

# Experiences from model-driven development of homecare services: UML profiles and domain models

Ståle Walderhaug<sup>1+2</sup>, Erlend Stav<sup>1</sup> and Marius Mikalsen<sup>1</sup>

{stale.walderhaug | erlend.stav | marius.mikalsen}@sintef.no,

<sup>1</sup>SINTEF ICT, SP Andersens vei 15b, N-7465 Trondheim, Norway

<sup>2</sup>Department of Computer Science, University of Tromsø, N-9000 Tromsø, Norway

**Abstract.** Model-driven development approaches such as OMG's Model Driven Architecture (MDA) have been proposed as the new paradigm for software development. However, the adoption of MDA is still low, partly because of the general-purpose modelling language being used. Domain specific modelling languages are being developed for technological and industrial domains to improve the expressiveness and effect of model-driven development techniques. The healthcare domain could benefit from these methodologies to improve development speed and software quality. In order to incorporate domain knowledge in a MDA process, information about workflows, artefacts and actors can be formalized in a UML profile and applied by MDA tools for design and development. This paper presents the results from the work done on model-driven development of smart homecare services in the MPOWER project. Following an iterative approach, two UML profiles to support development of Service Oriented Architecture based homecare applications are proposed. The profiles are based on a comprehensive domain investigation and best practice methods for domain specific modelling language development. Using homecare specific UML profiles indicate an improvement in the process for model-driven development of homecare services, and more evaluation will be conducted in an experiment in 2009.

## Introduction

Model-Driven Development (MDD) such as OMG's Model Driven Architecture (MDA) [1], has the potential to improve the quality of software systems. Quality attributes such as interoperability, reusability and appropriateness of software components and systems are main features of MDA.. By using abstraction and advanced automation techniques, software artefacts are created from formal models that are represented using languages such as the Unified Modeling Language (UML) [2]. The core of the MDA process, and similar MDD processes, is to use formal models as the main development artefacts in the entire development process, from domain analysis to implementation, deployment and testing.

Domain specific modelling languages (DSML) have been proposed as a means to overcome many of the shortcomings with UML and MDA. The scientific knowledge about applying MDD techniques in design and development of healthcare information

systems is scarce [3]. Creating DSMLs for the healthcare domain is a daunting task, and requires extensive investment of resources and time.

We set out to investigate how MDD with DSML support should be introduced and applied in a healthcare sub domain. In the MPOWER project [4], we have developed a framework for creating homecare software services using a model-driven approach. The framework defines a MDA toolchain which is a set of modelling, transformation and development tools that supports the complete MDA process as described in the MDA Guide [1]. A comprehensive model of actors and services in homecare along with the MDA toolchain for designing and implementing these domain specific web services has been developed and evaluated.

This paper presents research results from the project with focus on:

1. What is the domain knowledge in homecare that can be used as assets in the MDA process?
2. Which knowledge can be included in a UML Profile for homecare services and how can this knowledge be utilized by developers?

The MPOWER toolchain, providing model traceability, model transformation and code generation, has been evaluated in the development of two proof-of-concept applications and is currently being redesigned with improved UML Profile support for the domain. Based on the experience from developing the MPOWER framework and proof-of-concept applications a conceptual domain model and UML profile for service oriented computing in the homecare domain is proposed and discussed.

The rest of this paper is organized as follows. In the next section we describe the background for this work, including relations to and motivations for applying model driven development, domain specific modelling languages, and service oriented architecture. With this background we proceed to describe the method applied and main activities within the MPOWER project. The main results from each of the main activities are presented next, including conceptual domain models and our preliminary DSML approach based on two UML profiles. A discussion section follows this, before we conclude the paper.

## **Background and Related work**

The work presented herein is a part of the EU-IST project MPOWER (contract no. 034707) and of an ongoing PhD thesis work by the main author. MPOWER is a user driven research and development project where the main goal is to create a framework for rapidly creating standards-based homecare services. The framework includes the definition of a toolchain which is being used in the development of two proof-of-concept applications targeting elderly and cognitive impaired people living at home.

MDD promises a potential to improve the quality of software systems and their development by using formal models as first class entities in the entire development process. When MDD is done properly, the result may include the following improvements in the development process:

- Domain experts, system architects and developers can discuss concepts and requirements more effectively because care workflow and concepts are modelled formally.
- Formal modelling of domain concepts allows for more precise understanding of the target system's requirements, environment and use scenarios, which will increase the probability for developing a software system that fulfils its intentions
- Analysis of system behaviour and performance can be conducted in the design phase, facilitating rapid feedback and agile development processes.
- Keeping models independent of implementation technology and deployment platforms as long as possible allows for reuse of software models, thus improving interoperability and reducing development cost.

Tuomainen et al argues that modelling helps the understanding of healthcare activities by being illustrative, identifying improvements, simulate organisational processes and individual activities in healthcare [5]. They compare three model centric approaches; MDA, Business Process Modelling with BPMN and BPEL and the HL7 development framework. They conclude that in order to realise their full potential these approaches require local and project specific adaptation. This paper explains such an adaptation for the homecare domain.

## Methods

The main objective of the MPOWER project is to create a framework that facilitates rapid development of homecare services. To achieve this, it is fundamental to acquire knowledge about the homecare domain, and make this domain knowledge available to actors involved in the system development processes. Due to the complexity of the healthcare domain, it was considered imperative to iterate between domain modelling and system design. To facilitate this interaction, the MPOWER project defined three main activities:

1. Capture domain knowledge from experts on aging/dementia, healthcare workers in the domain, family carers and patients. The result will be a number of conceptual domain models defining the actors, actor relationships, core use cases and main system requirements / services of the homecare domain
2. Specify a MDA toolchain that support documentation of system requirements, modelling of design and development of services. Moreover, the toolchain must be evaluated in terms of usability and usefulness/performance by implementing two Proof-of-Concept Applications (PoCA). The results are a MDA toolchain with evaluation reports on developer acceptance and technical qualities
3. Design a DSML that incorporates the domain knowledge from task 1 and MDA toolchain experience from task 2. The result will be one or more UML profiles that can be used with a revised MDA toolchain.

The main activities and the expected results are illustrated in figure 1 and described in more detail in the following.

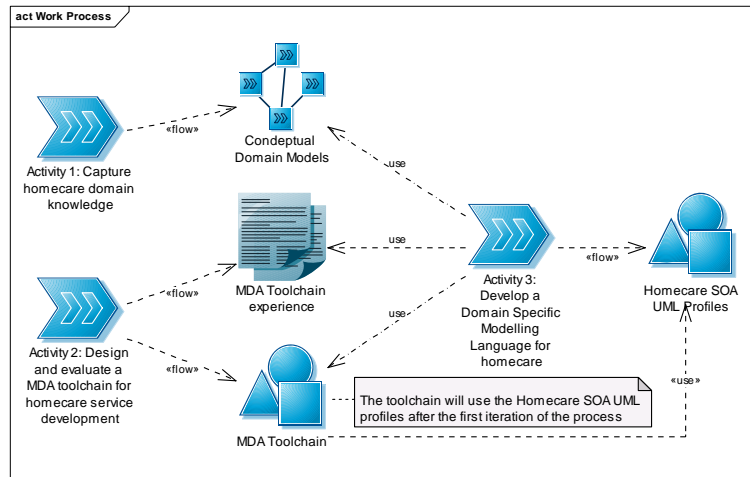


Fig. 1. The three main project activities and their work products

### Activity 1: Capture Domain Knowledge

The MPOWER project focuses on *smart homecare solutions for elderly and cognitive impaired people*. The domain models for the work being presented in this paper can be seen from two different viewpoints:

1. The Homecare viewpoint: this viewpoint focuses on organizational aspects of homecare as well as the main stakeholders (people and systems) involved. This model is the result of a comprehensive process involving a total of 140 domain stakeholders such as domain experts, professional caregivers, family carers and patients [6, 7].
2. The Homecare SOA viewpoint: this viewpoint focuses on the main system components and their relationships in terms of the principles of Service Oriented Architecture design. Important assets for this model are the design principles given by Erl [8], and SOA4HL7 methodology [9]. The structure and semantics of the domain model is supported by the SOA reference architecture from IBM along with the IBM UML profile for software services [10].

### Activity 2: Designing a Toolchain for MDD in Homecare

To have a formal way of specifying the domain models, proper modelling tools are needed. In the beginning of the MPOWER project, a set of tools were specified as the MPOWER toolchain to be used by all involved personnel for conceptual modelling, requirements specification, analysis, system design, system development, deployment and testing. A single toolchain for all involved stakeholders ensures that information provided by domain experts is readily available for system designers using the same toolchain. The process used for selection of tools matches the recommendation given by Staron [11], page 240: “The process [of creating domain models] should be tool

independent. The independence should be supported by using technologies that are open and unbounded, but at the same time supported by more than one tool.” The following tools were used:

- Microsoft Word 2003: Used for describing user scenarios. MS Word was available to all project participants involved in the process
- Enterprise Architect (EA) V6.5 from Sparx Systems: Used for UML modelling of use cases and services, model transformation and code generation of WSDL code.
- IBM’s UML 2.0 profile for Software Services: Used during modelling the services to structure models and stereotype core elements. Available from IBM [10].
- NetBeans V6.0: Used as Java IDE for generating service skeletons and implementing the services. Open source using open standards, <http://www.netbeans.org>

The described toolchain was used from the start of the project with only minor modifications such as EA upgrades and bug fixes. The two PoCAs were developed using the toolchain and the performance of the toolchain, were investigated from two perspectives: 1) Developer acceptance of the MPOWER toolchain: using the Technology Acceptance Model with two additional factors, as reported in [12], and 2) Technical review: Weekly scrum and quarterly technical meetings with workshop sessions on how to improve the toolchain.

### **Activity 3: Refine the MPOWER toolchain and develop a DSML**

The UML standard allows for the creation of a DSML in two ways: 1) Creating a new language based on Meta Object Facility [13], or 2) extending UML through the use of UML Profiles. As discussed by Selic, the latter will often be the most practical and cost-effective solution [14], and is also the chosen method for this paper. By using UML profiles to create a DSML, the semantics and syntax of UML can be inherited, and powerful UML/MDA tools can be used with the profile for software development.

The UML Profile standard [2], outlines several reasons for creating a DSML from UML. The most pertinent reasons for the challenges addressed in this paper are:

- Terminology adapted to the healthcare domain
- Add information that can be used during transformation
- Add constraints that restrict the way you can use the metamodel

Despite the fact that DSMLs are becoming more and more popular in systems modelling, there is not much knowledge in the scientific community about best practices for creating DSMLs with UML [14, 15]. However, in a paper from 2007, Selic summarized the basic steps for creating a DSML in terms of UML in [14]:

1. Create a conceptual domain model: The model should include the essential concepts of the domain, the relationships between the concepts, the constraints that govern the use of the concepts. In addition, syntax and semantics of the notation should be provided. A selection of UML models from Activity 1 makes up the conceptual domain model.

2. Map domain model to a UML profile, refining the core UML specification with stereotypes, tagged values and constraints. In addition, a library of domain specific modelling elements (model library) can be defined for reuse.

The process of creating a domain specific UML profile is not straightforward, since the level of abstraction and the intended use of the profile play an important role for the definition of the profile elements. This challenge is tackled with experience from the design of the MPOWER toolchain and development of two MPOWER PoCAs for the homecare domain.

To identify and model the elements of a UML profile is an iterative process. To guide this process, the Staron's guidelines for defining good stereotypes using a classification schema [11], is used. This classification is based on the stereotype's role and expressiveness.

## Results

The results presented in this section are based on the work carried out in the MPOWER project from October 2006 to June 2008.

### Activity 1: Conceptual Domain models

The domain models were developed in several iterations from October 2006 to September 2007. Details about the process and findings from the user needs investigation from which the domain models are derived can be found in [6].

Figure 2 shows the main concepts from a homecare viewpoint. To keep the model at an abstract level and not overpopulate it with unnecessary details, most attributes on the classes are hidden. The main classes and relationships are:

- **Subject of Care (SoC):** This is the person receiving care through a homecare program. The SoC has a unique identifier that is managed by the assigned healthcare organization. A SoC must be associated with at least one healthcare professional.
- **Homecare Program:** a class comprising the services, devices and healthcare organizations involved in providing homecare service to a SoC.
- **Carer:** an individual that is a part of the family, a healthcare professional or a friend. All HealthcareProfessionals must be associated with a HealthcareOrganization.
- **Healthcare Organization:** an organisation that is directly involved in the provision of care to a SoC.
- **Homecare Service:** a service provided to the SoC through a Homecare program. Three core types of services have been identified: information service (e.g., calendar, educational material access, medication list), communication services (e.g. SMS, email), and assistive service (e.g. indoor location service, heating control, burglar alarm, oven control).

Concepts in the model are aligned with the concepts presented in Continuity of Care (CONTSYS) standard from CEN TC251 [16], and service categories from [17]. Most concepts are also available in the HL7 RIM, but CONTSYS is more specific than HL7. These resources were found useful in selecting an appropriate abstraction level and structure in the domain model. The complete models of actors and services are presented in [7].

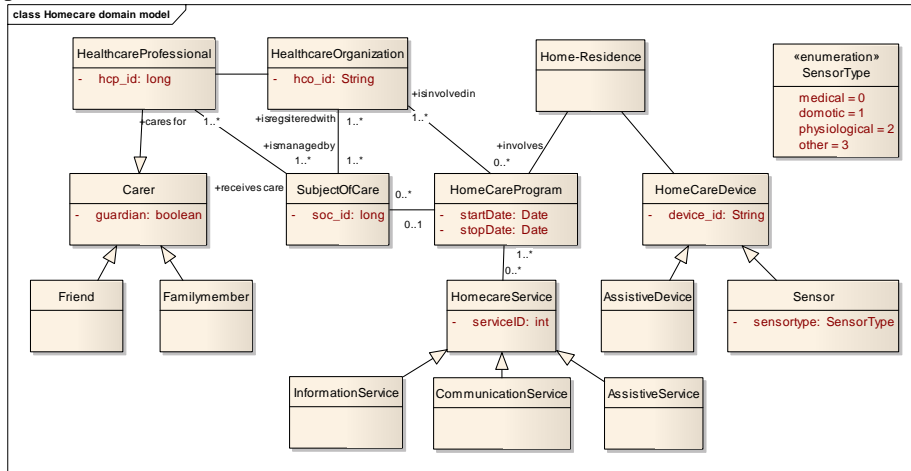


Fig. 2. Diagram showing the main concepts in a smart homecare domain

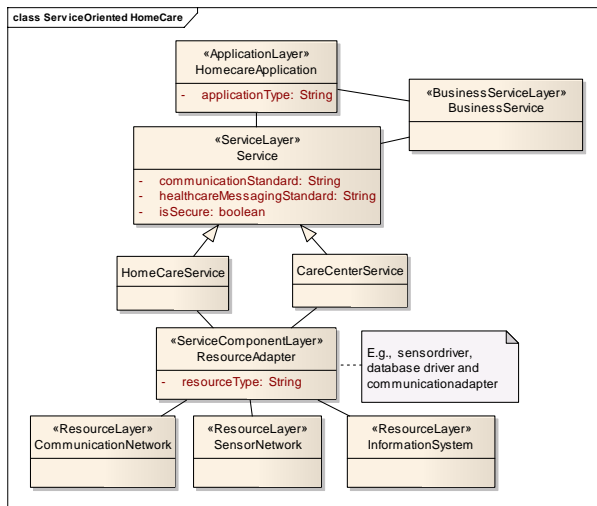


Fig. 3. The Service-oriented view on a typical homecare environment

Figure 3 shows the main components, stereotyped with the five layers of the IBM SOA reference architecture.

### **Activity 2: The MPOWER Toolchain**

The experiences from using the described toolchain for development of the services that form the PoCAs are grouped into developers' subjective experience and technical experience. The first group entails perceived characteristics such as the factors described by the Technology Acceptance Model [18]. The results from a developer evaluation of the MPOWER toolchain is presented in [12], and concludes that perceived ease of use and perceived usefulness are factors that affect the developers' adoption of MDA. It was also found that traceability between artefacts in the development process was useful. A major drawback with the evaluated toolchain was found to be the incomplete code generation features. A technical review of the MPOWER toolchain revealed that 1) WSDL model transformation incomplete: it was necessary to customize the transformation template for WSDL models, 2) WSDL code generation had errors: the built-in transformations in the EA tool generated some errors that had to be changed manually, e.g. using "type" references instead of "element" references in message definitions, and 3) Performance problems using HL7: the import of HL7 message types into WSDL resulted in tool crashes because of memory allocation problems. Recursive import of HL7 xml schema (xsd) definitions were not handled by the WSImport tools in Netbeans.

From the experience with the MPOWER toolchain, a set of new features were proposed. The developers would like more support in generating the implementation of the services – repetitive code (e.g. for DB management, handling of security, return status), and support for object/relational persistence service, such as generation of Hibernate mappings for the information elements declared in the message definitions of the WSDLs. These features may impact the design of a DSML as they could require domain specific information to be incorporated into the models during the service design.

### **Activity 3: Refined Toolchain - Mapping of Domain Concepts to DSML - UML Profile**

The process of defining a UML Profile for SOA in homecare use concepts from the conceptual domain models and experience from toolchain and PoCA development in an iterative approach. This section presents the preliminary results from Activity 3 after the first iteration (January-June 2008). Activity 3 is carried out by a core team of three researchers. However, some discussions are brought into the development team mailing list (16 developers). The profiles were updated in three main revisions: initial version for the start of the development, second version after first version of services and the third version after the first iteration of the PoCA development. The changes between version one and two were significant, whereas only tags and minor adjustments to relationships were done for version 3.

Two UML profiles are proposed, the Homecare UML Profile and the SOA Homecare UML Profile. The profiles can be used separately or together in a MDA development project, depending on how the profile elements are utilized in the development process.

### Homecare UML profile

Figure 4 shows the Homecare UML profile. The profile elements are mainly derived from the homecare conceptual model (figure 2). The tagged values on the stereotypes were identified based on experience from the MDA toolchain work. The mapping of concepts to the UML profile was also guided by the CONTSYS standard [16].

**Table 1.** Table describing the proposed stereotypes in and tagged values in the Homecare UML profile

<b>Stereotype</b>	<b>Comment</b>
SubjectOfCare	Subject of care (SoC) is defined in CONTSYS as "person seeking to receive, receiving, or having received health care" [16]. Used to decorate SoC modelling elements and to add information about the SoC that can be used during model transformation or code generation, e.g., to generate interface classes for EHR systems and individual plan systems. <i>SoC_type</i> : describes can be used to describe different types of SoC according to national or healthcare enterprise specific patient classifications.
Carer	A stereotype that should be used on all modelling elements representing an individual that provides care, professionals as well as non-professional caregivers. <i>roles</i> : can be used to set the default role, e.g. in terms of security, for the instances of classes marked with this stereotype.
Healthcare Professional (HcP)	Defined in CONTSYS as "person authorised by a nationally defined mechanism to be involved in the direct provision of certain health care activities" [16]. Should be used to mark all modelling elements of type Class/Actor that are representing individuals that fit this definition. The roles attribute is inherited from Carer.
Other Carer	Defined in CONTSYS as "person providing assistance for activities of daily living or social support". This stereotype should be used to mark modelling elements of type Class/Actor that are representing individuals such as family members, friends and other carers employed by non-healthcare organizations such as home services and security services. The roles attribute is inherited from Carer
Healthcare Organization (HcO)	Defined in CONTSYS as "organisation involved in the direct provision of health care" [16]. This stereotype should be used to mark all modelling elements of type Class/Actor that represents organisations that fits the CONTSYS definition. <i>orgainsationType</i> : is used to describe the type of organisation according to speciality levels, private versus public or other national classifications.
Homecare Device	Generic homecare device stereotype to be used on modelling elements of type Class/Actor. The stereotype can be useful for design-time checking of interoperability and interconnectivity of devices in a homecare system. <i>deviceType</i> : describes the type of this device, e.g. whether it is

	<p>medical or domotic.</p> <p><i>interfaceType</i>: describes the kind of interface used to connect to this device. In the UML profile, the <i>HomecareDeviceInterface</i> enumeration is defined based on the experience in the MPOWER project. This enumeration includes the most used interface types, and can be refined to fit other technologies.</p>
Healthcare Professional For SubjectOfCare	<p>A stereotype that is used to mark an association between a <i>HealthcareProfessional</i> and a <i>SubjectOfCare</i>. The stereotype can be used to ensure that a <i>SubjectOfCare</i> is associated with at least one <i>HealthcareProfessional</i>.</p>
EmployedAt	<p>A stereotype that is used to mark an association between a <i>HealthcareProfessional</i> and a <i>HealthcareOrganisation</i>. The stereotype can be used to check that all <i>HealthcareProfessional</i> “types” are employed at a <i>HealthcareOrganisation</i> “type”.</p>

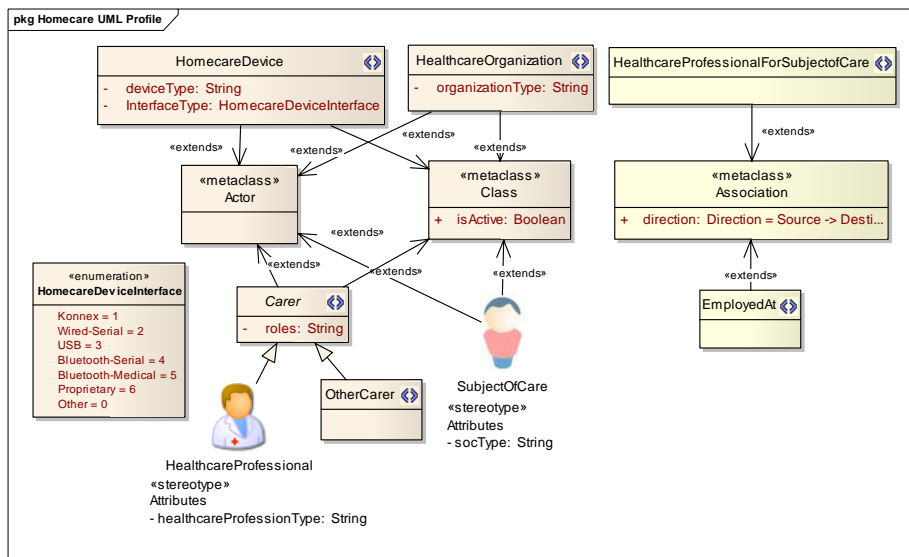


Fig. 4. First version of Homecare UML Profile

All the stereotypes in the Homecare UML Profile falls into the category “Virtual Metamodel Extension, restrictive” defined by Staron in [11]. These are stereotypes that reuse the semantics of the metaclasses (e.g. Actor and Class). Often the must be used with other stereotypes, making the stereotyped model element more precise and may also add a new icon to the concrete syntax to familiarize the model presentation (e.g. icons on *HealthcareProfessional* and *SubjectOfCare*).

### SOA Homecare UML Profile

The SOA Homecare UML profile enables developers to create precise models of SOA-based homecare systems.

**Table 2.** List of stereotypes and tagged values in the SOA Homecare UML Profile

Stereotype	Comment
Homecare Application	<p>A stereotype used to mark modelling elements of type Component and Class. The stereotype adds two properties to the modelling elements:</p> <p><i>securityLevel</i>: this describes the security level of the application. This can be used to check that a user (i.e., service, component, application) of the application must have at least the same access level in order to be allowed to use the service. For service composition, this information can be used to check that the application can access security-enabled services.</p> <p><i>applicationType</i>: this describe the type of application this is, e.g. in terms of deployment configurations.</p>
Homecare Service	<p>A service which is used in the homecare environment. The IBM Service Profile should be used in combination with this stereotype.</p> <p><i>securityLevel</i>: this describes the security level of the service. This allows for specification of the security requirements and rights for a modelling element that can be utilized during code generation. In the next version of the UML Profile, this stereotype will be updated with more security tags addressing service-service authorization and information encryption.</p>
Homecare Message	<p>To denote elements that are messages used in interactions of homecare services and applications.</p> <p><i>isPersistent</i>: indicates whether the message data is stored in a database or not. This can be used for creating Hibernate mapping code and database schema.</p> <p><i>messagingStandard</i>: the standard to which this message belongs. Can be used both for code generation, ensuring correct libraries are present, and for checking conformance with the standard.</p>
Assistive Service	<p>A type of homecare service that provides assistive functionality in the homecare system. Derived from Stefanov's classification for smart house services for elderly and cognitive disabled [17].</p>
Information Service	<p>An information service which will be used by stakeholders in a homecare setting. The service can be medically related, but can also be other services e.g. used for social interaction by the subject of care.</p> <p><i>serviceType</i>: defines the type of information service this service belongs to. An enumeration is proposed based on the experience in the MPOWER project. This enumeration can be refined in other projects using other kinds of information sources.</p>
UsesHomecare Message	<p>A stereotype that mark associations between a HomecareService and a HomecareMessage. The stereotype can be used to generate traceability information that can again be used when messaging standards are being updated or changed.</p>
UsesHomecare Service	<p>A stereotype that mark associations between a HomecareApplication and a HomecareService. The stereotype can be used to generate traceability information that again can be used when a homecare service is being updated or changed.</p>

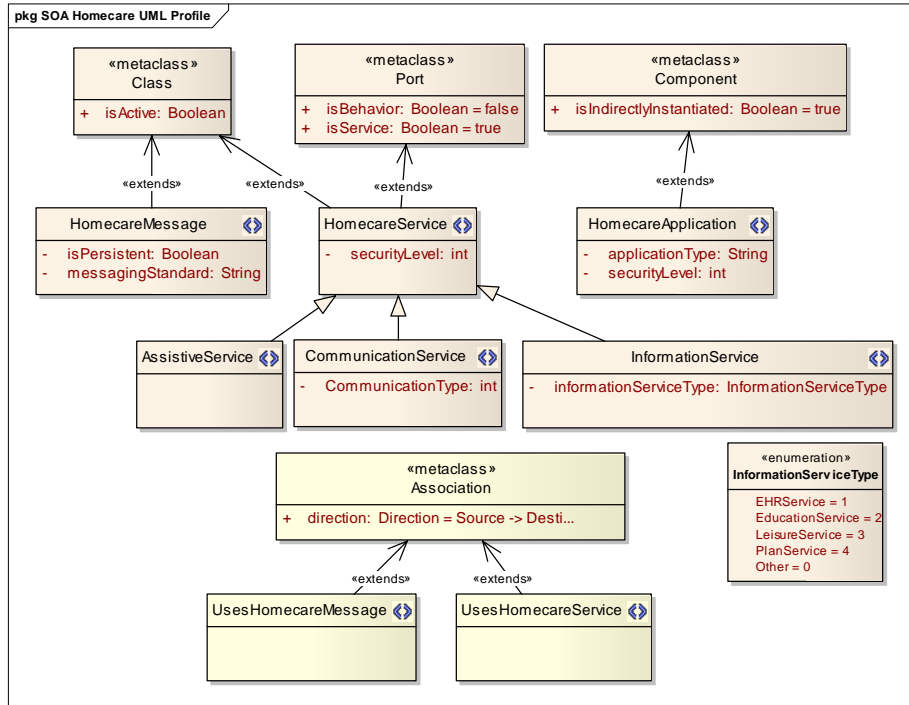


Fig. 5. The Homecare SOA UML Profile diagram

All stereotypes in the Homecare SOA UML profiles falls into the category “Code generation, restrictive” defined by Staron [11]. These are stereotypes that extend the base metaclass (e.g., Class and Port) with some properties to increase the precision of the semantics and restrict the usage.

## Discussion

The work presented in this article is a part of an initiative to develop a model-driven software development framework for healthcare, focusing on homecare services in the first phase. It is considered imperative to incorporate domain knowledge into the framework and make this knowledge readily available for architects and developers in all development phases. This paper presents the results from creating a domain specific modelling language for homecare using UML Profiles.

Capturing the conceptual domain knowledge for homecare, or any other healthcare sub-domain, is a daunting task. Many stakeholders are involved, as well as a plethora of information systems, involving many different coding standards and vocabularies. These factors, in addition to legislative factors and organisational aspects, make modelling of reusable healthcare domain models difficult, but not impossible. To succeed in creating a useful model-driven software development process, it is important to choose the right level of abstraction and in divide the healthcare application areas into well defined sub-domains.

The homecare domain model shown in figure 2 shows the most important actors and relationships between them. The model would fit for modelling most homecare solutions, is aligned with the CEN TC251 CONTSYS standard [16], and includes the main classifications from Stefanov's acknowledged paper on smart house technologies for elderly [17]. The model is the result of a comprehensive domain analysis process where 140 domain stakeholders from four European countries were involved in improving the validity of the model [6]. If new concepts are developed for the domain, these can be added as an extension to the existing homecare model, without compromising the original model and the related UML profiles.

The Homecare SOA model is based on the domain investigation from the MPOWER project, in addition to the IBM SOA reference model and IBM UML profile for software services [10]. The Homecare SOA model provides information about deployment of services and possible configuration and information sources. The model is on an abstract level, and could in certain cases be refined with details about security platforms and network connectivity details. Such domain knowledge could be useful in planning the distribution of services and integration with existing resources, but will also make the model less suitable for reuse across different healthcare enterprises and nations.

To make the domain knowledge readily available as assets in the development process, UML profiles were chosen, inline with the recommendations by Selic [14]. UML profiles builds upon the syntax and semantics of UML, and most UML tools support profiles. This is an imperative advantage, enabling developers to use their favourite UML tool for design and development. The process of selecting domain concepts to include as stereotypes, tagged values or constraints in a UML model, requires knowledge about model-driven development, but also experience from modelling systems in the domain in question. Experience from the development of a MDA toolchain (Activity 2) provided information about which target software artefacts that should be generated from the models and which models and diagrams that should be applied for achieving this. This information was of utmost importance when choosing the metaclass extensions for the elements in the UML profiles.

Model-driven development seeks to represent all development artefacts as modelling elements. UML profiles are used to customize the modelling language to include familiar concepts that enables more effective and precise system design and implementation. Stereotypes, the main UML profile mechanism, can be used for several purposes, as discussed by Staron [11]. The result from the mapping of domain concepts to a DSML (Activity 3) showed that all stereotypes in the Homecare UML profile are classified as Virtual Metamodel Extensions. This implies that this profile is mainly used to increase the expressiveness of the modelling language when designing systems for homecare. A "virtual model extension restrictive" stereotype, adds a domain specific icon such as a picture of a nurse to the modelling element, together with a well known domain specific label such as HealthcareProfessional.

The Homecare SOA UML Profile includes elements from the "Code generation, restrictive" category. This means that the main use of these stereotypes is to improve code generation by providing domain information so that code generation scripts can create high-quality code. In this paper code generation was restricted to WSDL and Hibernate code, but other software artefacts can be generated from the domain

information in a UML profile. Test cases, error-checking code, security policy verification, and result validation may be generated if the required information is available. Examples are:

- During design of a homecare service, mark the service class with HomecareService, add classes for the messages that will be used as input and output, mark these with HomecareMessage and set the tagged value MessagingStandard to the appropriate value. Create an association between the service and the message by applying the UsesHomecareMessage stereotype on the association. During code generation, the generator could look up messaging details (syntax, namespaces, etc) in an external resource and insert this code into e.g., the WSDL types definitions.
- When modelling the information models for a homecare system, mark the patient class with SubjectOfCare stereotype. To ensure that a patient is associated with a healthcare professional, a model validator based on OCL [19], can iterate through the patient class's relationships and look for a HealthcareProfessionalforSubjectOfCare stereotyped association.

The two proposed UML profiles can be used on the same models to provide different "views". In addition, the IBM Software Service UML profile should be used to complement the service design models for SOA Homecare systems.

## Concluding remarks

Model-driven development approaches can be improved by extending the modelling language with domain specific concepts. UML Profiles can be used as a mechanism for toolchains based on OMG's MDA and UML standards.

The UML Profiles must provide information that can improve the design and/or code generation processes. The two profiles proposed in this paper are based on solid work on capturing homecare domain knowledge and experience from developing homecare SOA systems using MDA. Though the profiles are still undergoing updates and improvements, they can improve modelling process performance and results.

Another finding from the process is that it is useful to have a development activity in parallel with specification of the conceptual domain and profile. The experience from this development activity gave valuable input to the mapping of concepts to DSML processes. This finding extends the proposed approaches to DSML development in [14, 15, 20].

The MPOWER Toolchain will be evaluated by university students in 2008. In 2009, an experiment measuring the subjective improvements (perceived characteristics) and objective improvements (e.g., reduction of errors, time spent for development) will be conducted with 20 developers from the healthcare domain.

## References

1. J. Miller and J. Mukerji, *MDA Guide Version 1.0.1*, omg/2003-06-01, J. Miller and J. Mukerji, Object Management Group (OMG), 2003.
2. Object Management Group (OMG), *UML 2.1.2 Superstructure and Infrastructure*, Object Management Group (OMG), 2007.
3. P. Mohagheghi and V. Dehlen, "Where Is the Proof?-A Review of Experiences from Applying MDE in Industry," *Model-Driven Architecture-Foundations and Applications: 4th European Conference, Ecmada-Fa 2008, Berlin, Germany, June 9-13, 2008, Proceedings*, 2008.
4. MPOWER Consortium, "Middleware platform for eMPOWERing cognitive disabled and elderly," 2006; <http://www.sintef.no/mpower>.
5. M. Tuomainen, et al., "Model-centric approaches for the development of health information systems," *Proc. Medinfo*, 2007, pp. 28-32.
6. T. Holthe, et al., "Analysing user needs prior to technology development for supporting older people and people with cognitive impairments at home. Experiences from the MPOWER-project," *International Journal of Medical Informatics*, vol. submitted July 2008, 2008.
7. S. Walderhaug, et al., "Reusing models of actors and services in smart homecare to improve sustainability," *Stud Health Technol Inform*, vol. 136, 2008, pp. 107-112.
8. T. Erl, *Service-Oriented Architecture Concepts, Technology, and Design*, Prentice Hall, 2006.
9. A. Honey and B. Lund, *Service Oriented Architecture and HL7 v3: Methodology*, HL7 Service Oriented Architecture Special Interest Group (SOA SIG), 2006.
10. S. Johnston, "UML 2.0 Profile for Software Services," April 13, 2005 2005; [http://www.ibm.com/developerworks/rational/library/05/419\\_soa/](http://www.ibm.com/developerworks/rational/library/05/419_soa/).
11. M. Staron, "Improving modeling with UML by stereotype-based language customization," School of Engineering, Blekinge Institute of Technology, Blekinge, 2005.
12. S. Walderhaug, et al., "Factors affecting developers' use of MDS in the Healthcare Domain: Evaluation from the MPOWER Project," *Proc. From code-centric to model-centric development, Workshop at European Conference on Model-Driven Architecture*, European Software Institute, 2008.
13. Object Management Group (OMG), *MOF 2.0 / XMI Mapping Specification, v2.1*, 2005.
14. B. Selic, "A Systematic Approach to Domain-Specific Language Design Using UML," *10th IEEE ISORC*, vol. 7, 2007.
15. F. Lagarde, et al., "Improving uml profile design practices by leveraging conceptual domain models," *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, 2007, pp. 445-448.
16. CEN TC251, *EN 13940-1: Health Informatics - System of Concepts to Support Continuity of Care - Part 1: Basic Concepts*, European Committee for Standardization, 2006.
17. D.H. Stefanov, et al., "The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 12, no. 2, 2004, pp. 228-250.
18. F.D. Davis, et al., "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science*, vol. 35, no. 8, 1989, pp. 982-1003.
19. Object Management Group (OMG), *Object Constraint Language (OCL), Version 2.0*, Object Management Group, 2006.
20. L. Fuentes-Fernández and A. Vallecillo-Moreno, "An Introduction to UML Profiles," *UML and Model Engineering*, vol. 5, no. 1, 2004.