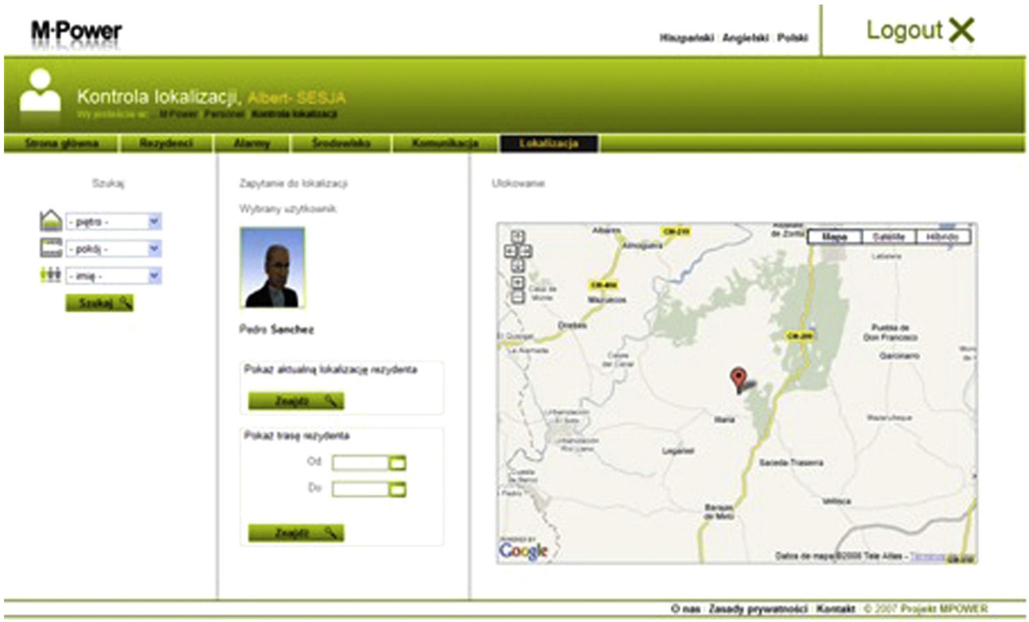

Fig. 14 – Screenshot from sensor-centric pilot system, the nursing cockpit interface.

sensor for measurement of the heart rate and the oxygen saturation, they can access this information through the nursing cockpit interface.

Elderly tracking feature is enabled by giving the elderly a GPS enables device that periodically reports the position of the elderly. This feature is typically used for elderly taking a walk in the forests nearby the elderly house, since there is a reasonable chance that the elderly will forget the way back to the elderly house.

Alarming feature enables elderly residents to activate an alarm in the case they do not feel well or need help in general. The alarming feature can be activated through a web interface for the elderly or through a mobile phone given to the elderly. The alarming feature contains a fine tuned alarm handling process that forces the elderly house personnel to follow standard alarm handling procedure. In the scope of the alarm handling process, elderly house personnel are alarmed using web interfaces and SMS messages repeatedly until they signoff the alarm as handled.

Audio/video communication feature enables establishment of audio and video communication between the personnel and residents of the house using the web interfaces of sensor-centric application. The communication can be initiated either by the personnel or the residents. Typically, audio/video communication is used as a first step in handling the alarm.

Video surveillance feature enables personnel of the elderly house to observe some rooms of the elderly house for potential problems.

Messaging feature enables sending of the messages between the residents of the house and the personnel. Typically, this feature is used to remind the residents of the house that some event is about to happen, e.g. lunch is about to start.

3.4.2.2. *Applied MPOWER services*. Table 6 summarizes the MPOWER services used in the creation of the sensor-centric system. The first column of the table designates the feature of the system, while the second column designates the MPOWER services that were applied in provisioning of the feature.

4. Evaluation results

This section presents results from the evaluations. The first sub-section presents the results from evaluation of the SOA approach on the developers in the project, which is main focus in this paper. The second sub-section gives a summary of the evaluation of the pilot applications.

4.1. *Evaluation of toolchain and SOA approach*

The model-driven development toolchain was evaluated on the 16 developers in the MPOWER project. The evaluation is presented in by Walderhaug et al. [13]. They conclude that:

> *"the respondents indicate that MDSD tools must be perceived useful and should be easy to use. Tool performance does not have a direct effect on MDSD use, although business analysis, traceability and code generation were found useful. It is especially important that MDSD tools are stable and provide complete and correct artefacts"* [13].

On the same cohort as the toolchain evaluation, the SOA approach was addressed with focus on the usefulness of service orientation. The results are analysed statistically using basic statistics: mean score/standard deviation and factor reliability (Cronbach's alpha). A more extensive analysis is not given due to the limited statistical material ($N = 16$).

Tables 7–11 show the scores for each item in the factors. All factors except the future use intention (FUI) are strong ($\alpha > 0.7$). With respect to perceived usefulness, the questions addressing usefulness and job quality score particular high (maximum is 5). All questions get a mean value higher than neutral (3

Table 6 – The MPOWER services applied in the sensor-centric system.

Feature	Applied MPOWER services
Security features	To execute the Security feature the system uses the *Authentication*, *Authorization*, *Token management* and *Access management services* of the MPOWER platform.
Environment information collection feature	Environment information collection feature is designed using a *FSA component* that collects the information and *FSA service* that is used to access to the collected information. The FSA component is upgraded with the drivers to access the temperature and window sensors.
Elderly health state information collection	Elderly health state information collection feature is designed using a *FSA component* that collects the information and *FSA service* that is used to access to the collected information. The FSA component is upgraded with the drivers to access the oximeter sensor.
Elderly tracking feature	Elderly tracking feature is designed using a *FSA-J2ME component* that collects the information about the geographical position from GPS sensor and *FSA service* that is used to access to the collected information. The FSA-J2ME component is upgraded with the drivers to access the GPS sensor.
Alarming feature	The alarming feature is designed by using of the *alarming service*, *notification service*, *complex BPEL based management of alarming process* and *External messaging service*.
Audio/video communication feature	The *Voice/Video Communication* service is applied for the management of the audio and video streams required for the audio/video communication.
Video surveillance feature	The video surveillance feature is enabled by management of the camera through *Camera Access* service.
Messaging feature	The management of messages is done by application of the *Message Board service*.

is “neither agree nor disagree”). No developers said that they strongly disagreed (value = 1) with the PU questions.

The questions addressing the perceived ease of use (PEOU) had a high internal consistency with alpha = 0.906. As for PU, all questions got better mean score than average. However, the standard deviation is quite high.

Perceived compatibility in this paper addresses how suitable SOA is for application development in the AAL domain. Again, the internal consistency is high and all questions get better than average score.

When asked about how their future intentions to use SOA in their work the results show that SOA will be used, but not in all types of development.

In addition the factors above, a set of claims were put forward and the developers rated the claims on a Likert-5 scale with scores strongly disagree (1) to strongly agree (5). As these questions are not a factor per se, internal consistency was not checked.

4.2. *Results from evaluation on end users*

This section presents some brief excerpts from project-internal report on the evaluation with end users that was undertaken on the two fully functional pilot systems. A more extensive report on the evaluation of the information-sharing pilot is available in [9], while the evaluation of the sensor-centric pilot is a topic for future publications.

4.2.1. *Conclusions from information sharing pilot system evaluation*

The evaluation is positive and emphasizes the importance of simple user interface. POCA and PPOCA are the internal terms used to denote Proof-Of-Concept-Application and Polish-Proof-Of-Concept Application in the project period. In this article they are named Information-sharing pilot and Sensor-centric pilot..

> *“There are several examples of respondents have benefited from the POCA. People with memory problems or other cognitive impairment have benefited too. The best thing with the calendar is to read what day and time of day it is. It is always correct. It seems that the simpler the digital calendar is, the easier it is to read and understand. We understood the tendency that the respondents were afraid to do something wrong, or make the page disappear if the touched anything. There seems not to be a need for many pages and many navigation buttons. Rather, one start page displaying day, time of day, date, clock and all appointments and messages for the current day seems to be best.”*

Moreover, evaluation results from the caregivers conclude that:

> *“. . .family carers and professional carers can by using the message board in the POCA, have the opportunity to improve the collaboration between them. This is a service that family carers highly appreciate, but it requires that all are able to add and delete messages.”*

Table 7 – Descriptive results for factor perceived usefulness.

Perceived usefulness (PU)	Cronbach’s alpha = 0.768				
	N	Minimum	Maximum	Mean	Std. deviation
I find Service Oriented Architectures useful in my job	16	3	5	4.31	.602
Using Service Oriented Architectures improves my job performance	16	2	5	3.69	.873
Using Service Oriented Architectures increases my productivity	16	3	5	3.44	.629
Using Service Oriented Architectures enhances the quality of my job	16	3	5	3.81	.544
Using Service Oriented Architectures makes it easier to do my job	16	2	5	3.50	.966

Table 8 – Descriptive results for factor perceived ease of use.

Perceived ease of use (PEOU)	Cronbach's alpha = 0.906				
	N	Minimum	Maximum	Mean	Std. deviation
Learning to use Service Oriented Architectures was easy for me	16	2	5	3.63	1.147
I think Service Oriented Architectures is clear and understandable	16	2	5	3.75	1.183
Using Service Oriented Architectures does not require a lot of mental effort	16	2	5	3.25	.856
I find Service Oriented Architectures easy to use	16	2	5	3.38	.957
Service Oriented Architectures is not cumbersome to use	16	2	5	3.38	.957
Using Service Oriented Architectures does not take too much time from my normal duties	16	2	5	3.13	.885

Table 9 – Descriptive results for perceive compatibility.

Perceived compatibility (PC)	Cronbach's alpha = 0.846				
	N	Minimum	Maximum	Mean	Std. deviation
Service Oriented Architectures is compatible with the way I develop Software	16	2	5	3.63	.885
Using Service Oriented Architectures is compatible with all aspects of my work	16	2	5	3.25	.856
Using Service Oriented Architectures fits well with the way I work	16	2	5	3.69	.793
Service Oriented Architectures is compatible with the way we organize our work	16	2	5	3.56	1.031

Table 10 – Descriptive results for future use intentions.

Future use intentions (FUI)	Cronbach's alpha = 0.846				
	N	Minimum	Maximum	Mean	Std. deviation
I intend do increase my use of Service Oriented Architectures for work in the future	16	3	5	4.00	.516
I intend to use Service Oriented Architectures in the future for my work	16	3	5	4.06	.680
Given a choice, I would prefer not to use Service Oriented Architectures in any future work	16	1	5	2.00	1.033
I would like to use Service Oriented Architectures in the future	16	3	5	4.25	.683

Table 11 – Descriptive results for SOA claims.

Claim	N	Minimum	Maximum	Mean	S.D.
In SOA, often, services are interoperable; they can be used by systems implemented on various platforms.	16	3	5	4.38	.619
SOA leads to simpler software design and implementation by decomposing complex problems into smaller more manageable ones.	16	2	5	3.94	.854
SOA enables improved adaptability to changing business requirements.	16	2	5	3.94	.929
SOA enables cost benefits consequent to the above benefits.	16	3	5	3.75	.775
SOA enables the business logic of IT services to reflect closer the core enterprise business processes.	16	2	5	3.81	.911
SOA enables us to reuse existing IT systems (legacy systems) instead of implementing new ones.	16	3	5	3.69	.704

4.2.2. Conclusion from the sensor-centric pilot system evaluation

The report concludes:

> *"Due to relatively low computer literacy within the users group, training activities were to be repeated nearly on all occasions when the project team was visiting the Residential House. The interviews carried out with the users belonging to the group of residents showed high acceptance of the application implementation in the environment of the Residential House. It became also obvious that the residents are particularly focused on these features of PPOCA which bring the improvement of their safety. In some cases, the communication with family members living far from the Residential House or even abroad was highly emphasized."*

Furthermore, more technical aspects of the evaluation revealed that: "*The features of the PPOCA which required better acquaintance with computer (sending text messages, triggering alarms from the touch screen) were not accepted in the same degree as features assuming passive role of the residents (location tracking, sensor for environment control). However, it was also observed that with every visit during the evaluation of the PPOCA the awareness and acceptance of specific features was improving.*"

5. Discussion

This section discusses the results achieved in terms of the research questions defined in Section 2. Following the discussion of the main questions addressed, a brief discussion on the implications for other system development initiatives in terms of the results presented herein is presented. The final subsection describes the open source project initiated from the project where all service designs and implementations are shared.

5.1. Developers' use of model-driven development and SOA in the AAL domain

One of the central questions addressed in this paper is how do the developers perceive the use of model-driven development and SOA for application development in the AAL domain. The question was investigated through an extensive design, development and evaluation approach involving 16 developers from five European countries in a two-year period.

With respect to using model-driven development in web services design and implementation, the developers reported that it was useful [13]. However, it is important that the code generated from the models is correct if the full potential of MDD should be achieved. Developers used the same tool and model database for service design as for use case modelling and information modelling. Domain knowledge such as processes, concepts (actor names and use cases) were readily available for developers during service design. As reported in [13], developers with no or limited experience with MDD or UML in general, found it both easy to use and useful in designing service for the AAL domain.

Using formal models such as UML use case, class diagrams and the UML extension for software services [24] enabled the creation of a AAL specific model library that incorporates the knowledge from the user needs investigation, the service designs with traceability to their origin use cases, and the possibility to generate high-quality programming code. Traceability between artefacts in the development process makes it possible to run important analyses on the design models, such as orphan, coverage and change impact analysis. Moreover, traceability is recommended by the FDA as a tool to improve software quality [36], and is described in more detail by Walderhaug et al. [12].

The model database constitutes a knowledge pool for both domain experts and developers. The use case, feature and actor models represent best practice and could therefore be reused in other system designs and even used in standardization processes. This is inline with the ideas of Lenz, Beyer and Kuhn in [37] where they argue for a separation of domain concepts and system implementation: "*in order to cope with domain evolution, modelling of domain concepts should be separated from IT system implementation. IT systems should be implemented by IT experts and medical knowledge should be modelled and maintained by domain experts.*"

Finally, with respect to create a shared understanding of the business (care) processes and the target system's functionality, formal models (e.g., UML models) are useful. Prior studies using UML to document domain knowledge and communicate between professions (e.g., healthcare and IT specialists) and has been successfully applied in several projects [38].

Using MDD for web services development has been evaluated in many settings and is generally deemed useful and productive [39,40]. However, to our knowledge, there are no evaluations that are published on a complete model-driven development process for SOA systems.

The results from the questionnaires are generally on the positive side. With respect to perceived usefulness, the developers clearly find SOA useful (mean 4.31 and range 3–5). The question is general of nature, but it is important to note that the questions addressing job performance and job quality are also positive. The developers also find SOA relatively easy to use and from the questions on job compatibility, it is clear that SOA aids development of AAL applications. This is further confirmed in the factor about future use intentions where the developers clearly report that they will use SOA in future projects.

An important finding is though that the developers do not think of SOA as a "silver bullet" as they report that they will not use SOA for all projects, despite its qualities related to integration and interoperability [22,41,42]. This emphasizes the need for technology evaluations in development projects exceeding demonstration level quality.

5.2. AAL domain specific reusable SOA services

The second of the main research questions in this paper is: what is the set of reusable SOA software services covering reoccurring user needs in the target domain?

An important outcome of the work on the user needs is the knowledge base recorded as a UML model. As described in Section 3.1, this comprehensive model describes actors, use cases, and features derived from these. The model further includes the design of the set of software services in the platform, including full trace information back to the features and use cases motivating their design. Reuse of software and design is not trivial. Krüger states that "*for a software reuse technique to be effective, it must reduce the cognitive distance between the initial concept of a system and its final executable implementation*" [43]. The models presented herein are the results of a formal process to reduce this distance. This approach helped the developers in understanding and remembering how the services were related to the user needs. The model is also of useful in comparing the user domain needs and services from MPOWER to those from other projects.

As presented in Section 3.3 the services developed as part of the MPOWER platform was organized in five categories covering both general and domain specific functionality. The results from using the services in the implementation of the pilot systems showed that the services are useable for system realizations at least in these two cases. Some of the services were used in both pilot systems, while others were used only in one of the systems. The information-centric pilot was also realized in two variants using different client technology, making it more probable the that services used in this pilot will be reusable in other situations.

While the results from using the services in the pilot development are promising, development of additional success pilot systems based on the services would add to the confidence in the reusability of the services.

5.3. *System development with reusable AAL SOA services*

The third of the main question questions in this paper were whether the reusable SOA software services are sufficient to build sustainable AAL systems to be used in real life environments. For a proof-of-concept demonstration of the evaluated and implemented services two functional pilot systems have been implemented via a SOA architecture. The two pilot systems are using common generic services for security and administration, etc. and also specific AAL services. As described, one pilot system is based on information managing services and the other pilot is related to physical domotic sensors.

As shown in Section 3.4 both systems have been designed to demonstrate certain kind of use cases and of course some services, mostly the generic services, are used in both pilot systems. The advantage of the SOA architecture in this case is that the system through rapid development comparatively easy can be tailored into similar systems, concerning the GUI or the interface in general to the user. In case there is a need for services beyond those existing in the platform, adding new services to the platform is easy. The platform documentation and the SOA architecture model [4] describe a structured process for this.

Also, in situations where only parts of the functionality provided by an application is needed (e.g. as provided by one of the two pilot applications), it is easy to reuse selected parts (e.g., services or service compositions) to produce a tailor made application with only selected functionality, for instance with some generic services and only the calendar management.

These two advantages substantiate that a rapid development of applications for different kind of user groups or single end users is possible. In the AAL domain this could be even more important than for other domains because the end users in the domain frequently would benefit from tailoring of the systems to their individual capabilities and needs (e.g. user interface personalization and configuration of sensors and actuators for the home). However, to reach the full potential for personalization and tailoring more high level tools are needed, enabling persons close to the end users to perform some of the tailoring, preferably without the need of professional software developer assistance.

AAL is an application domain with a lot of overlapping subdomains where interoperability so far has been limited. There is however a need for better interoperability between e.g. the eHealth domain, the home entertainment domain, the home automation domain, and the household appliance domain. A SOA structure has the potential to fulfil the needs of integrating different services facing different domains and domain applications and integrate them to one comprehensive application.

5.3.1. *Deploying and operating servers*

During the project we gained experience in setting up and operating applications on real trial sites with elderly users and their carers. One main conclusion from this experience is that setting up and running elderly care application should be done in the same safety/security/performance environment as classical telecom applications.

Specification of the equipment and devices for building the infrastructure of implementation site must be as precise and checked as possible. For instance, we experience that not all computers had the network card that was required for ADSL, and the installation company had to come back – caused stress for the respondent.

During the deployment and operation at end user sites we learned that wireless broadband network connection is yet too unstable to fill the needs of the users in the AAL domain. Even after switching to ADSL unstable network connectivity caused some problems. Resetting a computer after disconnection is sometimes a problem, and to work well for the end users and carers this should happen automatically (e.g. a as it does with mobile phones).

5.4. *Implications for software development in the domain*

In [3], Haux argues that it is necessary to explore new architectural styles to support trans-institutional information sharing. The AAL domain involves many institutions and domains, and is in this respect a relevant domain for exploring new architectural styles. Services for the aging and cognitively disabled do not and will not exist in isolation. The AAL domain consists of a large set of independently developed systems and services in existing environments. On top of the value of these individual systems additional value could be created by exchanging data between these systems or even aggregate the data in a system for analysis and to give proper feedback or alerts to relevant systems or users. These systems and services should be able to communicate with each other not only by exchanging data but also understanding each other's data. This can only be achieved by agreeing on a lot of issues in other words by using standards [44].

Healthcare systems are expected to maintain the continuity of care, shared care and the empowerment of patients in the management process. Many relevant systems and applications exist, that AAL services might need to interoperate with. The services presented herein, are based on SOA by using the web services WSDL and SOAP. Thus it is an interoperability enabler, as the web service front end allows heterogeneous platforms to interoperate (e.g. .NET and Java). This is not enough as the messages, which can be exchanged by applications and systems, can be very different and there can be application policies (such as security), which need special interoperability requirements. To this end, we have designed a reusable interoperability architecture that encapsulates the interoperability components in such a way that changes in these components or the external systems they relate to are hidden from the rest of the MPOWER framework.

The methodology and tools for designing the services, as well as the SOA platform employed in the pilot systems, may be useful to other designers and developers in the AAL domain, or care domain in general. The services and the tools are provided as open source, and the referenced publications document the methodology and lessons learned.

5.5. *Open source*

The advantage of the usage of open source in eHealth and AAL projects concerning the support of standards has been pointed out as a need by the member states of the European Union. There is a need to support actions that cover the development of standards addressing the interoperability of diverse systems and services and to explore in particular the possibilities of open source applications to achieve this objective. In this context, the need for future standards is clearly emphasized to solve interoperability concerns in a way that all stakeholders will benefit from the possible adoption of Open Source reference implementations for care services. In addition, an open and more free access to future and existing e-Health and general interoperability standards in the AAL and eHealth service providing should be recommended, taking inspiration from models such as the World Wide Web Consortium.

The exchange of experience in the use of open standards and open source solutions among health administrations in Member States should be promoted [45]. Success in developing a European e-Health Area and AAL standard conform platform will draw on sharing best practices and experience across the Union, as systems are deployed and organizations redesigned. Open source applications play an important role in achieving interoperability [45].

The MPOWER middleware platform and toolchain and the services and components developed, that constitute the MPOWER platform, has been release under the MIT license, which allows for open-ended business development of products and services not only to the Consortium members but also to the whole ICT and healthcare communities.

The source code of the MPOWER middleware is hosted at Source Forge under the project name Free-MPOWER, and is available from the following address: http://sourceforge.net/projects/free-mpower/.

The repository has free and open access, and allows anyone to:

- Browse and download the source code from the Subversion repository at the site
- Access basic documentation, including executive summary of every MPOWER service and component
- Submit requests for changes and additions

Members of the repository (currently the MPOWER partners) can in addition:

- Post code
- Make documentation
- Post pictures
- Choose who can join in as contributors to the platform

5.6. *Limitations*

The studies presented herein have several limitations. First, our study on developers' use of the Service-Oriented Architecture was limited to 16 subjects which is few in order to do draw conclusions from statistical analysis. Second, we used subjective measures only. The study relies on subjective reporting of use and not actual monitored use (such as logs). Third, the study also used subjects that were working in the project, and the results should be interpreted in light of their potential bias.

6. Concluding remarks and future work

6.1. *Concluding remarks*

This paper presents an overview of the core technical results achieved in the MPOWER project. The results cover the complete development process, from requirements elicitation to evaluation of a deployed system in real environments.

The services identified from the user needs have been designed, implemented using a model-driven development toolchain. Both the services and the toolchain are available as open source from the project.

Evaluations showed that developers found it useful to apply model-driven development techniques in developing SOA services, and using the services in applications development. The systems created from the services were appreciated by the end users, indicating that their validity in the domain is high.

6.2. *Future work*

The results from the MPOWER project will be further developed as part of the universAAL project, an EU IST FP7 research project running from 2010 to 2014 (http://www.universaal.org). The main objective of universAAL is to produce a platform for AAL and to promote widespread acceptance and adoption of the platform in industry. To achieve this goal, the universAAL focuses on consolidating results from major recent and ongoing European research projects in the AAL area, including MPOWER, PERSONA, SOPRANO, OASIS, AMIGO, and GENESYS. All major results from universAAL will be made available as open source, and the project seeks to build an active developer and end user community around the project.

The MPOWER service platform will also be extended and further tested in the EU AAL project CoLiving (2010–2012). In

Summary points

What was known prior to the study:

- There are reoccurring needs among potential users of AAL systems.
- There are theoretical frameworks and tools available for model-driven development which should make it technically possible to create a tool-chain for model-driven development of SOA applications in healthcare.

What this study added to our knowledge:

- A SOA platform with a set of reusable domain services is a suitable foundation for more rapid development and tailoring of assisted living systems covering reoccurring needs among elderly users.
- There are indications that more high level tools is needed built on top of this foundation which enable tailoring of AAL systems to individual user needs, preferably without the need of professional software developer assistance.
- Applying traceability from user scenarios via use cases and features to software services is feasible using currently available modelling tools, and is found useful by the developers.
- It is feasible to realize a tool-chain for model-driven development of SOA applications in the AAL domain, and such a tool-chain can be accepted and found useful by software developers.

this project a more extensive evaluation with end users will be conducted.

The toolchain is further being tested with students and developers from industry by Walderhaug as part of his studies as a PhD candidate.

Authors' contributions

All the authors have contributed directly in the writing of this paper. During the MPOWER project, all of the authors participated in the technical work in architecture and/or implementation. Mikalsen also was the project coordinator, while Walderhaug was technical manager for the MPOWER project.

Conflict of interest

We have not identified any significant conflicts of interest that could affect our results or conclusions.

Acknowledgements

The MPOWER project was funded by the EU under the FP6 IST-2005-2.5.11 e-Inclusion, contract 034707.

We are grateful for all the efforts the MPOWER participants put into the project, and which contributed to producing project results summarized in this paper.

We would in particular like to thank Dr Mariusz Duplaga and Torhild Holthe for their consent to include quotations from the report on the results of the evaluation of the pilot applications. Dr. Duplaga coordinated the evaluation task for the sensor-centric pilot in Jagiellonian University Medical College, while Holthe coordinated the evaluation task for the information-sharing pilot in Norwegian Centre for Dementia Research, both within MPOWER project activities.

REFERENCES

[1] L. Magnusson, E. Hanson, M. Borg, A literature review study of information and communication technology as a support for frail older people living at home and their family carers, Technology and Disability 16 (2004) 223–235.

[2] European Commission, "The 2009 Ageing Report: Economic and budgetary projections for the EU-27 Member States (2008–2060)," in 2009 Aging report, European Commission (DG ECFIN) and the Economic Policy Committee (AWG) Ed. Brussels European Commission, 2009.

[3] R. Haux, Health information systems—past, present, future, International Journal of Medical Informatics 75 (2004) 268–281.

[4] OASIS Open, "Reference Model for Service Oriented Architecture 1.0," OASIS Open soa-rm-cs, August 2, 2006. http://www.oasis-open.org (accessed 5.08.10).

[5] Middleware platform for eMPOWERing cognitive disabled and elderly, MPOWER Consortium, http://www.sintef.no/mpower (accessed 5.08.10).

[6] Object Management Group (OMG), "UML 2.1.2 Superstructure and Infrastructure," Object Management Group (OMG), 2007. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML (accessed 5.08.10).

[7] A. Honey, B. Lund, "Service Oriented Architecture and HL7 v3: Methodology," HL7 Service Oriented Architecture Special Interest Group (SOA SIG) November 10, 2006. http://hssp.wikispaces.com (accessed 5.08.10).

[8] T. Erl, Service-Oriented Architecture Concepts, Technology, and Design, Prentice Hall, Crawfordswille, IN, USA, 2006.

[9] T. Holthe, S. Walderhaug, Older people with and without dementia participating in the development of an individual plan with digital calendar and message board, Journal of Assistive Technology (2010), doi:10.5042/jat.2010.0278.

[10] B. Prazak, A. Hochgatterer, T. Holthe, S. Walderhaug, User requirements as crucial determinants for the development of new technological solutions for elderly care—exemplified in an European project, in: Proceedings from AAATE, San Sebastian, Spain, 2007.

[11] S. Walderhaug, E. Stav, M. Mikalsen, Reusing models of actors and services in smart homecare to improve sustainability, Studies in Health Technology and Informatics 136 (2008) 107–112.

[12] S. Walderhaug, G. Hartvigsen, E. Stav, Model-driven traceability in healthcare information systems development, in: Proceedings from MEDINFO 2010, Cape Town, South Africa, 2010.

[13] S. Walderhaug, M. Mikalsen, I. Benc, G. Loniewski, E. Stav, Factors affecting developers' use of MDSD in the healthcare domain: evaluation from the MPOWER project, in: Proceedings from Code-centric to Model-centric Development, Workshop at European Conference on Model-Driven Architecture, Berlin, Germany, 2008.

[14] S. Walderhaug, E. Stav, Experiences from model-driven development of homecare services: UML profiles and domain models, in: Proceedings from 2nd International Workshop on Model-Based Design of Trustworthy Health Information Systems (MOTHIS 2008), Toulouse, 2008.
[15] S. Walderhaug, E. Stav, M. Mikalsen, The MPOWER tool chain—enabling rapid development of standards-based and interoperable homecare applications, in: Proceedings from Norsk Informatikk Konferanse (NIK), Oslo, 2007, pp. 103–107.
[16] Mikalsen, M., Hanke, S., Fuxreiter, T., Walderhaug, S., Wienhofen, L. "Interoperability Services in the MPOWER Ambient Assisted Living Platform" (pp 366-370) Medical Informatics in a United and Healthy Europe, Proceedings of MIE 2009, Volume 150, Studies in Health Technology and Informatics, Edited by: K.-P. Adlassnig, B. Blobel, J. Mantas and I. Masic, August 2009, 1064 pp., hardcover, IOS-Press, ISBN: 978-1-60750-044-5.
[17] MPOWER source code, available from Sourceforge at http://sourceforge.net/projects/free-mpower (accessed 5.08.10).
[18] European Commission, "The 2009 Ageing Report: Economic and budgetary projections for the EU-27 Member States (2008-2060)," in 2009 Aging report, European Commission (DG ECFIN) and the Economic Policy Committee (AWG) Ed. Brussels European Commission, 2009.
[19] J. Wancata, M. Musalek, R. Alexandrowicz, M. Krautgartner, Number of dementia sufferers in Europe between the years 2000 and 2050, European Psychiatry 18 (2003) 306–313.
[20] C. Pettey, Gartner Says SOA Will Be Used in More Than 50 Percent of New Mission-Critical Operational Applications and Business Processes Designed in 2007, Garther Media Relations 2007 Press Releases, 2007.
[21] K. Kawamoto, D. Lobach, Proposal for fulfilling strategic objectives of the US roadmap for national action on decision support through a service-oriented architecture leveraging HL7 services, Journal of the American Medical Informatics Association 14 (2007) 146–155.
[22] J. Mykkänen, A. Riekkinen, M. Sormunen, H. Karhunen, P. Laitinen, Designing web services in health information systems: from process to application level, International Journal of Medical Informatics 76 (2007) 89–95.
[23] Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA, A. Arsanjani, http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/, November 2, 2007.
[24] UML 2.0 Profile for Software Services, S. Johnston, http://www.ibm.com/developerworks/rational/library/05/419_soa/ (accessed 5.08.10).
[25] CEN TC251, "EN 13940-1: Health Informatics – System of Concepts to Support Continuity of Care – Part 1: Basic Concepts," European Committee for Standardization, September 26, 2006.
[26] D.H. Stefanov, Z. Bien, W.-C. Bang, The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives, IEEE Transactions on Neural Systems and Rehabilitation Engineering 12 (2004) 228–250.
[27] S. Walderhaug, E. Stav, U. Johansen, G.K. Olsen, Traceability in Model-driven Software Development, in: P.F. Tiako (Ed.), Designing Software-Intensive Systems—Methods and Principles, IGI Global, Information Science Reference, Hersey, PA, 2008, pp. 133–160.
[28] Nederlandse Vereniging voor Maatschappelijke Dienstverlening en Samenlevingsopbouw en de Algemene Woonbond voor Ouderen, "Woonbewust, Handleiding bij de bijeenkomsten Over Wonen van Oouderen Gesproken," Amsterdam, Netherlands, 1993.
[29] F.D. Davis, R.P. Bagozzi, P.R. Warshaw, User acceptance of computer technology: a comparison of two theoretical models, Management Science 35 (1989) 982–1003.
[30] V. Venkatesh, F.D. Davis, A theoretical extension of the technology acceptance model: four longitudinal field studies, Management Science 46 (2000) 186–204.
[31] ISO/IEEE, "ISO/IEEE 11073-10404:2004 Health informatics – Point-of-care medical device communication," in Part 10404: Device specialization – Pulse oximeter.: ISO, 2005.
[32] ISO/IEEE, "ISO/IEEE 11073-20101:2004 Health informatics – Point-of-care medical device communication," in Part 20101: Application profiles – Base standard, 2005.
[33] ISO/IEEE, "ISO/IEEE 11073-10408:2004 Health informatics – Point-of-care medical device communication," in Part 10408: Device specialization – Thermometer, 2005.
[34] ISO/IEC, "ISO/IEC 14543-3-6:2007 Information technology – Home electronic system (HES) architecture –" in Part 3-6: Media and media dependent layers – Network based on HES Class 1, twisted pair, 2007.
[35] ASTM, "Standard Specification for Continuity of Care Record (CCR)," in Active Standard ASTM E2369 Developed by Subcommittee.
[36] U.S. Department Of Health and Human Services, Food and Drug Administration Center for Devices, and Radiological Health Center for Biologics Evaluation and Research, "General Principles of Software Validation; Final Guidance for Industry and FDA Staff," 2002.
[37] R. Lenz, M. Beyer, K.A. Kuhn, Semantic integration in healthcare networks, International Journal of Medical Informatics 76 (2007) 201–207.
[38] S. Walderhaug, E. Stav, S.L. Tomassen, L. Røstad, N.B. Moe, MAFIIA—an architectural description framework: "Experience from the Health Care Domain", in: D. Konstantas, J.-P. Bourrières, M. Léonard, N. Boudjlida (Eds.), Interoperability of Enterprise Software and Applications, Springer Verlag, Geneva, Switzerland, 2005, pp. 43–54.
[39] R. Grønmo, D. Skogan, I. Solheim, J. Oldevik, Model-driven web services development, The International Journal of Web Services Research 1 (2004) 1–14.
[40] D. Skogan, R. Grønmo, I. Solheim, Web service composition in UML, in: Proceedings from EDOC, 2004, pp. 47–57.
[41] T. Itala, J. Ukkola, A. Virtanen, J. Mykkanen, SOA approach for integration of departmental systems, Studies in Health Technology and Informatics 136 (2008) 723–728.
[42] J. Mykkänen, Specification of Reusable Integration Solutions in Health Information Systems, Doctoral Dissertation, Kuopio University Publications, Finland, Kuopio, Finland, 2007.
[43] C.W. Krueger, Software reuse ACM Computing Surveys (CSUR), vol. 24, 1992, pp. 131–183.
[44] The European Ambient Assisted Living Innovation Alliance, "Ambient Assisted Living Policy Recommendations," AALIANCE, European Commission, 2009. http://www.aaliance.eu/public/documents/policy-recommendations/aal-policy-recommendations-march-2009.pdf (accessed 5.08.10).
[45] European Commission, "e-Health—making healthcare better for European citizens: An action plan for a European e-Health Area," European Commission, Brussels, Belgium, 2004. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2004:0356:FIN:EN:PDF (accessed 5.08.10).