SERVICE REALIZATION AND COMPOSITION ISSUES IN THE HOMECARE DOMAIN

Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota and Marten van Sinderen
Department of Electrical Engineering, Mathematics and Computer Science
University of Twente, Enschede, The Netherlands

Keywords: Homecare, Independent living, Service realization, Service composition, SOA, Homecare platform, Wellbeing.

Abstract: Most of the industrialized countries are facing a new situation which is the increasing number of elderly people and the enormous cost of associated healthcare. IT-based homecare services have been proposed as a solution to alleviate this problem. However, the adoption of the homecare services is hindered by the fact that there are many heterogeneous IT solutions, which prevent interoperability and integration of existing as well as future applications. Service-Oriented Architecture and its advantage of service composition in a loosely-coupled manner are being considered as a promising approach to promote interoperability. In this paper, we explain our motivation for and understanding of a generic service-oriented homecare platform. We consider the SOA paradigm and its idea of logical separation of concerns to define a service realization and composition framework and to identify its challenges in order to realize and compose the homecare services. Using our framework, we evaluate three academic homecare projects to see how they address the identified challenges. The work presented in this paper helps researchers to understand the state-of-the-art of homecare-related platforms and technologies; and to outline issues for further research in the homecare domain.

1 INTRODUCTION

The population of many Western-European countries have an increasing percentage of elderly people and, consequently, the healthcare-related cost for these countries is growing (European Commission, 2007). Providing automated care support for elderly in their own home, i.e., homecare services, is proposed as a highly desirable solution to alleviate the problem of the growing healthcare related cost, and consequently has attracted attention from the ICT industry (European Commission, 2007) (Gassner and Conrad, 2010). Homecare service provisioning often utilizes a set of heterogeneous software and hardware solutions which must cooperate seamlessly. A comprehensive solution, like a common platform to make them interoperable, is still an open research area. Distributed, dynamic and heterogeneous environments increasingly challenge the goal of homecare services (Bricon-Souf et al., 2005)(Bottaro et al., 2008).

The recent progress in ubiquitous computing, Remote Monitoring and Treatment (RMT), and domotic applications leads to many useful components and technologies for the homecare services provisioning. Therefore, an integration solution which can bring all these components and technologies together seems more interesting than any time before. SOA with its service composition principle provides a promising approach for homecare platforms that support seamless cooperation of heterogeneous technology solutions. Several efforts to define such a service-oriented platform have been undertaken (Bottaro et al., 2008)(Mikalsen et al., 2009)(Gray et al., 2007). We believe in service-oriented homecare platforms and therefore, we aim to explain our motivation for and understanding of such a platform.

The homecare services hosted on the platform can be composed and realized at different levels, from low level hardware devices to high level end-user application services. To elaborate these composition hierarchy, we define a service realization and composition framework to show different layers of functionality while each layer employs the services, i.e, functionalities of its lower layer and provides new functionalities for its next higher layer. The lower layers mostly concern about service realization while the higher layers concern about service composition.

Our objective is to define a service realization and composition framework consisting of several functional layers, for a generic SOA-based homecare plat-

Zarghami A., Zarifi Eslami M., Sapkota B. and van Sinderen M..
SERVICE REALIZATION AND COMPOSITION ISSUES IN THE HOMECARE DOMAIN.
DOI: 10.5220/0003630203470356
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ACT4SOC-2011), pages 347-356
ISBN: 978-989-8425-76-8

form to position and classify related challenges as well as their corresponding solutions which are already utilized. As such, the proposed framework outline issues for further research on homecare service realization as well as composition challenges. The framework can also be used to explain the state-of-the-art with respect to homecare platforms, showing how these platforms address the challenges at different layers.

The remaining of this paper is organized as follows: In Section 2, we discuss interoperability from information and behaviour perspectives, and how existing service-oriented homecare platforms tackle the interoperability challenges. We also clarify our understanding of a generic homecare platform. In Section 3, we explain some of interoperability issues in the homecare domain which are used to classify the challenges in our framework. In Section 4, we elaborate our service composition framework. In Section 5, we map three homecare-related academic projects to our framework to evaluate their approaches with respect to the identified challenges. Finally, in Section 6, we conclude and discuss our work and explain our planning of future work.

2 SOA AND HOMECARE

The adoption of homecare services is hindered by the fact that there are many heterogeneous ICT solutions, which prevent interoperability and integration of existing as well as future applications. To this end, Service-oriented Architecture and its advantage of service composition in a loosely-coupled manner are being considered as a promising approach to promote interoperability (Bottaro et al., 2008) (Mikalsen et al., 2009) (Gray et al., 2007) (Nee et al., 2008).

2.1 Role of SOA in Homecare

The complexity of interoperability and its multifaceted characteristics lead to several definitions (Kosanke, 2006). The IEEE Glossary defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" (IEEE, 1990). System can be understood in terms of different perspectives such as information and behavior. The interoperability challenges can be considered based on these different perspectives of systems.

The interoperability challenges from the *information* perspective can be seen as the difficulty in sharing information between systems due to differences in coding, formatting and data representation schemes. An existing homecare platform, SAPHIRE, employs a cross-enterprise document sharing architecture to tackle information interoperability like discovery of and access to relevant electronic health records(EHR) (Nee et al., 2008). The interoperability challenges from the *behaviour* perspective can be seen as the difficulty in using each others functions due to differences in the protocols at which they are used. Another existing homecare platform, MPOWER, introduces a solution based on SOA and a message translator pattern to provides interoperability services for Ambient Assisted Living (AAL) environment (Mikalsen et al., 2009).

2.2 Homecare Platform

An application platform, according to ISO/IEC 14252, is defined as *"a set of resources, including hardware and software that support the services on which application software will run"* (IEEE, 1998). By homecare platform, we mean an application platform which provides technology-independent services for homecare applications in the sense that it hides the complexity of underlying software and hardware entities. The platform employs a set of physical devices like to provide services which are utilized by homecare applications. The platform services are accessible through a set of interfaces that make the characteristics of underlying technology as transparent as possible.

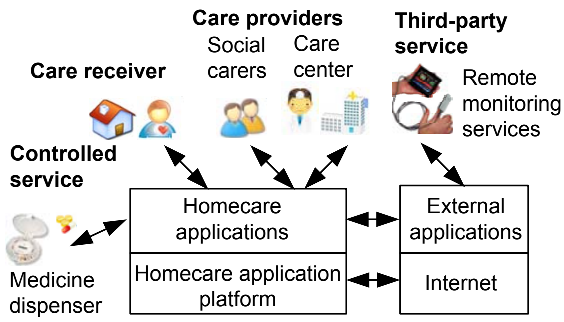

Figure 1: The overall view of a generic homecare platform.

Our overall view of a generic homecare platform is shown in Figure 1. There are four types of stakeholder who interact with the homecare platform and applications. The *Care receivers*, live at the place, for instance, apartment where the platform has been installed. They uses a set of interfaces either computer-based GUI or physical devices like a *Medicine dispenser* to receive care services. On the other hand, the *Care providers*, either professional from a *Care center* or *Social carers* like the neighbors and friends, provide care services through the platform. They can even act as an application developer, to create a new application based on exiting platform services (provided by the platform or the applications) or tailor the current applications for a specific care receiver. The *Controlled services* providers, as the third type of stakeholder, provide the components which are installed within the platform, for example, a *Medicine dispenser*. The platform should provide standard interfaces for the *Controlled services* providers to install and manage the life cycle of their components, without interfering the operation of other components. The fourth type of stakeholder is the *Third-party service* providers whose services are accessible through the Internet and can be exploited by applications running on the homecare platform. For instance, the *Remote monitoring services* provided by *MobiHealth* [1] can be used by other homecare applications to check oxygen saturation and blood pressure of the *Care receivers*. MobiHealth can have its own infrastructure and devices like COPD sensors inside the home separate than the homecare platform.

3 INTEROPERABILITY ISSUES

To achieve interoperability for the homecare platform, distribution, heterogeneity and dynamicity (Bottaro et al., 2008) should be addressed in design and implementation time. In this subsection, we explain these issues and their causes in homecare domain.

Distribution. In distributed system, a reliable medium for communication and seamless addressing is considered as an interoperability challenge. The medium should provide location transparency and hide the complexity of underlying communication infrastructure. Care receivers and care providers should be able to use services provided by homecare applications which are located in different places, for instance, different rooms, through the platform. Care providers may use homecare applications from both inside and outside home. External services provided by the *Third-party service* providers should be accessible through the Internet for the *Homecare applications*.

Heterogeneity. Interoperability is especially hard to achieve if distributed systems are made up of heterogeneous subsystems that are developed, managed and/or owned by different organizations. From the software point of view, the applications might follow different protocols for discovery (e.g., WS-Discovery, SLP), interaction (e.g., SOAP,RMI) and transportation (e.g. TCP, UDP). From a hardware point of view, they can have different capabilities varying from a resource-constrained sensor to a computing device with high computation resources. The software and hardware heterogeneity arise a set of constraints which should be taken into account in implementation time.

Dynamicity. We define dynamicity in homecare domain as any type of change and demand which makes current behaviour of homecare applications not desired for the stakeholders. The changes and demands, coming from the care receiver, homecare applications and environment, can arise dynamicity either in short-term or long-term (McBryan et al., 2008). From care receiver point of view, some of the dynamicity like a visual impairment development occur over a long period of time and can be addressed in design time. On the other hand, some others like location and activity might change in short-term and must be handled in runtime. From a system point of view, most changes like device availability and location, are short-term. As long-term systems' dynamicity, we can mention relocation of physical devices, for example, the closest medicine dispenser to the kitchen, after the relocation of physical devices, is not the closest one anymore.

4 SERVICE COMPOSITION FRAMEWORK

As explained before, a generic homecare platform enhances the creation of new homecare applications and tailoring the current applications for a specific care receiver. We propose a service realization and composition framework to show different functional layers and to identify corresponding challenges. To add to the understanding of our framework definition, we demonstrate functionality of the layers based on two well-known application scenarios for homecare applications as follows:

- *Remote Medicine Prescription.* A doctor collect information about health condition like oxygen saturation of John, as an elderly who has minor form of COPD (Chronic obstructive pulmonary disease), living in a home equipped with homecare services. The doctor may requires extra information such as John's position (for example, training or bed room) along with his oxygen saturation. The doctor can set different oxygen saturation thresholds for different positions of John, for instance, bedroom or training room to be notified. John can consult with the doctor through a audio/video communication on his

[1] http://www.mobihealth.com/

Tablet(Tablet PC) or PDA (Personal Digital Assistant) and finally the doctor prescribes the necessary medicine. A medicine dispenser device will be programmed based on the prescription.

- *Medicine Reminder.* The homecare services will remind John of taking the medicine on time. This reminder message can be shown through Tablet or PDA based on the John's preferences. For instance, if he is in living room, he prefers to see the message on the Tablet instead of PDA.

We demonstrate our framework in three steps. First, we explain our methodology for defining the framework. Second, we explain the functionality and the composition relationships of the layers. By composition relationship we mean that a layer composes a set of functionalities provided by the layer lower than itself (except the lowest layer). Third, we discuss the challenges imposed by the homecare-related interoperability issues for each of the composition relationships. Due to service composition hierarchy, we define the composition relationships between the layers.

4.1 Methodology

The functional aspect of homecare services are the concerns of the work presented in this paper and thus we focus on interoperability challenges from the behaviour perspective. We consider behaviour perspective because it allows us to look at the systems behaviour while using each others' functions. It is difficult to relate the information interoperability to the functional layers as defined in our framework. Therefore, we do not consider information interoperability challenges in this paper.

In order to classify challenges and break them down to smaller ones, we employ the functional layering of SOA as defined in (Arsanjani, 2004), as the basis for defining our service realization and composition framework. The service-oriented architecture provides guidelines to achieve interoperability that assists different organizations to be dynamically integrated regardless of their technologies, platform and application specification. The SOA paradigm can be considered as a set of distinct layers of abstractions which describes a logical separation of concerns (Arsanjani, 2004). Each layer employs the functionalities of its lower layer and provides new functionalities for its next higher layer.

Based on the work presented in (Arsanjani, 2004), we defined a service realization and composition framework as a set of five functional layers. These layers are numbered 1 to 5 and cover different aspects, of a homecare domain, spanning from low level hardware devices to high level end-user application services. The lowest layer is numbered 1 and the highest layer is numbered 5 as shown in Figure 2. The layers are defined such that each layer employs the functionalities of its lower layer and provides new functionalities to its next higher layer. Given two layers of the proposed framework, a layer is considered lower than the other if it covers more concrete entities of the homecare domain. This means that the lower layers are mostly concerned about service realization while the higher layers are mostly concerned about service composition. This kind of layering is useful in analyzing problems in the homecare domain in a systematic way.

In the proposed framework, physical devices like sensors are located in the lowest layer which is numbered 1. The software components which realize service interfaces to communicate with the physical devices are placed on the next layer which is numbered 2. The software components which provide a uniform service layer on top of all the existing service interfaces to hide the heterogeneity and distribution of underlying components are placed in the layer which is numbered 3. A set of basic functionalities which can be composed and configured by the care providers to create homecare applications are placed in the layer which is numbered 4. This layer hides the details of concrete implementations of the lower layers. The highest layer, which is numbered 5, contains the homecare applications which directly interact with care receivers.

4.2 Functionality of the Layers

Our proposed homecare service realization and composition framework is shown in Figure 2. The left part shows a functional layering which are provided by homecare applications. The right part presents a platform which provides several services like messaging and coordination to the application functional layers. The platform enhances the development and execution of the homecare applications. Since, we aim to emphasize on service realization and composition challenges from the application point of view, we do not discus the platform part in this paper. Not to mention, communication relationships among geographically distributed systems are established through the platform. Composition relationships among layers are shown by a line ended with circle and can have 1:n and 1:1 cardinality which will be individually discussed in each layer.

Figure 3 shows an instance of the service hierarchy of the homecare applications for the two scenarios which is mentioned in the beginning of this section. The lines between layers show the composition rela-

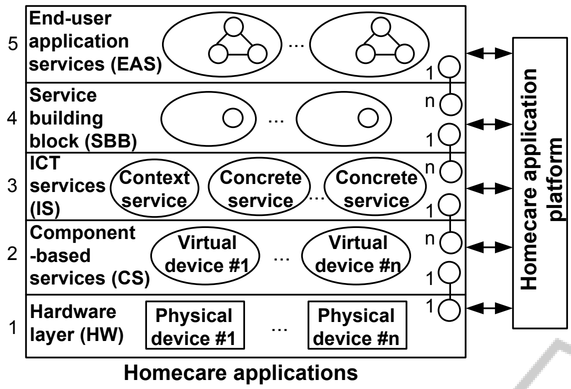

Figure 2: Service hierarchy of the homecare applications.

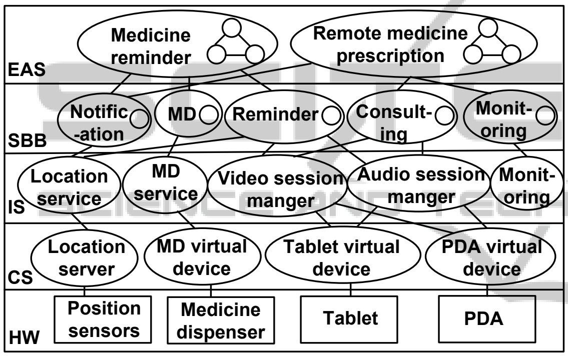

Figure 3: An instance of the service hierarchy of the homecare applications for the two scenarios.

tionships which follow the cardinality of relationships between layers shown in Figure 2.

Hardware (HW). In the this layer, we have a set of physical devices such as sensors, actuator and multimedia interfaces. For the target scenarios, we can mention position sensors, medicine dispenser, Tablet and PDA. The COPD sensors to check oxygen saturation is managed by the *Third-party service* providers, therefore the platform does not have any responsibility for the COPD sensor and we do not consider it in this layer. The positioning sensors are employed to determine John's location. The constituents of the *HW* layer can only provide their functionalities to their representative components located in the next higher layer. Therefore, any interaction with the devices located in the *HW* layer, should be done through their corresponding *virtual devices* implemented in the *Component-based services (CS)* layer. The composition relationship between *HW* and *CS* layer has 1:1 cardinality. It means that for each physical device in the *HW* layer, there is only one representative *Virtual device* in the *CS* layer.

Component-based Services (CS). In this layer, we have a set of *virtual devices* which can be seen as service realization out of the physical devices in the lower layer. In general, the *virtual devices* realize the services by three main strategies: wrapper, adapter and generic service interface. As the wrapper, a software component, encapsulates a physical device and provides its own service interface like a OSGi bundle (Gray et al., 2007). As the adapter, a software component or an application, plays an intermediary role between the physical devices and the upper functional layer. For instance, a component so-called frame sensor adapter (FSA) allows to uniformly collect data from sensors (Loniewski et al., 2009). As the generic service interface, the *virtual devices* provide a set of generic interfaces to call physical devices. The *Controlled services* providers who produce the physical devices can implement the interfaces in their own. For the target scenarios, *MD (medicine dispenser) virtual device* provides access to program and activate the physical *Medicine dispenser*.

ICT Service (IS). In this layer, we have two type of services: *Context* and *Concrete services*. *Context* services aggregate raw data from one or several *virtual devices* to infer whether trigger a particular event or not. Accordingly, the composition relationship between the *IS* and *CS* layers has 1:n cardinality. We define the *Concrete services* to hide the heterogeneity and distribution of the *virtual devices* and to provide a uniform protocol to discover and interact with them. The *Concrete services* have two main functionalities: mediation and messaging. The mediation provides interoperability between two or several *virtual devices* which might use different discovery and interaction protocols. The messaging provides transparent communication with the *virtual devices*. These two functionalities are supported by the platform services. For the target scenarios, the *Video session manager* provide a uniform access for Tablet and PDA virtual devices to mange a video communication session with a standard protocol like SIP (Session Initiation Protocol). The video streaming infrastructure is considered as part of the platform. The *Monitoring* concrete service is provided by the *Third-party service* providers through the platform services over the Internet without any corresponding virtual device in our framework.

Service Building Block (SBB). In this layer, we place homecare related service building blocks (SBB), i.e., the smallest manageable services, which cannot be broken down further into smaller services from the care providers' point of view. However, from the platform's point of view, these services might exploit one or several of the *Concrete services* provided by the *IS* layer. For instance, the *Reminder service* employs the *Location service* to identify John's location and remind him through *Audio session manager* or *Audio session manager*. Therefore, the *SBB* layer has 1:n composition relationship with the *IS* layer. In our scenarios, from the care providers' point of view, there are several *SBB*s like *MD* and *Reminder*. The *SBB*s are reusable and context-independent basic services which can be composed to create and tailor homecare applications. Each of the *SBB*s can be seen as a service interface with a set of operations. Beside the *SBB*s, the homecare applications needs a set of events to implement their logics. Due to the SOA paradigm, we define the *Notification* services to handle these events. The *Notification* is defined based on one or several events triggered by the services located in the *IS* layer. For instance, a medicine reminder notification to trigger *Medicine reminder* application can be defined based on two events, the availability of the medicine dispenser and John being at a predefined place.

End-user Application Services (EAS). In this layer, we place the homecare applications which are represented by a set of service plans to fulfill the homecare scenario objectives. For instance, the objective of the *Medicine reminder* scenario is to remind the care receiver to take medicine on time. The service plan specifies the application behaviour in runtime and can be defined by composing services. Each homecare application has one or several service plans consist of one or several *SBB*s. Therefore, the composition relationship between the *EAS* and the *SBB* layer has 1:n cardinality. We classify the functionality of this layer in three steps: generation, evaluation and transformation. In the first step, a service plan is generated by the care providers based on existing service interfaces of *SBB*s. In the second step, the service plan should accommodate the care receiver's preferences and the runtime situation of the environment, to prioritize the alternative possibilities of service execution. In the third step, the service plan must be transformed to an executable composed service by mapping the service interfaces to concrete service providers like an available medicine dispenser.

4.3 Challenges between the Layers

As mentioned in our methodology, we focus on the interoperability challenges of systems as understood from the behaviour perspective in the homecare domain. We associate the interoperability challenges with the composition relationships between the functional layers as defined in our framework. These challenges which are categorized into the three homecare-related interoperability issues (explained in Section 3), are discussed below:

HW-CS. The composition relationship between these two layers is one-to-one. This means that from the *IS* layer there is no distinction between physical devices and their corresponding virtual devices. Therefore, we do not consider any dynamicity and distribution challenges in this composition relationship. However, we consider the heterogeneity challenge because the hardware devices could come from different manufacturers and with different proprietary solutions. We identify three challenges with respect to device heterogeneity. a) *Behaviour*: providing a uniform service interface for the physical devices despite their heterogeneous behaviour is a tough challenge. The uniform service interface should support both static and non-static characteristics of the behaviour of the virtual devices. Different virtual devices, for example, can have the same static characteristics like service interface signature with different non-static characteristics like various order of function access. b) *Resource-constrained devices*: having various physical devices with resource limitations (e.g., memory size, computation power) leads to several constraints on their usage. c) *communication*: physical devices could have different communication protocols which the virtual devices should support to provide communication protocol transparency or even location transparency.

CS-IS. The composition relationship between these two layers is many-to-one. This means that one ICT service could communicate with one or more component-based services. Therefore, we consider dynamicity, heterogeneity and distribution challenges in this composition relationship. Specifically, we identify three challenges with respect to dynamicity. a) *Life-cycle of virtual devices*: providing a cooperation environment is required to enable several *Controlled services* providers to manage their component life-cycle (for example, install and configure) without interfering other components' operations. b) *Plug-and-Play support*: availability of the virtual devices, in homecare environment, is not guaranteed. The virtual devices should be allowed to register itself to the system automatically and be operational as soon as they become available. c) *Dynamic binding*: specifying concrete virtual devices in service plan is difficult because the virtual devices could come and go. Therefore, the platform should bind the service plan to the concrete virtual devices based on runtime situation.

We identify one challenge with respect to the heterogeneity. *Discovery/Interaction protocols*: definition of the *CS* layer implies that there should be no heterogeneity issue above that layer. However, some strategies used by the *CS* layer like the generic service interface, only address the heterogenous behaviour of the virtual devices. Therefore, there is a need between *CS* and *IS* layers to support different discovery (for example, WS-Discovery, SLP) and interaction (for example, SOAP, RMI) protocols. We also identify one challenge with respect to distribution. *Location transparency*: component-based services could be physically distributed. In that case, providing a location transparent communication infrastructure is a challenge.

IS-SBB. The composition relationship between these two layers is many-to-one. This means that one service building block could communicate with one or more *IS* services. We assume that the IS layer hides all issues related to distribution and heterogeneity. Therefore, we consider only the dynamicity challenges in this composition relationship. We identify three challenges with respect to dynamicity. a) *QoS-aware provisioning*: providing information about dynamic QoS of home network (Bottaro et al., 2008) which is used as the underlying communication technology in the homecare domain (for example, dynamic bandwidth) is essential to improve the homecare application performances. It is required to bind *SBB* to more reliable services which can decrease the possible future interoperability problems that may appear otherwise, i.e., if the *SBB* is bound to unreliable services. b) *Runtime matching*: in runtime, the platform might bind a particular *SBB* to various concrete services due to their (un)availability in different times. This can be done by automatically matching the available concrete services with the *SBB*s. By using QoS information provided by the *IS* layer, even in case of availability of several concrete services, the platform can bind the *SBB*s to the most reliable concrete service. c) *Generic SBB model*: providing a generic mechanism to model the *SBB*s, understandable for both the platform and care providers, is considered as a challenge, specially due to the complexity of the behaviour of the services provided by the lower layers and their dynamic QoS which are not known at design time. The *SBB*s should have two representations: internal and external. The internal view is used to be interpreted by the platform while the external format is used to present the *SBB*s to the care providers.

SBB-EAS. The composition relationship between these two layers is many-to-one. This means that one end-user application service could be composed by one or more *SBB*s. In this composition relationship, we consider only the dynamicity challenges but not the heterogeneity and distribution challenges because of the same reason as mentioned in *IS-SBB* composition relationship. We identify two challenges with respect to dynamicity. a) *Runtime adaption*: in runtime, if there is no available concrete services for a specific *SBB*s, the platform has to adapt the service plan itself by substituting that *SBB* with one or several other *SBB*s whenever possible. It can automatically compose, for instance, a set of available *SBB*s to fulfill the functionality of one particular *SBB*. The service plan adaption should be able to accommodate the information about QoS of services provided by the lower layer for the most reliable adaption. b) *Simple service plan*: To have a realistic solution, with respect to large variety of possibly changes coming from both the care receivers and homecare applications, providing a convenient approach for the care providers to assist them in generating the service plan is an essential need. The summary of these challenges are shown in Table 1.

Table 1: The summary of challenges between the functional layers (Dy:Dynamicity, Hg:Heterogeneity, Ds:Distribution).

	Interoperability challenges	Dy	Hg	Ds
SBB-EAS	Runtime adaption	X		
	Simple service plan	X		
IS-SBB	QoS-aware provisioning	X		
	Runtime matching	X		
	Generic SBB model	X		
CS-IS	Life-cycle of virtual devices	X		
	Plug-and-play availability	X		
	Dynamic binding	X		
	Discovery/Interaction protocols		X	
	Location transparency			X
HW-CS	Behaviour		X	
	Resource-constrained devices		X	
	Communication		X	

5 EVALUATION

Based on our study, there are two well-known technologies exploited by homecare applications: OSGi and web services. To cover a wide range of solutions in SOA-based homecare applications, we have studied academic homecare projects from the both aforementioned technologies. We have selected three of these projects as examples to show the feasibility of our proposed framework. They use discrete heuristics in each of the layers as defined in our framework. The three selected projects are as follows: (1) *MATCH* [2]*(Mobilising Advanced Technologies for Care at Home)*, (2) *MPOWER* [3] *(Middleware Platform for eMPOWERing cognitive disabled and elderly)*, (3) *Amigo* [4] *(Ambient Intelligence for the networked home environment)*.

[2]http://www.match-project.org.uk/main/main.html

[3]http://www.sintef.no/Projectweb/MPOWER/

[4]http://www.hitech-projects.com/euprojects/amigo/

Table 2: The technologies and solutions are employed by the three projects.

	Match(1)	MPOWER(2)	Amigo(3)
EAS	XML, Directed graph Interactive evaluation	BPEL, UML service models	BPEL, OWL-S Online semantic reasoning
SBB	XML, Interaction model OWL-based events	WSDL, IBM UML Profile	OWL-S, Amigo-S Rule-based notification
IS	Message broker component	Open ESB	Middleware SDP and SIP meditation
CS	OSGi, UPnP Wrapper	Platform-independent models Adapter	OSGi, .NET, UPnP Generic service interface
HW	Residential gateway	Proprietary framework (FSA)	Domotic infrastructure

Table 2 summarizes the technologies and solutions exploited by these projects in each of the functional layers as defined in our framework. Moreover, Table 3 shows to what extent these projects support the identified interoperability challenges for each of the functional layers. In continue, we explain the two tables together from the lowest layer and gradually move to upper layers.

In the ***HW* layer**, *Match* uses OSGi component, i.e., bundles to represent physical devices. With respect to *Resource-constrained devices*, these bundles are located in a residential gateway or any intermediary which has JVM installed and can communicate with their corresponding devices through heterogeneous *Communication* protocols. *MPOWER* exploits a proprietary framework, so-called Frame Sensors Adapter (FSA), to collect data from several sensors. *MPOWER* delegates management of heterogeneous *Communication* protocols and *Resource-constrained devices* to FSA. *Amigo* has developed a domotic infrastructure to decouple the technologies used in the *HW* layer from the virtual devices in the *CS* layer. The infrastructure supports heterogeneous *Communication* protocols and employs C as programming language to deal with *Resource-constrained devices*.

In the ***CS* layer**, *Match* uses OSGi to manage the dynamicity of *Life-cycle of virtual devices*. It utilizes UPnP protocol to address *Plug-and-play support* by providing simplified installation upon internet-based communication protocols. The OSGi bundles address the heterogeneity of *Behaviour* of physical devices by using wrapping techniques. *MPOWER* employs a platform-independent model which can be realized on several platforms such as web services, .NET, CORBA and J2EE. We could not find explicit information with respect to the support for *Life-cycle of virtual devices* and *Plug-and-Play support* in the literature concerning *MPOWER*. *MPOWER* uses the FSA as an adapter to hide the heterogeneous *behaviour* of the physical devices. *Amigo* develops a middleware which can be installed on individual platforms to support both OSGi and .Net standards or even independent of both of them. It also follows OSGi and UPnP based techniques as used in *Match* to support *Life-cycle of virtual devices* and *Plug-and-play support*. *Amigo* has developed a domotic service model specification to address the heterogeneity of *behaviour* of physical devices. The virtual devices can use the generic service interface instead of directly communicating with the heterogeneous physical devices.

In the ***IS* layer**, *Match* has developed its own message broker for the messaging functionality of the *Concrete services*. This broker is implemented by a OSGi bundle to address *dynamic binding* and *Location transparency*. As *Match* has employed wrapping in the *CS* layer, there is no need for the mediation and the heterogeneity of *Discovery/Interaction protocols* are already supported by the *CS* layer. *MPOWER* has defined an enterprise service bus based on Open ESB for the messaging. It supports *Location transparency* but it does not completely support *Dynamic binding* (for example, dynamic endpoints). *MPOWER* like *Match* also does not need the mediation due to use of adapter in the CS layer. Amigo has developed an interoperable middleware core which provides service discovery protocols (SDP) and service interaction protocols (SIP). The SDP addresses the heterogeneity of *Discovery protocols* by parsing the input protocol and composing the target output protocol. The SDP dynamically instantiates a stub from the description and reference of the services. The stub which addresses the heterogeneity of *Interaction protocols* and *Location transparency*, can act as intermediary mediation. None of these three projects, provide *QoS-aware provisioning*. It means that runtime matching and adaption in the *SBB* and *EAS* layers, are completely unaware of the quality of *Concrete services* during runtime.

In the ***SBB* layer**, *Match* describes the *SBB*s in XML format from very abstract level to concrete interaction interface, similar to Concur Task Trees (Mori et al., 2004). The abstract level can be used as the external language for the care providers to support *Generic SBB model*. This simplicity is enhanced by a technology-agnostic description of services written in OWL (Web ontology languages) to associate a set of predefined events to their possibly corresponding virtual devices, using human-understandable concepts.

The ability of *SBB*s interface matching (McBryan et al., 2008) supports the dynamic *Runtime matching*. However, it does not accommodate the QoS of the corresponding *Concrete services*. *MPOWER* has defined *SBB*s as WSDL files for the internal purposes. Due to the complexity of WSDL for unskilled users, IBM UML profile for software services (Johnston, 2005) has been exploited as the external language to address *Generic SBB model*. We could not find explicit support mentioned in the publications of *MPOWER* for *Runtime matching*. *Amigo* develops the Amigo-S language based on OWL-S to describe services for both external and internal language with different levels of abstraction to address *Generic SBB model*. Moreover, *Amigo* provides rule-based awareness and notification services (ANS) which can enhance *Generic SBB model* for the *Notification* services. Amigo-S is part of a comprehensive approach towards semantic service description, discovery, composition, adaption and execution (SD-SDCAE) which specifies QoS as well as behaviour of services. It uses runtime semantic reasoning to support *Runtime matching*.

Finally, in the ***EAS* layer**, *Match* uses directed graph to define service plan, by matching the *SBB*s interaction interfaces (McBryan et al., 2008). It generates several service plans for a target scenario by traversing all possible paths in the graph. The abstract service plan, excluding detailed service interface, supports *Simple service plan*. The possible multi traversal paths enable the platform to provide *Runtime adaption* by generating alternative service plans based on available *concrete services* at runtime. However, QoS is not considered to evaluate possible alternative service plans. *MPOWER* has a set of predefined UML service models which are implemented by its own domain-specific modeling language (DSML) and understandable for care providers. Based on the target scenario and the existing service plan models, it generates the service plan and transforms it to executable composed service using BPEL standard. We could not find explicit support mentioned in the publications of MPOWER, for the support of *Runtime adaption*. In *Amigo*, to address *Simple service plan*, tasks are modeled by abstract workflow based on semantic contextual model which is created by a user-friendly GUI. In runtime, by using semantic service discovery, several alternative service plans are generated by composing available *Concrete services* in different ways. The *Runtime adaption* is addressed by runtime semantic reasoning. Although the service plans can be evaluated based on QoS and context properties, due to lack of QoS-aware service provisioning, the *Runtime adaption* is partially supported. Finally, the service plan is transformed to an executable composed service using BPEL standard.

Table 3: The challenges addressed by the three projects (Supported:S, Partially supported:P, Not supported:N).

	Challenges	(1)	(2)	(3)
SBB-EAS	Runtime adaption	P	N	P
	Simple service plan	S	S	S
IS-SBB	QoS-aware provisioning	N	N	N
	Runtime matching	P	N	S
	Generic SBB model	S	S	S
CS-IS	Life-cycle of virtual devices	S	N	S
	Plug-and-play availability	S	N	S
	Dynamic binding	S	P	S
	Discovery/Interaction protocols	S	S	S
	Location transparency	S	S	S
HW-CS	Behaviour	S	S	S
	Resource-constrained devices	S	S	S
	Communication	S	S	S

6 CONCLUSIONS

In this paper, we introduce a functionality layering framework to analyze the issues related to service realization and composition in the homecare domain. This framework is useful for both the service developers as well as the researchers. Developers can systematically breakdown the problems in different layers and find the corresponding solutions whereas the researchers can use the framework to find issues in the existing solutions. We also discussed what interoperability challenges exist between different functional layers. This helps developers and domain experts to identify such challenges pertaining to the homecare domain without the need of a specific methodology to apply the proposed framework for the same purpose.

Finding the related works to compare the proposed framework was one of the difficulties we faced while performing the work presented in this paper. To the best of our knowledge, the proposed framework is the first one that attempts to classify service realization and composition issues in the homecare domain. A service composition framework introduced in (Rao and Su, 2005) assumes availability of services and focuses on automatic service composition. In the homecare service provisioning, availability of the homecare services is not a valid assumption. Due to several low level hardware devices which might be employed by the homecare service provisioning, service realization challenges should also be taken into account.

We used our framework to analyze three existing SOA based homecare projects and to demonstrate the applicability of the proposed framework. Though we selected only three SOA based homecare projects, the proposed framework can be used to analyze other SOA based homecare projects to investigate to what extend the identified challenges are addressed by them. Analysis of larger number of SOA based homecare platforms would even help us define homecare challenges more accurately and even to refine the proposed framework. If we perform a large scale analysis of homecare platforms we could possibly identify some new challenges or the challenges which are already identified may disappear. Such a large scale analysis, however, would require more research efforts and is left as part of our future work.

Based on our study, we observed that existing solutions can handle distribution and heterogeneity issues. However, the dynamicity issues, which affects almost all the functional layers and plays an important role in homecare service realization, are not well addressed and therefore further research is required. With respect to dynamicity, the service plan should be simple but detailed enough to enable the homecare platform to match and adapt them to the available concrete services at runtime in a lightweight and QoS-aware manner. With this observation, we aim at addressing these issues in our future works.

ACKNOWLEDGEMENTS

This work is part of the IOP GenCom U-Care project(http://ucare.ewi.utwente.nl) which is sponsored by the Dutch Ministry of Economic Affairs under contract IGC0816.

REFERENCES

Arsanjani, A. (2004). Service-oriented modeling and architecture. *IBM developer works*.

Bottaro, A., Simon, E., Seyvoz, S., and Gerodolle, A. (2008). Dynamic web services on a home service platform. In *Proc. 22nd Int. Conf. Advanced Information Networking and Applications*, pages 378–385.

Bricon-Souf, N., Anceaux, F., Bennani, N., Dufresne, E., and Watbled, L. (2005). A distributed coordination platform for home care: analysis, framework and prototype. *Int. J. of Medical Informatics*, 74(10):809–825.

European Commission (Jun. 2007). Ageing well in the information society - an i2010 initiative - action plan on information and communication technologies and ageing. Technical report, EU.

Gassner, K. and Conrad, M. (nstitute for Innovation and Technology (IIT), March 2010). ICT enabled independent living for elderly - A status-qua analysis on products and the research landscape in the field of Ambient Assisted Living (AAL) in EU-27.

Gray, P. D., McBryan, T., Hine, N., Martin, C. J., Gil, N., Wolters, M., Mayo, N., Turner, K. J., Docherty, L. S., Wang, F., and Kolberg, M. (2007). A scalable home care system infrastructure supporting domiciliary care. Technical report, Department of Computing Science and Mathematics University of Stirling.

IEEE (December 1998). IEEE guide for developing user organization open system environment (OSE) profiles, Portable Applications Standards Committee, IEEE Computer Society. IEEE Std 1003.23-1998.

IEEE (New York, NY, 1990). Institute of electrical and electronics engineers: Ieee standard computer dictionary: A compilation of ieee standard computer glossaries.

Johnston, S. (2005). Uml 2.0 profile for software services. *IBM developerWorks http://www. ibm. com/developerworks/rational/library/05/419_soa*.

Kosanke, K. (2006). Iso standards for interoperability: a comparison. *Interoperability of Enterprise Software and Applications*, pages 55–64.

Loniewski, G., Ramon, E., Walderhaug, S., Martinez Franco, S., Cubillos Esteve, J., and Marco, E. (2009). Data Management in an Intelligent Environment for Cognitive Disabled and Elderly People. *Electronic Healthcare*, pages 50–57.

McBryan, T., McGee-Lennon, M. R., and Gray, P. (2008). An integrated approach to supporting interaction evolution in home care systems. In *PETRA '08: Proceedings of the 1st Int. Conf. on Pervasive Technologies Related to Assistive Environments*, pages 1–8, New York, NY, USA. ACM.

Mikalsen, M., Hanke, S., Fuxreiter, T., Walderhaug, S., Wienhofen, L., and Trondheim, N. (2009). Interoperability services in the MPOWER Ambient Assisted Living platform. *Stud Health Technol Inform*, 150:366–70.

Mori, G., Paterno, F., and Santoro, C. (2004). Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Transactions on Software Engineering*, 30(8):507–520.

Nee, O., Hein, A., Gorath, T., Hulsmann, N., Laleci, G. B., Yuksel, M., Olduz, M., Tasyurt, I., Orhan, U., Dogac, A., Fruntelata, A., Ghiorghe, S., and Ludwig, R. (2008). Saphire: intelligent healthcare monitoring based on semantic interoperability platform: pilot applications. *IET Communications*, 2(2):192–201.

Rao, J. and Su, X. (2005). A survey of automated web service composition methods. In Cardoso, J. and Sheth, A., editors, *Semantic Web Services and Web Process Composition*, volume 3387 of *LNCS*, pages 43–54. Springer Berlin / Heidelberg.