Service Tailoring: A Method and Tool for User-centric Creation of Integrated IT-based Homecare Services to Support Independent Living of Elderly

Mohammad Zarifi Eslami

Enschede, The Netherlands, 2013

Ph.D. dissertation commitee:

Chairman and secretary:	Prof.dr.ir. A.J. Mouthaan (University of Twente, the Netherlands)
Promotor:	Prof.dr. R. J. Wieringa (University of Twente, the Netherlands)
Assistant Promotors:	Dr.ir. M. J. van Sinderen (University of Twente, the Netherlands)
Members:	Prof.dr.ir. H.J. Hermens (University of Twente, the Netherlands)
	Prof.dr.ing. D. Konstantas (University of Geneva, Switzerland)
	Prof.dr.ir. W. Kraaij (Radboud University Nijmegen, the Netherlands)
	Prof.dr. G.Q. Maguire Jr. (KTH - Royal Institute of Technology, Sweden)
	Prof.dr.ir. L.J.M. Nieuwenhuis (University of Twente, the Netherlands)

CTIT Ph.D. Thesis Series No. 13-246
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands

SIKS Dissertation Series No. 2013-13
This research has been carried out under the auspices of SIKS, the Dutch research School for Information and Knowledge Systems.

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.

Printed and bound by Ipskamp Drukkers B.V. The Netherlands
Cover designed by Hajir Gharib
ISSN 1381-3617
ISBN 978-90-365-3528-1
http://dx.doi.org/10.3990/1.9789036535281

SERVICE TAILORING:
A METHOD AND TOOL FOR USER-CENTRIC CREATION OF INTEGRATED IT-BASED HOMECARE SERVICES TO SUPPORT INDEPENDENT LIVING OF ELDERLY

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
Prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday the 07th of June 2013 at 12.45

by
Mohammad Zarifi Eslami

born on 27 July 1978
in Tabriz, Iran

Dit proefschrift is goedgekeurd door:
Prof.dr. R. J. Wieringa (promotor) en Dr.ir. M. J. van Sinderen (assistent-promotor)

تقدیم به

حسنیه و علی اکبر، مادرم و پدرم، که به من عشق ورزیدند

لیلا، همسرم، که عشق ورزیدن را به من آموخت

و

آراز، پسرم، که فرصت عشق ورزیدن را به من داد

Abstract

This thesis addresses the problem of supporting independent living of elderly people through IT-based homecare services. Independent living is seen as one way to deal with the consequences of an aging population (especially in industrialized countries), which include rising healthcare expenditures and a required shortage of healthcare professionals.

Our main goal is to improve the service creation process for the homecare domain, in terms of reducing the IT skills, time, and effort needed to create new services, while aligning the individual needs of service users and the functionality of the created services. We call our approach to service creation "service tailoring". Service tailoring, as proposed in this thesis, is a way of creating new services, and adapting previously created services, involving healthcare professionals (care-givers) in the creation process and targeting elderly people (care-receivers) as the primary users of the created services.

For requirements engineering, we started with a literature study of existing homecare systems and user-centric service personalization techniques, and a market survey of current homecare technologies and products. Then, we performed five series of interviews with care-givers in a care institution in the Netherlands[1]. This institution consists of residential blocks where elderly persons can live and receive care services round the clock from professional care-givers. The first, second, and third series of interviews were done before prototyping our approach, and the fourth and fifth series of interviews were performed to check the usability of the approach after the prototype was used by the care-givers and care-receivers. We performed the first series of interviews by using questionnaires to analyze the existing situation and to learn what types of services were expected from an IT-based homecare system. These interviews were helpful to gain insight into commonly performed tasks and how these tasks are performed. After designing our approach, we performed the second series of interview with the same care-givers to validate our design and refine it before implementing a prototype of our approach.

[1]http://www.orbisconcern.nl/

Use of new IT-based services can introduce new types of risks in the environment where these services are implemented. This is particularly true in the homecare domain, since people's wellbeing and health may depend on the services. Risks can possibly lead to unwanted or dangerous situations, and lack of trust in the services, e.g. due to multiple risk occurrences, may lead to a decline in the use of the IT-based services in practice. We proposed a Risk Driven Requirements Specification (RiDeRS) method to identify potential risks of using IT-based services, and to specify additional requirements on these services (and the underlying IT system) to mitigate or prevent these risks. To define RiDeRS, we performed a literature study of existing requirements elicitation methods which use risks to elicit risk-reducing requirements. Comparing RiDeRS with existing methods, what is new about RiDeRS is that it takes into account the properties of the system's environment more systematically than existing methods. In RiDeRS, we considered users' properties in addition to their goals to identify a list of possible risks and specified the requirements which can prevent or mitigate the risks. Accordingly, we performed risk assessment using RiDeRS by performing the third series of interviews with the same care-givers to identify and analyze potential risks of using our services and consequently to identify further requirements of the services and the underlying IT system.

After evaluating the result of the first three series of interviews, we designed the architecture of a service tailoring platform. To evaluate this architecture, we developed a prototype of the service tailoring platform as part of the U-Care[2] project, which was subsequently used in two series of experiments to validate the properties of the approach. The experiments were conducted in a near real-life setting at the care institution. A number of users (8 care-receivers and 4 care-givers) volunteered to use the U-Care system[3].

We assumed that service tailoring should be done by a care-giver. We identified different types of care-givers who interact with and help care-receivers in their daily life in the homecare domain. The identified care-givers include professional nurses, family members, informal care-givers (volunteer non-professional care-givers), occupational therapists, physiotherapists, physicians, pharmacists and psychologists. We found that a professional nurse, as a care-giver, is the one who should tailor the services, as care-receivers spend more time with professional nurses than other types of care-givers.

[2]http://www.utwente.nl/ewi/ucare/

[3]The U-Care system includes a tailoring platform, a provisioning platform, and some homecare applications

We aimed to achieve IT-transparency by using the concept of service building blocks (SBBs). The SBB concept is used to denote the smallest manageable unit of service functionality from the point of view of care-givers (e.g., a reminder SBB notifies care-receivers to do something). A SBB provides a generic service interface to be used by care-givers in the process of service tailoring. A SBB also provides a list of configuration parameters to allow a care-giver to specify different aspects of the SBB, such as service operations and user interface modalities. Each SBB represents a concrete service or alternative concrete services, abstracting away the technical details that are not relevant to service tailoring (i.e., irrelevant to the care-giver). The outcome of a service tailoring process is a so-called service plan, which represents a composite service tailored to the needs of a specific care-receiver as understood by the care-giver. A service plan contains sufficient information to allow the automated derivation of a complete implementation on a target execution platform. Designing such a service plan from scratch is a difficult and a time consuming task. We use the concept of a design pattern to simplify the process of creating a service plan. We make use of treatment patterns as a starting point for the tailoring process, where a treatment pattern is an activity structure for handling a generic homecare task (e.g., blood pressure monitoring task). Thus, the care-giver does not have to create a service plan from scratch, but rather selects the homecare task to be supported from a menu. The tailoring platform then presents the corresponding treatment pattern as the initial service plan, which should be further refined and completed by the care-giver.

The service tailoring platform is responsible for enhancing the creation and tailoring of the service plans by providing a graphical user interface (GUI) to the care-givers. To show the feasibility of the proposed architecture, we developed a prototype of the tailoring platform. The prototype was evaluated in two series of experimental field tests (with a total duration of 4 months). After the first series of experiments, we performed a fourth series of interviews with the care-givers, evaluated the results and improved the system. After the second series of the experiments during which the users used the improved system, we performed the last series of interviews with the users to evaluate the usability and usefulness of the service tailoring platform. We also asked their opinion of the system in order to see if such a system could be used in practice and indeed if it could save care-givers time and could increase the quality of life of the care-receivers. Finally, we reflect on the work done and the results achieved in the context of the homecare domain, and subsequently discuss whether the proposed approach can be generalized for use in other homecare applications.

Acknowledgements

This thesis project has taken placed at Information Systems (IS) research group in the faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) at the University of Twente (UT). During the four years PhD research, I have experienced one of the most memorable times in my life. Working in the warm and friendly environment of IS group, I feel home living with the IS family.

I will start with the mother of this family: Suse. I will never forget the day I came to the Netherlands. It was Friday evening and you were there with your beautiful smile to take me and my family with your own car to the house, which you had rented for us. You also took us in the weekend to buy some furniture. Your unsparing support and kindness never stopped. You are an angel walking on the earth and the IS family is lucky to have you.

The father of this family, Roel, who manage this group not as a boss, but truly as a father. I am really proud of being your student and enjoyed working in your group. As my promoter, in addition to scientific aspects, I have learned two important things, which I will remember them in my life: (a) your positive attitude: when I send you a piece of work, no matter which quality it had, you always enumerated its positive aspects first. This was really motivating to work harder and go further. After that was a “However” followed by your constructive comments. (b) You always keep your promises no matter how busy you are. This also motivates your students to prepare the deliverables on time and keep their promises.

Marten, I really have felt that you were my friend rather than a formal supervisor. The door of your office was always open for me and you had always time for me. You were always supportive. Your support was not limited only at work. My friends back home may not believe that my supervisor helped me with his own car to take furniture to our house.

Shahin, you are truly a brother for me. Without doubt, this work could not reach such useful and interesting practical outputs without you. And now, I can not imagine living in the Netherlands without you and your lovely family, Soude and Ava. I am now looking forward and counting the days to see your PhD dissertation and be in your PhD defense which is nearby.

Hassan, you are truly a friend in-need. I always come with the questions about my work and you kindly share your thoughts with me and helped me in writing the papers. I would really want to put your name as a co-author in some of my papers, but you asked me not to do so. I really enjoyed all those times which we spent together specially our workouts at the gym of the university.

Brahmananda, you read, edit and provide your constructive comments almost in my whole thesis. This acknowledgment section is probably the only section of my thesis which you have not read. I really enjoyed all those times that we walked to/from university and you chill me out with your calm heart.

I would like to extend my gratitude to all my present and past colleagues from IS group: Andre, Camlon, Chen, Eelco, Klaas, Lianne, Luis, Maya, Nelly, Pascal, Sergio, Silja, Steven, Wilco, Wolter, and Zornitza. Thank you all for making IS group a pleasant working place.

I am also deeply indebted to Eelco and Andrea who helped me in designing and writing RiDeRS method. The related work of Chapter 4 (and its Tables *4-3*, *4-4* and *4-5*) is mainly Andrea's work.

I would like to thank the members of my defense committee and in particular Gerald Maguire for his meticulous reading from cover-to-cover. Your comments and feedbacks polished this dissertation.

This work is part of the IOP GenCom U-Care project (http://ucare.ewi.utwente.nl), sponsored by the Dutch Ministry of Economic Affairs. I would like to extend my gratitude to all its partners. Specifically I would like to thank, other two PhD students and their supervisors: Jan-Willem and Bert-Jan, and Lucas and Maria; Mobihealth and particularly Tom; and Orbis and particularly Cindy, John, Lilian, Maarten and Milou.

I would also like to thank all my friends in the Netherlands: Afshin and Elham, Alireza, Amirhosein, Ashiful, Barbara and Bastiaan, Catherine and Kees, Eduardo, Hajir, Henry, Majid, Mehrdad, Meisam, Michel, Mohammadreza, Mohsen, Nicolas, Parviz and Marian, Ida, Jan-Pieter, Sina and Selda, Victor, and UT-Kring Futsal team.

The last but not least my family. I would like to specially thank Leila, my wife (my life), Araz, my son (my sun), my parents Hosniyeh and Ali-Akbar, my sister Maryam and my brother Amir.

Mohammad Zarifi Eslami
Amstelveen, June 2013

Contents

Chapter 1

Introduction

"Become a student of change. It is the only thing that will remain constant."
— Anthony J. D'Angelo

The increasing percentage of elderly people puts healthcare services in industrialized countries under great pressure. Moreover, elderly people prefer to live in their own home and live an independent life as much and as long as possible. However, supporting independent living at home would be difficult because of the increasing gap between the number of elderly in need of care and the number of available care-givers able to provide homecare. Providing IT-based care solutions to elderly in their home is one of the means to close this gap. There are various IT services which are already available and can support elderly in their homes, such as remote blood pressure monitoring and medicine dispensing services. However, most of these services are designed without considering the individual needs and preferences of care-receivers. Personalized health services should increase the quality of care, and thereby the quality of independent living, and this in turn would help in the wider diffusion of homecare services at lower cost. In this thesis, we propose an approach for personalizing IT-based homecare services by introducing a tailoring process and architecture.

This chapter introduces the topic of this research and discusses the objectives, research questions, scope, and research approach of the work. Section 1.1 discusses the background information to establish the context of this thesis. Section 1.2 presents an application scenario that illustrates the motivation of this work. Section 1.3 outlines the main objectives of this thesis, and Section 1.4 elaborates research questions to be addressed in order to achieve those objectives. Section 1.5 describes the research methods used in this thesis and Section 1.6 presents the scope of the thesis. Finally, Section 1.7 presents the structure of the rest of this thesis.

1.1 Background

Nowadays, the aging population issue has received increasing attention, especially in industrialized countries. For example, in the Netherlands, in 2003, 14% of the Dutch population was over 65 years of age and as of 2030, this number is expected to raise up to 22% [148]. Therefore, in the near future, the industrialized world will face an aging population where one of its consequences is lack of qualified staff to support elderly people [55]. Moreover, elderly people prefer to live at home and live an independent life as much and as long as possible [112]. Currently care services and in general, any health services for elderly (care-receivers) are provided manually by qualified healthcare staff (care-givers). It is expected that with an aging population, there will be a shortage of qualified health personnel able to provide homecare services.

Due to the aging population and the expected shortage of professional care-givers, an IT-based system could play an important role in providing care services. The use of IT services for healthcare provisioning at home can have several benefits [161, 92] such as:

- Improving the quality of care (monitoring the elderly continuously 24/7 in their preferred familiar environment).
- Improving the quality of life of the elderly by facilitating and prolonging independent living (often in a unintrusive way and without needing a care-giver to measure their vital signs).
- Saving the time of healthcare professionals by providing more efficient healthcare solutions and unburdening institutionalized healthcare.

Today, by exploiting technologies (such as sensor technology, fast and easy communication, and information technology) a lot of IT services which support the elderly at home have been developed including health monitoring, event-based alarms, and automated analysis to communicate health-related information [92, 53, 178, 107, 120, 13, 6]. Although some homecare services have been proven in practice, and several promising prototypes have been developed in research projects, many technological challenges are still ahead [174]. Several technological challenges concerning homecare applications have been previously studied, but our focus is to address one of them, which is the uniqueness of each care-receiver and his/her context.

Our proposed solution will be to apply a service-tailoring approach. The current homecare systems are generally 'stand-alone' systems for specific diseases and assume a 'standard' patient. Most of these systems were designed without considering the individual preferences, needs, and situations of care-receivers [105]. However, in reality, each care-receiver is unique in the way (s)he experiences or is affected by a disease, or a combination of diseases, not only because of his/her mental and physical conditions, but also because of his/her social and physical environment. Services designed for the 'standard' user may not be suitable for real users. This is particularly true for elderly persons in their home, since they are subject to different physical and mental limitations with respect to using services [47]. This calls for the provisioning of personalized services, according to the individual needs of each care-receiver.

For homecare systems, especially for elderly people, we considered the following motivations for tailorability:

1. *Personalization*: different individuals have different needs, preferences', and situations (from now on, we use term '*needs*' instead of needs, preferences, and situations) with respect to monitoring and support functions. These requirements for personalization reflect what the user wants, needs, and likes; all of which may depend on the context at hand (for example, location, user activity, physical characteristics of the environment, available resources, and people nearby). Such requirements may not be known prior to deployment and are different for different people.

2. *Dynamics*: health problems of individuals change over time (in the case of elderly people, health problems normally increase), and therefore needs change accordingly. For this reason services have to operate in a constantly evolving environment of people, content, electronic devices, and legacy systems [38].

3. *User-system interaction*: even if the health problems of an individual care-receiver would not change over time, service (re-)tailoring might still be needed to improve the system support. Improvement may stem from the fact that the care-giver acquires a better understanding over time of the health problems of the care-receiver or of the context in which the system should provide support, or from the fact that the care-receiver has become familiar with the system and is ready for more advanced support.

For all the reasons above, the application functionality provided to users as services should (1) be aligned with the uniqueness of each user's needs, (2) evolve with changes in these needs, and (3) take the dynamic context of the user into account. Ideally this would call for tailor-made services; however, developing such services from scratch for each care-receiver would be economically and technically infeasible. Personalization of services based on dedicated design, implementation, and deployment of software and hardware is not feasible because of the cost and time involved for doing so. It is economically infeasible to develop personalized homecare systems for each individual patient. So, homecare systems should provide a set of patient-neutral healthcare-related functions which can be configured and composed according to the needs of each individual patient (tailorability).

For simplicity and brevity, we adapt the following convention throughout the thesis. To refer to care-givers we use feminine pronouns whereas to refer to care-receivers we use masculine pronouns. Moreover, all defined terms and concepts, which are introduced, specialized, or used, is described in the glossary at the end of the thesis.

1.2 Application Scenario

We use the following homecare application scenario to motivate the work presented in this thesis and to clarify our discussion:

"John and Mary are 84 and 82 years old, respectively. They live together in a special home for the elderly. They have different needs with respect to care. They also suffer from several comorbidities, i.e., presence of one or more disorders (or diseases) simultaneously but independently, in addition to a primary disease or disorder [34]. They are patients with a minor form of Chronic Obstructive Pulmonary Disease (COPD). Their quality of life is improved when they are active and regulating their weight. However, when being active, for example when walking, it is important to monitor their oxygen saturation level for safety reasons. If the saturation level drops too low, exacerbation may occur, leading to hospitalization and more expensive long-term care. Moreover, they have high blood pressure and their blood pressure should be measured regularly. Additionally, John has a hearing disorder while Mary has vision impairment. Besides that, John suffers from amnesia and needs to be reminded to his tasks. Mary suffers from Parkinson disease and she cannot move around easily, so she uses a wheelchair. John and Mary are prescribed to take certain medicines at certain times. Nancy, as their professional care-giver, is responsible to create and tailor the homecare services installed in the care facility".

As described in the example scenario, $John$ and $Mary$ have individual needs, and $Nancy$ must create different personalized services by tailoring already deployed services. Creating such a service requires both domain and IT knowledge. As an example, to monitor John's blood pressure, Nancy should specify when and how often John should measure his blood pressure, how many times and how much time in advance John should receive a reminder to measure his blood pressure, the content of the reminder message and modality of the message (e.g., when a message should be presented as text or as a spoken message). Nancy should also specify that if John does not measure or his blood pressure is not within the specified range, an alert should be sent to a specific care-giver with a specific alert message.

1.3 Objectives

The main goal of this thesis is to enable an end user (specifically a non-technical care-giver) to create personalized homecare services by tailoring already deployed basic homecare services. This thesis proposes and validates a service tailoring method and tool, which can be used to assist a care-giver to create a personalized composite service for a care-receiver as easily and as quickly as possible.

As discussed in Section 1.1, there are some ongoing research targeting homecare services. However, there are still several challenges which need to be addressed. Specifically, current automated homecare support systems are hard to use by non-technical end-users and hard to change or adapt when new requirements have to be met. The top level goal of the work reported in this thesis is to improve homecare systems. We want to improve homecare systems by facilitating the service creation process, in terms of reducing the IT skills, time, and effort needed by care-givers to create new services that suit the individual needs of care-receivers. The improvement criteria can be classified as:

- Same or reduced costs of care provisioning to elderly persons
- Better or same quality of care provided to elderly persons
- Improved or same quality of life experienced by elderly persons

In this thesis, we are trying to answer how the required homecare IT services can be tailored to the individual needs of the care-receivers, while requiring minimal technical knowledge and skills from the care-givers. To answer this question, we focus on the following properties of our service tailoring approach:

1. **Tailorability**: Care-receivers have different aging and health problems, impairments, and abilities, as well as different lifestyles and living environments. Therefore, care-givers should be able to create services that are aligned with the personal needs of the care-receivers. We define a tailoring process and architecture, to allow care-givers to configure patient-neutral services and to adapt them to a specific care-receiver's needs. Moreover, it should be possible to incrementally change the result of the initial tailoring at later points in time (e.g., because the care-receiver's needs have evolved) by subsequent applications of tailoring.

2. **IT transparency**: Care-givers are expected to drive the tailoring process, as they have the domain knowledge need to decide the content of the services. However, we cannot expect that care-givers to have deep IT knowledge or to be willing to invest time in acquiring such knowledge. Therefore, service tailoring should be possible at a proper abstraction level for the care-giver, hiding the underlying technology that is used to realize the service tailoring process and the service.

3. **Allow creation of context-aware services** - A care-receiver may use the created service in different circumstances, and may have different needs depending on the circumstances. Therefore, if such circumstances can be foreseen by the care-giver, she should be able to include the use(r) context-dependent options in the created services.

4. **Model-based** - The needs of a care-receiver change over time. Instead of creating new services from scratch, it should be possible to adapt previously created services. For this reason, the creation process should be decoupled from the software and hardware, by exploiting a service model from which application data and code can be automatically generated. The service model should be retained and made available on request of the care-giver as a basis for modification in accordance with the changed needs and requirements of the care-receiver.

1.4 Research Questions

The starting point of this research work was the need for a new method to personalize homecare services by non-technical end-users as quickly as possible. In order to achieve and to analyze the objective of this research more precisely, the following research questions are considered. These questions help to break down the tailoring problem into smaller subproblems, and resolving these subproblems help us to reach our research goals.

– **RQ1**: *What are the common homecare tasks performed by care-givers and their corresponding IT-based homecare services?*
There are several tasks which care-givers perform in the homecare domain. Some of these tasks cannot be automated such as cleaning the room and washing the care-receivers, but some other tasks such as monitoring care-receivers' vital signs and scheduling different activities can be automated. Among those tasks which can be automated, the most common ones and their required IT-based services should be identified.

– **RQ2**: *What is a (generic) service tailoring process?*
A *service tailoring process* should define the flow of actions to help an end-user create the personalized homecare service that is needed. The service tailoring process should clearly specify this flow of actions in order to obtain an executable composition of basic services. This service tailoring process should be sufficiently 'generic' that it could be used in any homecare system with little or no adjustment.

– **RQ3**: *What are the components and entities, needed for service tailoring?*
In a typical scenario, a service tailoring environment consists of a service tailoring platform and users (i.e., care-giver, care-receiver, and possibly an IT technician such as a programmer or a maintenance personnel member). The required components and entities should be identified for this service tailoring environment and evaluated as to whether the identified components are sufficient for the users to achieve their goals.

– **RQ4**: *What can and cannot be automated in the service tailoring process?*
  To facilitate the tailoring process and make it easier for a care-giver, there is a need for some level of automation (e.g. using information stored in a user profile of the care-receiver). However, homecare services are safety-critical systems, i.e., systems for which a malfunction could lead to injury or even to the loss of a life [94]. In addition, it has been found that completely relying on automated systems for the elderly with several comorbidities may have undesirable effects [34]. Therefore, the tailoring process should explicitly specify what can be automated and what requires human intervention.

– **RQ5**: *What are the risks of using our approach?*
  Accountability is an important factor in the homecare domain, which means that if something goes wrong, somebody should be responsible for the decisions that have been made while tailoring the services. Furthermore, besides giving the ability to configure services to the care-givers, in order to reduce human mistakes, the tailoring platform should be able to check for abnormal values; possibly by adding some constrains and rules for acceptable configurations.

– **RQ6**: *Is the tailoring platform usable and useful?*
  A prototype of the proposed approach and architecture should be implemented and tested by care-givers to evaluate its usability and usefulness. We should be able to answer the following questions concerning using the tailoring platform and its user interface:

  a) Can care-givers effectively tailor the services without our assistance?

  b) Does the system indeed save care-givers time and how efficient is the service tailoring platform?

  c) Do care-givers perceive the tailoring platform as usable (i.e., perceived satisfaction of users)?

  d) Does the tailoring platform in particular and the IT-based homecare system generally improve the quality of care and increase the quality of life of care-receivers?

1.5 Research Methodology

We follow the design science methodology as described by Wieringa in [167, 164], while addressing the problems identified by this thesis. As shown in Fig. *1-1*, we study an artifact that interacts with a problem context to produce effects. We will evaluate these effects with respect to specific design criteria.

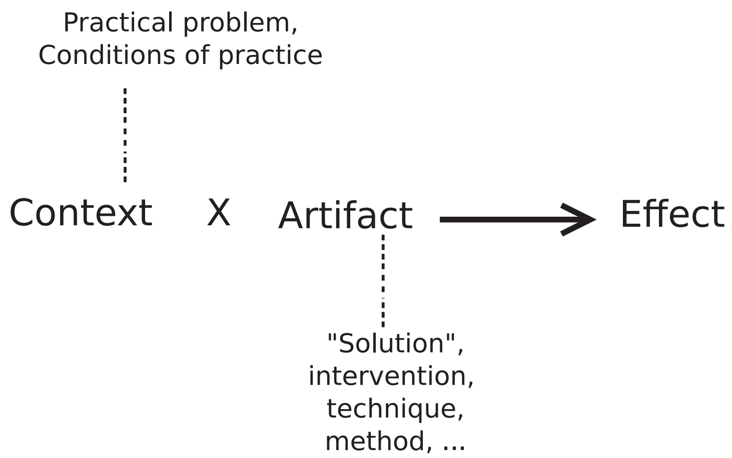

Figure 1-1 : Design science definition, as defined in [168]

In the context of the work presented in this thesis, we define the artifacts, context, criteria, and effects as follows:

- **Artifacts**:
  Service tailoring process and platform for personalizing IT-based homecare services by care-givers.

- **Context**:
  (a) Care centers which take care of elderly people.

  (b) Homes equipped with mobile and IT-based care services to enable elderly persons to live as independently as possible.

- **Effect**: Service plans created by the care-givers using the proposed service tailoring platform and process which satisfy the individual requirements of the care-receivers.

- **Criteria**:
  (a) *Effectiveness*: Whether the care-givers can create service plans using the proposed approach and without our assistance.

(b) *Efficiency*: How long does it take for a care-giver to tailor a service plan in relation to the accuracy and completeness of the service plan.

(c) *Learnability*: If the care-givers learn the service tailoring process over time, they should be able to create the service plans faster.

(d) *Satisfaction*: What is the care-givers' perception about the ease-of-use and benefits of the service tailoring platform (to be investigated through interviews with the care-givers).

To carry out the research presented in this thesis, we follow the engineering cycle introduced by Wieringa in [163]. As shown in Fig. *1-2*, this cycle includes five main parts: *problem investigation*, *treatment design*, *design validation*, *treatment implementation* and *analysis of results*.

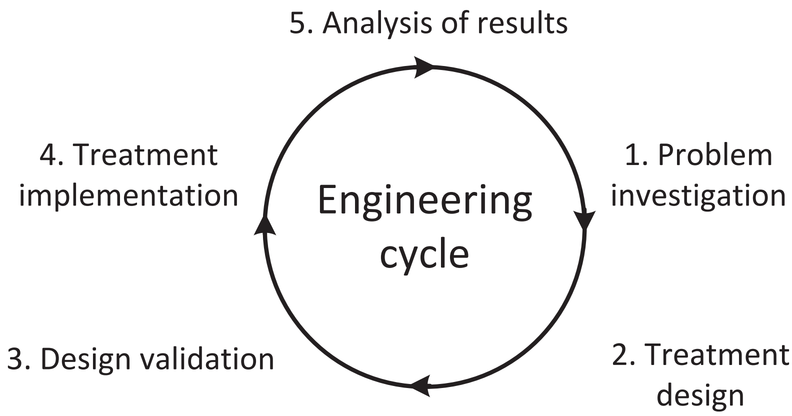

Figure 1-2 : Research methodology used in this thesis, as defined in [168]

In the context of the work presented in this thesis, as shown in Fig. *1-3*, we perform the following actions for each part:

1. **Problem investigation**: For the problem investigation we perform the following actions:

   – *Selection of common homecare tasks*: We study the current situation of providing care services to care-receivers in a care institute in the Netherlands. We perform the first series of interviews with care-givers at the care institute, to identify a list of common homecare tasks. Moreover, we perform a literature study of existing homecare systems. From these two studies, we choose a list of important homecare tasks which we can use in this thesis as examples and evaluate the tailorability of them during a pilot study.

- *Identification of available and applicable homecare IT-services*: We also identify a collection of homecare IT-services by conducting a technology survey of existing solutions and a literature study, constrained in scope by the usability and applicability of the technology for the identified homecare tasks.

- *Identification of existing service tailoring techniques*: We perform a literature study of user-centric service tailoring techniques.

- *Identification of existing risk based requirement elicitation methods*: We perform a literature study of existing requirements elicitation methods which use identified risks to elicit risk-reducing requirements.

2. **Treatment design**: based on the result of the problem investigation, we design the following components:

   - *Abstract IT-services*: After identification of available and applicable homecare IT-services, we make a list of elementary homecare services by abstracting the technical details that are irrelevant to the service tailoring by the care-giver.

   - *Treatment patterns*: Based on the result of the two previously mentioned studies, namely *identification of common homecare tasks and available IT-services*, we design treatment patterns for each homecare task. We make use of treatment patterns as a starting point for the tailoring process, where a treatment pattern is an activity structure for handling a generic homecare task.

   - *Risk Driven Requirements Specification*: Based on the result of *Identification of existing risk based requirement elicitation methods* study, we introduce a Risk Driven Requirements Specification (RiDeRS) method to identify potential risks of using IT-based services, and to specify additional requirements on the services (and the underlying IT system) to mitigate or prevent these risks.

- *Service tailoring process and architecture*: Based on the result of *Identification of existing service tailoring techniques*, we design the service tailoring process and architecture.

3. **Design validation**: Before prototyping the approach, we return to the same care-givers who participated in the identification of homecare tasks, and interview them two times as follow:

   - *Interview concerning the correctness of the treatment patterns*: We perform the second series of the interviews with the care-givers. We explain how the service tailoring platform will work after implementation. We also show them the treatment patterns to validate their correctness. This study helps us to refine the designed patterns and the tailoring process based on the result of interviews before prototyping the approach.

   - *Interview using RiDeRS*: We perform risk assessment using RiDeRS by performing the third series of interviews with the same care-givers to identify and analyze potential risks of using our services and consequently to identify further requirements on the services and the underlying IT system.

4. **Treatment implementation**: To show the feasibility of the proposed approach, we develop a prototype of the service tailoring platform to be tested in the care institute. To validate the approach, we perform a field test in two series of experimental studies with a total duration of 4 months.

5. **Analysis of results**: The experiments are conducted in a near real-life setting at the care institution and after each series of experiments, we perform an interview with the care-givers. The interviews were analyzed to evaluate the usability of the service tailoring platform in terms of its *effectiveness*, *efficiency*, *learnability*, and *satisfaction*. We also analyze if such a system can be used in practice and if it will save care-givers time and increase the quality of life of care-receivers.

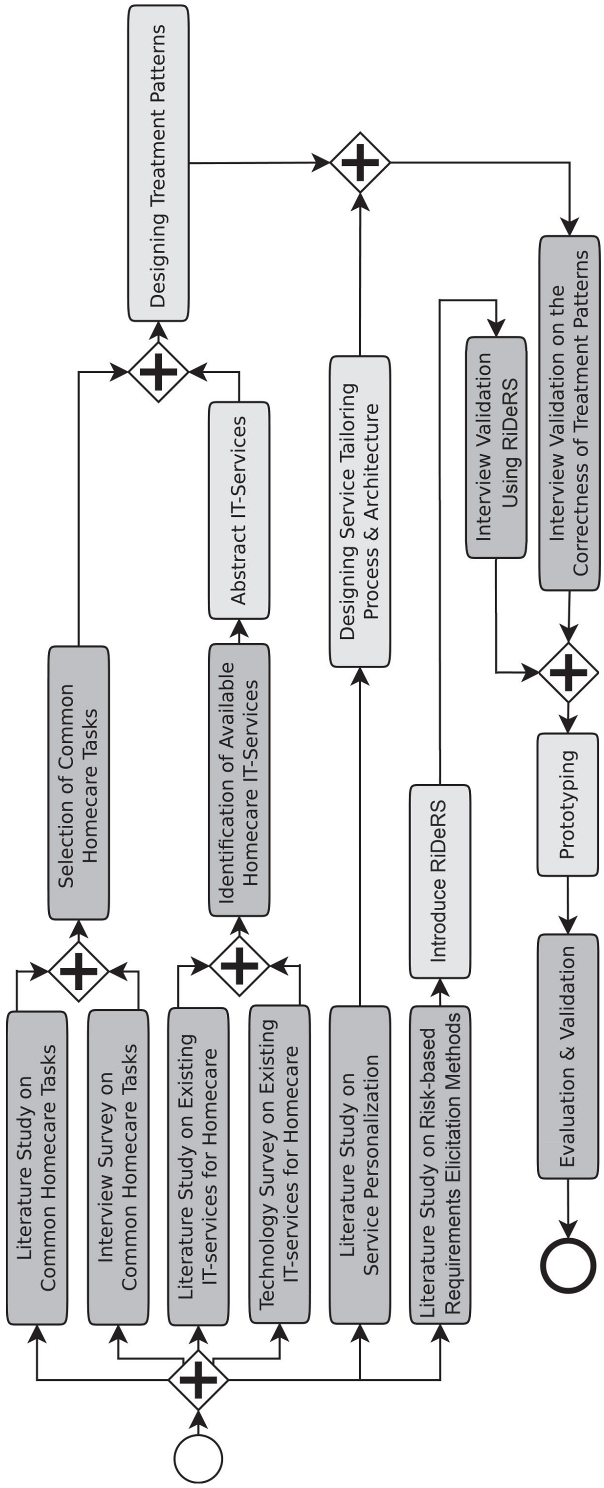

Figure 1-3 : Approach to thesis research

With respect to the case study, the population we consider consists of the care-givers and care-receivers of a care institute, Orbis[1], in the Netherlands (to use and evaluate the approach and the prototype). We are doing only one case study and within this case study we have a limited number of users. However, the case study is studied in detail considering the fact that the users are real users and the scenarios are real scenarios. We will discuss to what extent we can generalize lessons learned from this case study by analogy (i.e. by considering whether similar cases would show similar phenomena). This in turn is motivated by the similarity of the structure of the population units in the care center, and of the similarity in competencies (capabilities) of the elderly people. Moreover, care-givers and care-receivers are aware that they are participating in research, and that they can stop their participation in the experiments whenever they wish.

1.6 Scope

In this thesis, we concentrate specifically on the process of configuring and combining existing homecare services, in the scope of scenarios for independent living of elderly persons. The homecare services created via service tailoring at design-time (prior to provisioning of the services) should satisfy the individual needs of care-receivers. A service tailoring method and tool is designed and is then validated twice in an elderly-care institute.

For this study, we assume that:

- The proposed method will be used in an industrialized world where an aging population is an issue.
- IT-based homecare services are offered by mobile devices.
- Professional nurses (as care-givers) are available to tailor the services and they are the ones who currently provide care services specifically to elderly people (i.e., they are not hospital nurses).
- Care-receivers who live in an elderly-care institute or in their own homes, receive support from professional care-givers (and possibly receive help from other elderly with respect to using the system).

As mentioned in Section 1.1, homecare services are a new paradigm and have become popular and hence, a lot of research has been done

[1] http://www.orbisconcern.nl/

to identify and address the challenges of providing care services in the home of care-receivers. However, in this thesis, we focus on the personalization of homecare services, while being aware of the technological state of the art and addressing related technological challenges.

In this thesis, we do **not** address the following issues:

– *User privacy, security, and trust*: Privacy and security are major concerns in healthcare systems. The system must have appropriate mechanisms in place to control who can access specific personal and health-related information [56].

– *User interface*: User interfaces of applications for the elderly play an important role in the successful introduction of IT-based homecare systems. However, in this thesis, we are not focusing on designing the user interface of applications for the elderly, but rather focus on how care-givers create these services and also the design of the user interface of the tailoring platform.

– *Service provisioning at runtime*: This thesis focuses on the tailoring platform and how a care-giver configures the services prior to runtime. The detailed explanation of how the output of the tailoring platform is executed by a provisioning platform is out of the scope of this thesis.

– *Memory space limitations*: For the long-term recording of health-related information -24/7 monitoring- a huge memory space will be required [136].

– *Identity establishment*: In family situations, it should always be possible to distinguish between the individual members of the family [136].

– *Power consumption*: Battery-powered devices must have a useful lifetime and replacing or recharging batteries should be done routinely by IT personnel or the care-givers, if not automatically [95].

– *Self diagnosis/healing*: Considering the fact that people should be able to use the system without supervision or help from healthcare professionals or technical experts, the system's robustness and reliability should be high [95].

1.7 Thesis Structure

Each chapter of the thesis starts with a short introduction to the chapter and ends with a summary of the chapter. Figure *1-4* presents the thesis structure, indicating how the chapters of the thesis cover the above mentioned research methodology. In the following, we introduce each of the chapters of the thesis:

Part I: Problem Investigation

– *Chapter 1 - Introduction:* provides an introduction to the thesis by presenting background information, motivation, objectives, research questions, research methodology, scope, and a short introduction of the approach we have used.

– *Chapter 2 - Homecare and Tailoring:* provides an overview of the existing solutions which are trying to address a similar problem. In particular, we try to answer the following questions: (a) What are the existing homecare systems and how do they deal with service tailoring? Answering this question, helps us to study existing homecare systems and current homecare technologies and products, and to see whether they consider personalization aspects, and if so how they deal with these aspects. (b) How existing works support service personalization and what features are considered as part of user-centric service creation. This helps to understand their advantages and disadvantages and to identify gaps in exiting work and to frame the need for our proposed solution.
  This chapter is partly based on the following papers:

  – Mohammad Zarifi Eslami and Marten van Sinderen. Flexible home care automation adapting to the personal and evolving needs and situations of the patient. In The Proceedings of The 3rd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pages 1 - 2. IEEE, 2009 ([174]).
  – Ingrid Mulder, Yvonne Schikhof, Martijn Vastenburg, Alan Card, Tory Dunn, Andreas Komninos, Marilyn McGee-Lennon, Mark Santcroos, Gabriele Tiotto, Mieke van Gils, Jan-Willem van 't Klooster, Annelies Veys, and Mohammed Zarifi Eslami. Designing with Care: The Future of Pervasive Healthcare. IEEE Pervasive Computing, 8(4):85-88, 2009 ([121]).

- *Chapter 3 - Roles, Tasks, and Treatment Patterns:* provides part of the results of a first series of interviews with the care-givers. Using these, we analyze the existing situation and present commonly performed homecare tasks and the way these tasks are performed. It also shows what types of care-givers play a role in the homecare domain and what types of services are expected from an IT-based homecare system. This chapter answers RQ1.
  This chapter is partly based on the following papers:
  - Ashiful Alam, Mohammad Zarifi Eslami, and Klaas Sikkel. Elderly and homecare tasks: A literature review on problems. In The Proceedings of The 4th IADIS International Conference on e-Health, pages 211-216, 2012 ([3]).
  - Henry Been and Mohammad Zarifi Eslami. A Survey on Tasks Performed in Eldercare. In The Proceedings of The 4th IADIS International Conference on e-Health, pages 176-181, 2012 ([16]).

Parts II & III: Treatment Design & Design Validation

- *Chapter 3 - Roles, Tasks, and Treatment Patterns:* provides the results of the second series of interviews with the care-givers. Here, we present the different stakeholders who play a role in the homecare domain and common homecare tasks. Moreover, we present treatment patterns of each common homecare task in business process model and notation (BPMN) and shows how we evaluate them together with the care-givers.

- *Chapter 4 - Risk Driven Requirements Specification (RiDeRS):* presents a conceptual framework and a method for systematically eliciting documenting risk-driven system requirements. IT-based care systems can introduce new types of risks such as those related to availability and accountability. In order to prevent this, we propose a method to identify potential risks of using such a system, and to specify additional requirements of the system to mitigate or prevent these risks. This chapter answers RQ4 and RQ5.
  This chapter is based on the following papers:
  - Mohammad Zarifi Eslami, Brahmananda Sapkota, Alireza Zarghami, Eelco Vriezekolk, Marten van Sinderen, and Roel Wieringa. Risk Identification of Tailorable Context-aware Systems: a Case Study and Lessons Learned. In The Proceedings of The CAiSE'12 Forum at The 24th International Conference on Advanced Information Systems Engineering (CAiSE), volume 855 of CEUR Workshop Proceedings, pages 40-49. CEUR-WS.org, 2012 ([173]).

  - Mohammad Zarifi Eslami, Brahmananda Sapkota, Andrea Herrmann, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Risk Driven Requirements Specification (RiDeRS) of IT-based Homecare Systems. In The Proceedings of The CAiSE'13 Forum at The 25th International Conference on Advanced Information Systems Engineering (CAiSE), 2013.

- *Chapter 5 - Service Tailoring Platform and Process:* presents our proposed service tailoring process and architecture. This chapter answers RQ2 and RQ3.
  This chapter is based on the following papers:

  - Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Flexible Homecare Application Personalization and Integration Using Pattern-Based Service Tailoring: Supporting Independent Living of Elderly with IT. In The Proceedings of The 11th IEEE International Conference on Computer and Information Technology (CIT), pages 467-474. IEEE Computer Society, 2011 ([176]).
  - Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service Tailoring: Towards Personalized Homecare Services. In The Proceedings of The 4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing (ACT4SOC), pages 109-121. SciTePress, 2010 ([175]).

- *Chapter 6 - User Profile:* presents our proposed structure for a user profile and how this user profile can help care-givers to tailor the services. This chapter answers RQ3.
  This chapter is based on the following paper:

  - Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service Tailoring: Towards Personalized Homecare Services. Procedia CS, 5:409-417, 2011 ([177]).

Part IV: Treatment Implementation

- *Chapter 7 - Experimental Prototype*: presents the prototype implementation of our approach.

Part V: Analysis of Results

– *Chapter 8 - Validation: Experiments and Results*: presents two experimental studies using the prototype in the care institute. Using the results obtained from this pilot study, we evaluate and validate the work presented in this thesis. This chapter answers RQ6.
  This chapter is based on the following paper:

  – Mohammad Zarifi Eslami, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Care-giver Tailoring of IT-based Healthcare Services for Elderly at Home: A Field Test and its Results. In The Proceedings of The 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013.

– *Chapter 9 - Conclusions and Future Work*: reflects on the work presented in this thesis. It discusses lessons learned and reusable results in other cases/domains. It further presents some challenges in the area of user-centric service tailoring in the homecare domain and highlights potential future research directions.

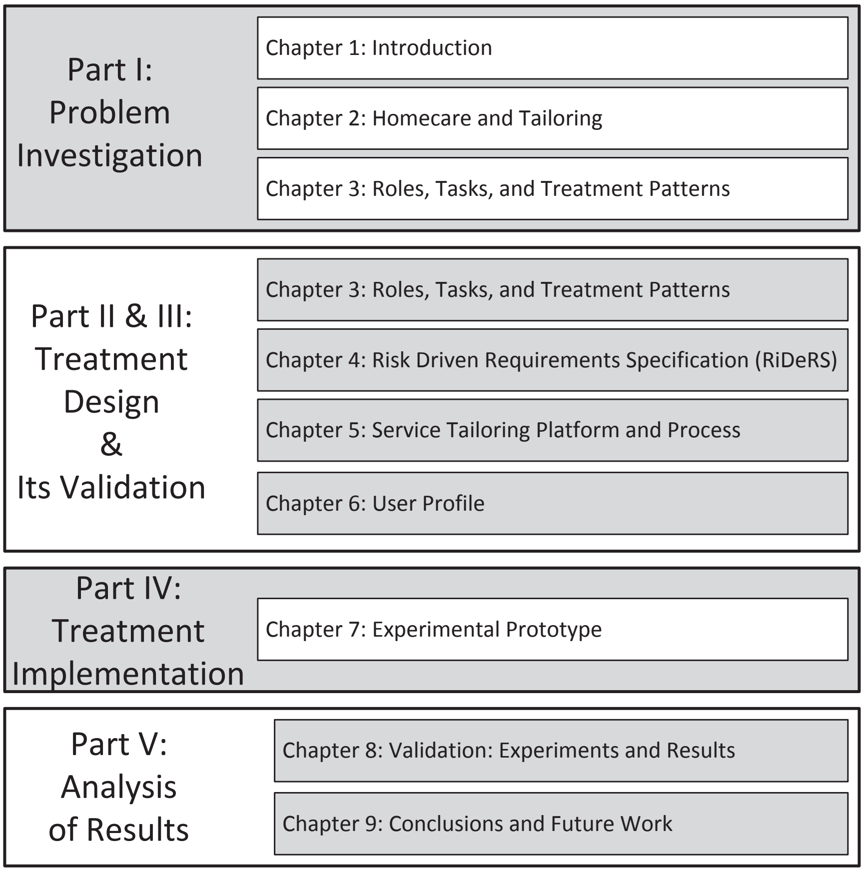

Figure 1-4 : Thesis structure

Chapter 2

Homecare and Tailoring *

"When you are finished changing, you are finished."
— Benjamin Franklin

IT has penetrated every aspect of human life in the current society and healthcare is no exception. IT-based homecare systems are employed to provide care services to care-receivers in their home environment. One of the biggest drivers behind "care at home" is the increasing aging population. In Section 2.1, we study a few homecare systems, which are implemented in real life, then we enumerate the specific characteristics and challenges of those systems.

Application functionality provided to users as services are usually designed for a general purpose, user, or situation. In reality, different people have different needs, therefore they prefer tailor-made services. Service tailoring has been considered a practical approach to accommodate the differences between individuals. Service tailoring is increasingly becoming a factor in all IT-based systems. Section 2.2 presents some common service tailoring techniques.

---

*This chapter is partly based on the following papers:

– Mohammad Zarifi Eslami and Marten van Sinderen. Flexible home care automation adapting to the personal and evolving needs and situations of the patient. In The Proceedings of The 3rd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pages 1 - 2. IEEE, 2009 ([174]).
– Ingrid Mulder, Yvonne Schikhof, Martijn Vastenburg, Alan Card, Tory Dunn, Andreas Komninos, Marilyn McGee-Lennon, Mark Santcroos, Gabriele Tiotto, Mieke van Gils, Jan-Willem van 't Klooster, Annelies Veys, and Mohammed Zarifi Eslami. Designing with Care: The Future of Pervasive Healthcare. IEEE Pervasive Computing, 8(4):85-88, 2009 ([121]).

This thesis focuses on using IT services to provide care to elderly people or people with physical/mental impairments. These users have specific but widely varying requirements (capabilities, preferences, health status, etc.). Therefore, a homecare system should support tailoring of homecare services to the needs of each individual, and it should be possible to make incremental adaptations if and when needed. Hence, in Section 2.3, we discuss tailoring some of the existing homecare systems and then we position the work described in this thesis. Finally, in Section 2.4, we summarize the chapter.

2.1 Homecare System

It is hard to articulate the inclusion criteria for the term "elderly people". This is because age does not necessarily indicate a fixed set of physical conditions, as this depends on lifestyle, economic and cultural conditions, availability of healthcare, etc. In this thesis, we refer to an *elderly person* as a person who needs the help of professional care-givers to deal with age-associated diseases. Examples of aging-associated diseases are cardiovascular disease, type 2 diabetes, and hypertension. The incidence of these and other diseases increases rapidly with age [36].

The style of life has radically changed in recent years. Earlier, children and family members usually took care of their parents as they grew older within the extended family home. This is still the case in some developing countries. But in modern societies, this attitude has changed due to the following reasons:

- Nowadays, in most cases, both wife and husband have their own jobs and work outside their home.
- In many cases children move to other places for economic and/or social reasons.
- The size of families has decreased and fewer family members are available to provide care for their parents.
- More elderly prefer to live independently.

Consequently, people are less responsible for their elders. Elder-care is now being provided mainly by professional institutions. These *elder-care institutions* hire experts and volunteer *care-givers* to provide care and social services to their *care-receivers*.

Due to the increasing elderly population, providing care via conventional care institutions might not be feasible anymore. This gives birth to a new paradigm in health industry: "IT-based home healthcare" [161]. From now on, for brevity we simply call *"IT-based home healthcare"* as *"homecare"*.

Arguments for homecare include the following:

1. **Economical**: providing more efficient healthcare solutions with less labour

2. **Social**: facilitating and prolonging independent living

3. **Medical**: providing continuous health monitoring and care services a round the clock

A *homecare system* includes platforms, services, devices, data and networks that are required to support independent living of the elderly. In this thesis, as one of the homecare systems' platforms, we focus on a tailoring platform. A *tailoring platform* includes a architecture and tailoring software, where the combination provides an environment for tailoring of services by the care-givers, and requires minimal technical knowledge and less time/effort.

In the remainder of this section, we study different types of homecare systems and their associated challenges.

2.1.1 Current Homecare Systems

There is an emerging trend in industrialized countries for using IT-based homecare services [13, 92, 161, 178, 105, 55, 47]. We are now witnessing many innovations in the area of homecare, thanks to recent technology advances in areas such as sensor technology, body area networks, wireless communications, and information processing. This has already enabled an array of applications, ranging from health monitoring, event-based alarms, and automated analysis to communication of health-related information. Several research projects have studied the requirements of homecare systems and evaluate their feasibilities. Examples of these projects are presented in Table *2-1*.

We classify homecare systems as follows:

– *Monitoring systems* - Homes of the elderly are equipped with a variety of devices and sensors to enable remote monitoring of care-receivers' vital signs and activities. The monitoring system will notify care-givers when a hazardous condition is detected for a care-receiver. Moreover, care-givers have access to the collection

Table 2-1 : Some existing homecare projects

Project name	Project Description
CLEAR [28]	a system for managing treatment of chronic diseases of the elderly
epSOS [45]	an interpretable patients' data system
T-Seniority [152]	a system with e-care services through digital TV for the elderly
Dreaming [40]	elder-friendly monitoring and alarm services
MATCH [107]	middleware for integration of social and healthcare services at home for elderly persons
MPOWER [120]	middleware that dynamically integrates heterogeneous systems to achieve interoperability between services and devices in mobile and home networking
AMIGO [6]	an open platform to ease developing and deploying services for the elderly

of health information of each elderly person. That information could assist care-givers to prescribe appropriate treatments and take proactive actions to maintain care-receivers' health. Monitoring vital signs such as blood pressure or oxygen level and dispensing medicines and monitoring elderly persons daily activities are examples of services that can be provided by monitoring systems.

– *Notification systems* - Forgetfulness is one of the common problems of the elderly. Due to this, an elderly person might forget the exact time of their necessary health activities such as taking medicine or measuring their own vital signs. Notification systems are used to remind care-receivers of a specific activity; preferably using various modalities.

– *Interactive systems* - These systems support two-way (audiovisual) communication between care-givers and care-receivers. This enables care-receivers and care-givers to discuss care related issues remotely. It also allows care-givers to provide some advice to care-receivers or to provide instructions w.r.t. treatments. This communication can be established manually either by care-receivers or care-givers when it is needed, or automatically by the system upon detecting an emergency situation.

– *Social activity and interaction systems* - Due to physical impairments and psychological conditions, elderly persons usually do/can not participate in social activities and have fewer social interactions than they would like to have. Social interaction systems, similar to interactive systems, provide two-way audiovisual communication between care-receivers and family members, friends, or other care-receivers to stimulate social connectedness and reduce isolation. Elder people visit their friends less, less frequently eat out at a restaurant, and find it difficult to sustain other kinds of social activities, such as attending church. Social activity systems can remind and motivate elderly persons to attend various activities which suit their interests, for example by showing the menu options of the day for a nearby restaurant or showing the schedule of a nearby theater. It can also enable two or more care-receivers with similar interests to meet and discuss their activities. Moreover, it may include entertainment services such as eBooks, eNews, and on-line puzzles and games to assess cognitive decline, and to stimulate the elderly physically and mentally.

Homecare systems may take contextual information into account, i.e., they can adapt their behavior based on a model of the user's current context and changes in the user's context [11]. Context-awareness is becoming an important feature of all information systems. In the literature, there are a number of definitions of context information [67, 133, 37, 85], but no consensus has arrived on how to define context. We consider *context information* as any information about the physical context of the system which can be used to adapt the response of a homecare system in order to add value to the provided services to entitled users. For instance, a context-aware homecare system decides to trigger or not to trigger an alert based on the care-receiver's current blood pressure and predefined threshold levels. In such systems, context information such as the care-receiver's location, health condition, and physical activities is combined with information entered by keyboard, such as information about preferences, all of this information is then utilized in order to, for example, send appropriate alerts to care-givers and notifications to care-receivers.

There are other research domains that share similar activities with homecare such as smart homes and mobile-based care services. Smart homes contain interactive and pro-active devices, such as sensors and actuators, to be aware of their state and react to events and user's needs through extensive inter-operation and user interaction [2]. These systems are general and do not target the independent living requirements of elderly people. Mobile-based care services are used to measure and transmit bio-signals of care-receivers to be monitored and analyzed by remote healthcare organisations [91]. Some homecare services might be made of mobile-based care services.

2.1.2 Homecare System Challenges

Although some homecare applications have been proven in practice, and several promising prototypes have been developed in research projects, many challenges still need to be resolved before homecare systems become cost-effective and efficient. Some of these challenges are numerated and explained below:

– *Computational and storage limitations* - The computational and storage cost of processing and recording health-related information -24/7 monitoring- for the long time is a technically challenging issue [136]. The devices involved in the homecare domain such as mobile devices have limited computational and storage capacity.

– *Identity sensitivity* - In a homecare system, distinguish between the individuals using the system is vital [136, 178].

– *Power consumption* - Battery-powered devices must have a appropriate lifetime and replacing or recharging batteries should be done routinely by service providers, if not automatically [95, 136].

– *Self diagnosis/healing* - Considering the fact that the elderly should be able to use the system without supervision or help from healthcare professionals or technical experts, the system's robustness and reliability are of major importance [161, 55].

– *Privacy protection* - Privacy is a major concern in homecare systems. The system must have appropriate mechanisms in place to control who can/cannot access specific personal and health-related information [56, 122]. In the homecare domain, different roles have different authorities and privacy aspects should be achieved by providing different authorities to different roles such as care-givers, patients or family members.

– *Security issues* - The care-receiver data such as personal information, health records, or created and configured services must not be altered/stolen. Attempts to do so could be done by individuals or organizations who break into the system and its network. Security is an important part of any homecare system [7, 140].

– *Non-intrusiveness* - People prefer to have as much of a normal life as possible in their home. The less intrusive a homecare system is the more desirable the system would be [92, 136].

– *Ease of use* - Typical users of homecare systems have no or limited technical skills, i.e., home care systems should be highly usable [92, 148].

– *Proper business model* - Homecare is a complex domain with multiple stakeholders. A successful homecare system should include a viable business model in which all stakeholders benefit and can agree upon. Therefore, a value network of homecare systems should be defined where all actors including producers, providers, and consumers of such systems are defined and their value exchanges described [111].

– *Safety of care-receivers* - The use of IT systems can provide benefits in homecare domains. However, these systems can also introduce new or increased risks for the care-receivers. Such risks arise from the assumptions that are made during the design of the system, which are not appropriate for the execution context of the system. There should be a set of strategies that enhance the reliability and availability of the provided services, as well as enhancing the safety of the care-receivers [94, 5].

– *Tailorability* - An additional concern, apart from the ability to offer certain application functionality, is how to cope with the uniqueness of the care-receiver's needs and preferences, and the dynamicity of the care-receiver's condition and circumstances [160, 55]. The threshold level for a blood pressure alert for one individual may be different from that of another, and can also change over time. Therefore, in homecare systems, the ability of the system to cope with individual needs is essential. Due to an elderly person's physical/mental limitations, one can not rely on the care-receivers to do the service tailoring. In this case, service tailoring should be done by someone who can decide upon care related issues of each elderly person according to the needs of each person.

In this thesis, we focus only on the *tailorability* challenge, while remaining aware of the other challenges. Moreover, in chapter 4, we study the *safety* challenges that arise from using a homecare system and define a method to do a risk assessment of such systems *before* using them in practice. We also identify additional requirements of the system in order to mitigate those risks.

2.2 Service Tailoring by End-users

In this section, we discuss the state-of-the-art in the area of user-centric service tailoring, also known as End-User Development (EUD) [99]. Service tailoring by domain experts [33] is not a new paradigm and it has been discussed extensively in the literature. The idea of empowering end-users to develop systems themselves, and to adapt the system to their needs, originated in the very beginning of the computer systems discourse. In 1977, Kay [86] proposed the idea of providing end-users with domain-specific design environments that empower them to create their desired products.

In information systems, tailorability is motivated by the idea that is economically infeasible to design turn-key systems (packaged software) which is appropriate for all users and all situations [104]. In the literature, several similar concepts have been used to describe what we call *tailoring*, such as adaptation, customization, end-user modification, personalization, etc. [116]. Even though all these concepts describe similar activities, they sometimes refer to different phenomena or do not specifically mean tailoring. *Service tailoring* is a set of activities simply to modify a computer application within the context of its use by its end-users [117, 64].

To enhance reusability and end-user tailoring, the concept of building blocks is used in many domains. Building blocks enable end-users to create their own desired objects [115]. For example, using Lego bricks, children can make different shapes. The building blocks can be considered as words which by composing them in different ways makes it possible for people to create different sentences. The notion of building blocks is also used in IT applications, for example components in a component-oriented applications [151] and services in a service oriented architecture [128, 46]. In information systems, *building blocks* are the smallest distinct self-contained units, and they are represented as a abstraction of implementation components which can be used in tailoring applications [115]. The underlying implementation details of each building block have been abstracted, so they can be easily used by end-users to design and implement end-user tailored applications.

With respect to our work, tailoring consists of two distinctive activities: composition of building blocks and configuration of each building block. This complies with the definition provided by Mørch [116], who describes tailoring on three different levels: *customization* of configuring parameters of the selected building blocks, *integration* (composition) of the composition of building blocks (including adding, removing, or changing the order of building blocks in a composition), and *extension* to extend the functions of building blocks.

We believe that extension is out of the scope of this work, because of the following reasons:

- Extension requires radical changes (e.g., changes to the underlying code itself) [106, 114] and this requires relatively deep IT knowledge by the end-users. In our work, this would require that care-givers (as domain experts) have some IT knowledge. We do not make such an assumption.

- Extension could raise a number of obvious issues concerning correctness and consistency. However, as the homecare domain is a safety critical domain [52] and any mistake could harm the life of care-receivers, such extensions are unsuitable for our setting.

- Care-givers have limited time to provide their services. Extension could be a time-consuming task. It could make tailoring an obtrusive activity, which distracts the care-givers' attention from their primary task and the cognitive load of switching from extending to tailoring would be increased [99].

Having ruled out extension, in the remainder of this section, we describe different methods which are based on customization, composition, or a combination of them.

2.2.1 Customization Approaches

Customization, also known as parametrization and configuration, empowers end-users to tailor systems by assigning possible values to a range of predefined parameters and helps them to select a coherent configuration among a set of alternatives [117]. Customizable systems usually contain various run-time configuration parameters with default values and depending on the type of the parameters, a user can select, (un)check, or assign a value to the parameters. In the context of component-based tailorability, customization modifies the parameters of already deployed components [118]. To support customization, a tailorable system usually includes configuration forms (user interfaces), which allow a user to edit attribute values of various parameters and these values are interpreted to tailor the system. Depending on the specific domain, these configuration forms should provide a user friendly interface, which can be easily understood by the users knowledgable in that domain. These forms may be static, dynamic, nested, or wizard-based forms [57].

Some customization methods support context-awareness. This means that an end-user can specify system behavior in a given situation by configuring the appropriate parameters. Beyond simple parameter specification, rule-based methods can be employed to empower users to define or modify the behavior of a context-aware system. Decision rules determine the possible adaptation at runtime, based on evaluation of the rules with runtime data (e.g., context dependent values) and predefined configured parameters. For example, a decision rule, in a care application that uses voice messages, can specify that voice messages should be played immediately unless another person is present based on run-time context value (e.g., people nearby) and user (privacy) preferences. As an example of such a rule-based method, Event-Condition-Action (ECA) rules [39] can be used to model context-aware behavior, where on an *event* (context change), based on values previously set for predefined *condition* parameters, the system can perform some desired *action*. The condition is a boolean expression with the event as an input variable; if the condition evaluates to 'true' the action will be executed. In context-aware systems, the event is typically something that happens in the real world (captured by a sensor), the condition is a test whether some situation is true, and the action can be a service offered to an end-user.

An important question in customizable systems is what can be parameterized and what not. This is because a developer should identify potential parameters by considering what a user would like to tailor. On the other hand, the number of configuration parameters and their possible values should be limited. Having more parameters (and their possible values) yields extra tailorability; however, it also increase the complexity of the system, making correct configuration more error prone and requiring more time of the user. Therefore, the developer should perform a trade-off analysis to provide an efficient tailorable system with the appropriate number of parameters and their appropriate values.

2.2.2 Composition Approaches

Integration, also known as composition, is another level of tailoring which goes beyond customization. It allows end-users to interconnect pre-defined building blocks within or between (an) application(s) [116]. Composition utilizes the idea of "plug-and-play" to construct larger blocks of functionality from independent smaller blocks where each building block performs different tasks. The output of a composition is a new value-added functionality that should be designed to satisfy the end-user's requirements. This functionality cannot be delivered by any of the individual building blocks.

Such tailorable systems should enable end-users to construct their own application using their domain knowledge and without requiring programming skills or access to implementation details of each building block. A graphical editor should enable end-users to visually connect their selected building blocks without directly manipulating code.

Different methods have been introduced for end-user composition. In the context of component-based tailorability, composition creates or modifies assemblies of components [169, 118]. These realizations provide a graphical tailoring environment. In this environment, instances of components are visualized as boxes, interfaces are represented as connectors at the surface of the boxes, and a connection between two ports is represented by a line. End-users tailor the system by adding or removing (instances of) components and interconnecting the components to realize their composition [169].

The user centric service composition paradigm was introduced in the service-oriented computing literature to enhance end-user composition by means of composing different distributed network-based services [59, 102]. The concept of mashup is used to describe both the services and the composition process in an intuitive way [103, 179, 69]. A mashup allows end-users to combine different services represented in visual forms that abstract away the technical detail of each service and the service composition process.

2.2.3 Template-based Approaches

There are two issues concerning composition based tailorability methods. First, selecting the appropriate building blocks which are needed for one specific composition, i.e., selecting the ingredients of one composite application. The system should support users in selecting the appropriate building blocks from a larger set. The second issue is how to present these building blocks and their composition in a way that is easy for the users to understand. To overcome these issues, some methods use *templates* (also known as patterns). The motivation for using templates is that the composition of building blocks from scratch is a difficult and time consuming task for end-users. Templates are designed by developers to be used as a starting point for a tailoring activity. A template-based tailorable system proposes an appropriate template to the users based on a specific task, the template can subsequently be edited by the users [4]. The template-based approaches are an example of model-driven development [134, 9], where the user provides only a conceptual description of the intended activity (i.e., selecting a template for a specific task). The underlying conceptual model is used by the system to generate the corresponding executable code [99].

Template based approaches are suitable only in those situations in which the number of potential services, activities, and users are limited [59]. When, there are a limited number of possible compositions, then developers can identify and implement suitable templates at design time. These templates can later be edited by end-users.

In business process modeling, there is a concept of a configurable reference model [61]. This concept is a similar to that of a template. A configurable reference model is the generic process of a specific task. A developer defines such a model by studying how a task is generally carried out in that domain. Later, at run-time, domain experts customize this model by specifying their application's context. For example, a developer can define a general process for issuing a 'driving licence'. Later, each municipality can customize this process by removing or adding activities to the general process.

Although using templates is handy and may be suitable in some situations, assuming that users can always easily manipulate templates for their own needs does not hold true in all situations [59], especially when users have limited IT-knowledge and/or limited time. In these cases, the user simply selects a suitable template from among available templates that satisfy his/her requirements without manipulating the template. Then, the user can further configure each building block within the template, and as a result the building blocks are bound to concrete services at run-time based on the user's configuration values and the available context information.

2.3 Discussion

Generally speaking, tailoring can be done by several actors in different stages of the service life-cycle. In the literature, tailoring is divided in two phases: *design-time* and *as-used* [116]. Tailoring during *design-time* provides mechanisms to facilitate the communication between developers and domain experts. To easily and efficiently capture user requirements, this type of tailoring might provide a set of graphical representation tools to model different aspects of systems and scenarios. This helps the developers to thoroughly elicit the user's requirements in order to develop an appropriate design. Tailoring *as-used* starts after the initial design of the system and is used to address the unicity and dynamics of users and their environments. In this work, we will adapt the second type of tailoring i.e., service tailoring by users during the actual use of the system (not necessarily during run-time) but after the initial design of the system [169].

As opposed to customization of the user interface (e.g. selecting font size or screen size), by tailoring we mean the tailorability of the behavior of the application at the functional level. Nevertheless, tailorability of functionality, might affect the user interface.

A major objective in tailoring is to minimize the adaptation cost of the tailoring process in the existing processes and activities of the organization. In other words, the less change the tailorability implies to the ongoing processes of that organization, the more desirable this tailoring is [169].

A system may offer a range of different tailorability levels. However, supporting a higher level of tailorability may increase the system's complexity, thus users need more technical knowledge in order to tailor the system. Therefore, there is an inevitable trade-off between the required level of tailorability and the complexity of the system [145].

By *tailorability of a homecare system*, we mean that a care-giver can configure the behavior of the system without help from technical personnel. As discussed in Section 2.2, service tailoring, as proposed in this thesis, is mainly about enabling end-users to configure the parameters of each building block and to edit the composition of building blocks as an integrated service.

To simplify the composition activity, we will use different alternative patterns, where each pattern is a composition of different building blocks selected to satisfy requirements of a homecare task. The alternative patterns can be provided by a tailoring environment for each homecare task, so that a care-giver can select a suitable pattern according to the specific requirements of a specific care-receiver. We made this design choice for two reasons:

- **Limited Composition Possibilities:** Care-givers must follow medical protocols when carrying out homecare tasks. These medical protocols dictate specific compositions and this limits the allowable compositions of the activities in care services. Possible patterns are defined based on existing medical protocols and care-givers' recommendations. Each alternative pattern is defined to satisfy the needs of a particular individual care-receiver.

- **Simplicity vs. Tailorability:** As noted earlier, there is an inevitable trade-off between the level of tailorability of a system and its level of complexity. On one hand we want to have different types of compositions to cover as many different individual needs as possible and on the other hand we would like to make the tailoring process as simple as possible so that care-givers can do so with the minimum effort. The more tailoring options the system provides, the more intricate the tailoring would be. Since the main emphasis of the care-givers' responsibility is driven by a medical perspective of the treatments, rather than on tailoring **and** we found through interviews with the care-givers, that they prefer service tailoring be a quick and IT-transparent activity.

Although some of the existing homecare projects also pointed out the necessity and importance of satisfying the individual care-receiver's needs, to best of our knowledge, none of them support end-user tailoring in the way we explained in the previous section. For example, the Amigo project [6] focuses on provisioning of homecare services rather than their tailoring. Amigo contributes to the Ambient Intelligence domain by providing a context-aware service composition mechanism to support automatic dynamic adaptation.

The Match project [107] framework includes components such as sensors and interaction components as logical software "bundles" within the system which can be dynamically added and removed at runtime. By traversing all possible paths in a directed graph, the framework identifies several composite components for a target scenario. This enables the platform to provide run-time adaption by generating an alternative composition based on the care-receiver's preferences and the available concrete services at runtime. Match mainly focuses on run-time adaptation and there is a limited support of tailorability by care-givers to express their preferences.

The Mpower project [120] is a service-oriented middleware platform that supports rapid adaptation, development, and deployment of services for cognitive disabled and elderly persons. Mpower services are created by coordinating the basic services of the platform as business processes. Despite the similarity between the Mpower project's goal and ours, the focus of the Mpower is more on developers rather than domain experts when adapting services. As a result, the Mpower platform enables developers to rapidly create new and interoperable applications, but care-givers can not adapt these applications.

2.4 Summary

In this chapter, first, we defined the concept of a homecare system and the related terminologies such as the elderly, elder-care institutions and homecare. Then, we studied several existing homecare systems and enumerate a number of characteristics and challenges of these systems.

We also defined service tailoring based on definitions provided in the literature. We then described a number of common service tailoring techniques and presented the state of the art. Further, we reasoned that based on the requirements of this work, we employ the *composition of building blocks* and the *configuration of each building block* as tailoring activities. To simplify the composition activity, we will use the concept of patterns.

Finally, we discussed the fact that to some extend service tailoring is supported by existing homecare systems against which we positioned the work described in this thesis.

Chapter 3

Roles, Tasks, and Treatment Patterns *

"In order to learn, one must change one's mind."
— Orson Scott Card

The tailorable IT-based homecare system aims to increase the quality of life of care-receivers and reduce the burden on care-givers by personalizing and automating currently performed tasks. Before such a system can be used in practice, insight is needed into the current work situation. It is also important that the design of such a system be grounded in real situations. At the end of the day, apart from the actual level of services, the most important thing is meeting the user's *expectation*(s), i.e., what are the functionalities and services, that the tailorable IT-based system should concentrate on? If the user's expectations are not met, then there will be significant dissatisfaction.

To develop a better understanding of the system's requirements and to analyze the existing situation, we used an elderly care-institution in the Netherlands for our case study. Section 3.1 describes this case study. We conducted two rounds of interviews with professional nurses who provide care services in this institution separated by two weeks. As a result of first interview, we identified different roles in the homecare domain and common homecare tasks which are presented in Section 3.2. Section 3.3 presents results of the second interview concerning treatment patterns of each identified task, and finally, Section 3.4 summarizes the chapter.

---

*This chapter is partly based on the following papers:

- Ashiful Alam, Mohammad Zarifi Eslami, and Klaas Sikkel. Elderly and homecare tasks: A literature review on problems. In The Proceedings of The 4th IADIS International Conference on e-Health, pages 211-216, 2012 ([3]).
- Henry Been and Mohammad Zarifi Eslami. A Survey on Tasks Performed in Eldercare. In The Proceedings of The 4th IADIS International Conference on e-Health, pages 176-181, 2012 ([16]).

3.1 Description of the Case Study

We performed research in a realistic simulation environment. It is realistic, because we used a care-institution in the Netherlands, Orbis[1], to do the requirements analysis and test our prototype with nurses and elderly from the care institution. It is still a simulation, because our field test was set up specifically for this experiment and did not involve the use of real medicines. The case study is an action case study [166], in which we aim to improve the current situation of providing care by using a tailorable IT system.

Orbis has residential blocks where elderly can live and receive care services provided through professional care-givers. The aim of this institution is to provide round the clock services to their care-receivers and at the same time to enable them to live an independent life as much and as long as possible.

We aimed to develop a prototype of a tailorable IT-based homecare service system as part of the U-Care[2] project, to be evaluated in this care institution as a field test. U-Care stands for User-tailored Care services platform. The institution is one of the partners in the project and therefore they were aware of the project's general goals.

Before developing the prototype, we have conducted a survey by interviewing professional care-givers in two rounds (on different days) in the care institute. In the first round, we intended to identify the common tasks/services in the homecare domain as provided by care-givers and also to understand the way these tasks are handled by them in the current situation. Then, we went back to the lab and defined the corresponding treatment patterns for each identified homecare task based on the detail explanations given by the care-givers. In the second round, we showed the care-givers our proposed treatment patterns for the identified tasks (based on the first interview), and asked them to refine these patterns. After the each interview, we sent the result of the interview to the interviewee to validate the results internally. The results of the interviews are as follows:

– *Roles* - Identifying the different type of professionals who are active in supporting care-receivers in their daily life, and how much time each profession spends with the elderly. This will provide direction when selecting care-givers to do service tailoring.
– *Tasks* - Learning what types of common tasks are currently performed in the institution and what type of services are expected from an IT-based homecare system for those tasks. This will help us to focus on those tasks which can benefit from an IT-based system.

[1]http://www.orbisconcern.nl/
[2]http://www.utwente.nl/ewi/ucare

- *Treatment patterns* - Learning the characteristics of common tasks and how these tasks are currently performed by the care-givers without using an IT-based system. Further, these detailed descriptions can be used to set up the treatment patterns of each task.
- *Building blocks* - Identify required health service building blocks for creating the treatment patterns and analyzing their tailorability (configuration and composition) possibilities.

In the remainder of this chapter, we explain each round of interviews and their results.

3.2 Roles and Tasks

Before the first interview started, we briefly presented our objectives and also our motivation for doing this interview. The objective was to create among the participants a shared understanding of the meaning and role of service tailoring in the U-Care project. A example scenario similar to that presented on Section 1.2 was used to explore the idea of service tailoring in homecare applications. In addition, we wanted to identify requirements for applications of homecare systems, specifically the requirements for tailorable homecare systems.

We have interviewed 4 professional care-givers (one male and three females) with 32, 30, 24, and 16 years experience in providing care to elderly persons. We did the interview with all of them at the same time, in a meeting setting rather than individually. This way they could discuss the questions/facts and after reaching an agreement they answered the questions. We used written questionnaires as an interview tool. This questionnaire had the following questions:

- *Q1* - Could you tell us, are there other departments/persons who are working with you in providing care services to the care-receivers? How important are they and how involved in the process?
- *Q2* - What are the common health problems (diseases) among the elderly you look after?
- *Q3* - How do you usually provide care services to them?
- *Q4* - What are the important tasks you usually perform while providing care services?
- *Q5* - Are there any tasks which you think are common among care-receivers? What is difference in these common tasks for different people?

3.2.1 Roles

For service tailoring, we assume three distinct types of users depending on their individual knowledge and skill sets: care-receiver, care-giver, and service developer (someone proficient with the service-tailoring facility and the underlying technologies). A care-receiver has contextual knowledge about his own needs and physical environment, but we cannot assume that each care-receiver has in depth medical knowledge or possesses enough technical knowledge. A care-giver has domain knowledge about healthcare practices and procedures, but we cannot assume that each care-giver has intimate knowledge of the personal situation of the care-receive or has full technical knowledge of the devices. Finally, a service developer has technical knowledge about service modeling and technology, but probably possesses little contextual and domain knowledge. Therefore, although the proposed service tailoring process and architecture in this thesis is common to all the different types of users, they may need different interfaces (for tailoring) and may have various authoritative roles (permissions for different levels of tailoring). Moreover, actors may have distinct roles when they are interacting with the system, for example, a care-receiver because of his knowledge of healthcare may also take on the role of care-giver for another elderly person.

Anyone involved in providing care to the elderly in their home might be considered a care-giver in the homecare domain [110]. We identify different types of care-givers who interact with and help care-receivers in their daily life based upon the answers to the *Q1.* The interviewee prioritized the care-givers based on the amount of time they spend with the care-receivers. This helps to specify the target user of any IT service which is developed to assist care-givers in providing care services and provide us with direction when selecting those professionals who will tailor the services using our proposed system.

The identified care-givers (ordered based on their importance) are: professional nurses, family members, informal care-givers (volunteer non-professional care-givers), occupational therapists (who give the client a meaningful day, by means of recreational activities), physiotherapist, physicians, pharmacists, and psychologists. We selected professional nurses as the only care-givers to perform service tailoring activities, because the care-receivers spend most of their time with professional nurses when receiving care services in comparison to other care-givers. Moreover, the professional nurses, in consultation with other care-givers (such as physicians) decide upon the care activities of the care-receivers and guide the care-givers to perform those activities (such as, what medicine is to be dispensed and when it is to be taken or what vital sign should be measured and how often).

In the rest of this thesis, we mean the professional nurses when we refer to care-givers.

3.2.2 Tasks

The goal of the first interview was to identify the common tasks/services, in the homecare domain, that are provided by care-givers and also the way these tasks are handle by them in the current situation. To do this, by asking *Q2* and *Q3*, we concentrated on cataloging common health problems and the process for providing care for each of the problems. This helped us to identify specific tasks performed by the care-givers for each problem.

The care-givers mentioned 15 different health problems which are common among elderly people: incontinence, high blood pressure, dementia, diabetes, stroke or cerebrovascular accident (CVA), bad hearing and/or vision, chronic obstructive pulmonary disease (COPD), sensory problems (hands and legs), arthritis (rheumatism), psychiatric, amputation, Parkinson, cancer, epilepsy, and broken legs and hips.

We wrote down each of these 15 health problems separately on different (small) cards. After gathering a complete list, to limit the scope of our work, we asked them to prioritize these health problems from one to ten based on their commonality among elderly persons. Then, the next question was, how do they usually provide care services for each problem. Answering this question helped us to identify specific tasks for each problem. The top 10 most common health problems among elderly and their required care services, based on our interview, are as follows:

- *Incontinence*
  - A doctor through an examination, specifies that what causes this problem (e.g., because of muscle weakness) and to what level it occurs. This is important to know for treatment and to determine the size (i.e., amount) of incontinence material.
  - The nurses need to change the incontinence materials (a diaper) 3 or 4 times per day (usually at fixed times, such as morning, afternoon, before going to bed, or when the care-receiver takes a medicine and they need to urinate).
  - There are some medicines to treat this health problem, but there is no effective way to cure the problem.
- *High/Low blood pressure*
  - Care-receivers with high/low blood pressure need to take medication. They also have to follow a specific diet.
  - Care-givers need to measure care-receivers' blood pressure according to the doctor's advice.

  - For care-receivers with normal blood pressure, their blood pressure should be measured every morning, usually before they get out of their bed.
  - For care-receivers with high/low blood pressure, their blood pressure should be measured 3 or 4 times per day.
  - If the measured blood pressure is high, the care-givers try to measure it again at a later time, and if it is still high, they contact the doctor for advice.
  - The following blood pressure levels are critical, thus the care-receivers should receive immediate medical relief: if the diastolic level is less than 55 or higher than 100, and if the systolic level is less than 80 or higher than 200.

- *Dementia*
  - Care-receivers with dementia usually forget (one or all of) three things: time, place, and people.
  - Their short term memory usually fails before their long term memory.
  - There are different levels of dementia and usually it gets worse with the time.
  - For the care-receivers who suffer from the first stage of dementia, a reminder service can be helpful, i.e., they usually do the things when reminded to do. They need reminders for different activities such as, when to eat, to drink, to take a shower, and to go bed.

- *Diabetes*
  - These care-receivers need to drink a lot and urinate. They also have a specific diet and need to eat at specific times.
  - The care-receivers are diagnosed by a doctor and usually take medicine as a tablet or an injection. The doctor specifies which medicine and how much medicine or food they should take in.
  - The nurses measures their blood sugar level 4 to 5 times per day, or when there are signs indicating that their sugar level is low/high e.g., when the patients are thirsty or if they shake.
  - The following sugar levels are critical and the care-receivers should receive immediate medical attention: if the sugar level is less than 4.0 mmol or higher than 15.0 mmol.

- *CVA*
  - These care-receivers usually have limited abilities, i.e., have a problem speaking or using their hands or arms, or half of their body does not function.
  - Physiotherapy can help to improve their health condition.

- *Bad hearing and/or vision*
  - There is no specific treatment for this problem.
  - Everything in their house needs to be located in a specific place, so the care-receivers can find it easily.
  - Vocal/Visual navigation systems can be useful.
- *COPD*
  - These care-receivers keep oxygen supplies at home to connect when they need to get extra oxygen.
  - They usually receive some medicines, mainly in the morning.
  - Physiotherapy can help them to gain more stamina to perform daily tasks and to help keep up the strength of their body and also to increase their oxygen level by training their lungs.
  - Doing exercises such as walking can improve their health condition.
  - The nurses remind them to take a walk, if the care-receivers did not walk enough.
  - The care-receivers have a specific diet, for example they should not drink milk or they have extra vitamins/protein in their drinks.
  - In this institute, the nurses could not currently measure the level of oxygen in the patient's blood, however, they would like to do so.
- *Sensory problems (hands and legs)*
  - This problem is mainly the side effect of other health problems.
  - These care-receivers suffer from numbness or painful sensations in the hands and legs.
- *Arthritis (Rheumatism)*
  - These care-receivers receive daily medicine and usually use a wheelchair for moving around.
  - Physiotherapy, walking, and swimming are helpful for them.
- *Psychiatric*
  - These care-receivers are either aggressive or calm.
  - There is no cure, but the nurses can make it easier, by talking with them, giving them extra attention, or by doing some activities, such as playing games.

The next interesting discussion was that usually care-receivers have a combination of health problems. For example, since falling often happens for the elderly, in addition to their various health problems, they also suffer from having broken legs or hips. Another common combination is suffering from incontinence in combination with other health problems.

After identifying specific tasks for each health problem, by asking $Q4$, we have queried for the list of the important tasks that the interviewee performed on a normal day. Some of these tasks can be the same as those identified for specific health problem. They named 13 general tasks. These tasks and their characteristics are:

- *Giving medicine to the care-receivers*
  - The care-receivers usually receive medicines at specified times, mostly 4 times per day: 8:00, 12:00, 17:00, and 21:00. Except for some medicines, such as antibiotics that need to be taken at specific times.
  - There are different type of medicines such as tablets, capsules, syrups, injections, creams, bandages, sprays, and drops. Based on the type of medicine, some care-receivers need the assistance of care-givers to be able to take their medicine.
  - The time to take some medicines are dependent on eating food, i.e., before/after lunch, for example, injections for diabetic diseases should done before lunch.
  - For some care-receivers who can remember when to take their medicine, the whole set of medicine for that day is given in the morning to the care-receivers.
  - For some other care-receivers who forget when to take their medicine, care-givers personally provide the medicine at the correct time of the day.
  - There is a third type of care-receivers who need to be attended by the care-givers in order to make sure that they take their medicine.

- *Reminding the care-receivers to do different activities*
  - Remind them to eat food or to drink.
  - Remind them to take their medicine.
  - Remind them to go for a social activity.
  - Remind them of their appointments with doctors, family, etc.
  - Remind them to wake up or go to bed.

- *Washing the care-receivers*
- *Feeding the care-receivers*
- *Cleaning the care-receivers' room and disposing of garbage*
- *Changing the care-receivers' clothes or diapers*
- *Helping the care-receivers in going to the toilet*
- *Laundry*
- *Monitoring/Measuring the care-receiver's vital signs* such as blood pressure, sugar level, and weight
- *Making phone calls* to doctors, family members, pharmacy, and etc. to give reports about the situation of the care-receivers

- *Writing reports* about the situation of the care-receivers and consulting with other nurses when they change shift
- *Transportation and navigation* of the care-receivers from place A to place B
- *Guiding student nurses*, specifically showing them how the things are done

Looking at the list of the tasks performed by the care-givers, some of these tasks can not be automated (such as cleaning the room or washing the care-receivers). If such a thing would be possible, it would involve introducing robots into very intimate situations for the care-receivers. But some other tasks can be automated. The list of the homecare tasks which can benefit from IT support are:

- *Scheduling different activities*, such as
  - To measure blood pressure
  - To take medicine
  - To measure the weight of the care-receivers
  - To do training
  - To go for a social activity
  - To eat food or to drink
  - To make an appointment with doctors, family members, etc.

- *Monitoring care-receivers' vital signs or activities* such as
  - Blood pressure and heart beat (pulse)
  - Medication taking
  - Oxygen saturation level
  - Sugar level
  - Temperature
  - Weight
  - Training activity

After identifying common tasks which can be automated, we needed to select some of those tasks to be implemented and evaluated in a pilot study. We realized two types of services which could be evaluated in the pilot study: care (i.e., health monitoring) services and social activity services. Based on the available application services in the market and the case studies' requirements, together with the care-givers we selected four tasks for the care services: blood pressure monitoring, oxygen saturation monitoring, weight monitoring, and medication intake support. To automate each care service, we needed to identify their treatment patterns and required configuration parameters, as we will discuss in the following subsections.

The tailoring process of the social activity services are mainly about the scheduling different activities and assigning these activities to the interested care-receivers. This is mainly an *on-off* tailoring, in the sense that a care-receiver may/may not be interested in an activity, and does not require defining a composition of services or identifying configuration parameters.

3.3 Treatment Patterns

When carrying out homecare tasks, care-givers must follow specific guidelines and protocols, i.e., medical protocols [49, 54, 60]. These *medical protocols* (guidelines) are usually in a textual format and consist of a sequence of connected steps to be followed by care-givers for each specific task. For example, a guideline associated with a care-receiver suffering from COPD, specifies that if the oxygen saturation of the care-receiver drops under a specific level, then the care-giver has to, first, give the care-receiver a medicine, then call a doctor and finally, check the oxygen level again after a specific time period.

These guidelines need to be translated and adapted when an IT-based system is used. For example, the required steps to provide IT-based care services for the blood pressure task, which was introduced in the previous section, are:

- Care-receivers should be reminded to attach the blood pressure measurement tool and measure their blood pressure themselves.
- If the care-receivers ignore the reminder to attach the measurement tool, the reminder should be repeated after some time.
- The reminder message, number of repetitions, and their modality can be personalized based on individual requirements.
- If the measured blood pressure value is high/low, the relevant care-giver should be informed.

We call this activity structure for handling a generic homecare task, a *treatment pattern*. In the case of medical protocols and guidelines, these treatment patterns are generic in nature and can be configured to satisfy the requirements of a specific elderly person. After the first interview, we returned to our lab and used the information contained in the medical protocol (guidelines) and the information collected from the first interview to define treatment patterns for each of the four identified homecare monitoring tasks.

After defining the treatment patterns in the lab, we conducted a second interview with the same care-givers. We showed them our proposed treatment patterns for the identified tasks (based on the first interview), and asked them to refine these patterns. We use BPMN [125] like notation for representing the treatment patterns. The reason behind this decision is that the medical protocols resemble closely to a workflow, i.e., both of them provide a depiction of a sequence of operations that need to be performed to perform a task. Moreover, the workflow-based techniques were developed to support communication between technical and non-technical users.

We used a shopping example to introduce the BPMN notation to the care-givers. Doing shopping is a perceptible scenario, since everybody experience it in his everyday life. As illustrated in Fig. *3-1* (we presented this shopping process diagram in Dutch language to the care-givers), in this example, the process starts by making a list of required products, then going to the Albert Heijn (AH) supermarket, if all the required products are there, then the shopper pays, otherwise he/she leaves the AH and goes to the Lidl supermarket. To pay, if the amount is larger that 50 Euro, he/she pays with a debit card, otherwise he/she pays in cash and the process ends. This example and its process diagram helped us to describe the essence of activities, decision rules, and data objects in a business process to the care-givers. After describing the shopping process, not only did the interviewees understand it, but as a proof of this they pointed out a life-lock in the process, which we did not notice.

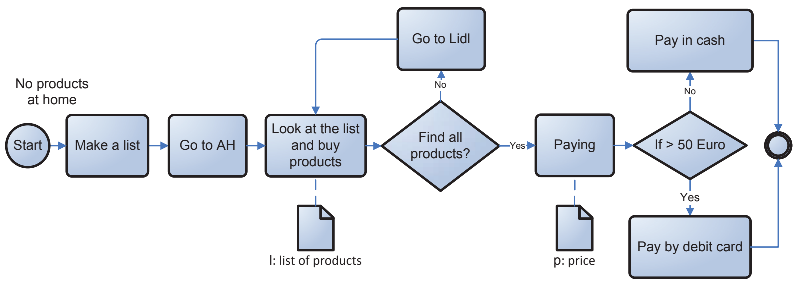

Figure 3-1 : A shopping example

Each treatment pattern consists of several activities (referring to the use of services) and their configuration parameters (annotated as a data item for each activity). Beside activities, a pattern contains some decision points to specify the behaviour of the pattern at runtime. A care-giver can easily configure the plans by specifying/modifying the configuration parameters' values. At run-time, the processing of patterns are triggered half an hour prior to their scheduled time in an agenda. This is because there is no medical protocol where a care-receiver should receive a reminder more than 30 minutes before a scheduled deadline. So it is safe to start any process 30 minutes before the scheduled time.

Fig. *3-2*, shows the treatment pattern of the blood pressure monitoring task. The pattern has three basic services namely $Bloodpressure$, $Reminder$, and $Alert$ and each service has several configuration parameters (some with default values). Each pattern also has one $Agenda$ service for scheduling the tasks which we exclude from the patterns in order to increase readability. The details of each basic services and their configuration parameters are discussed in Section 5.3.1.

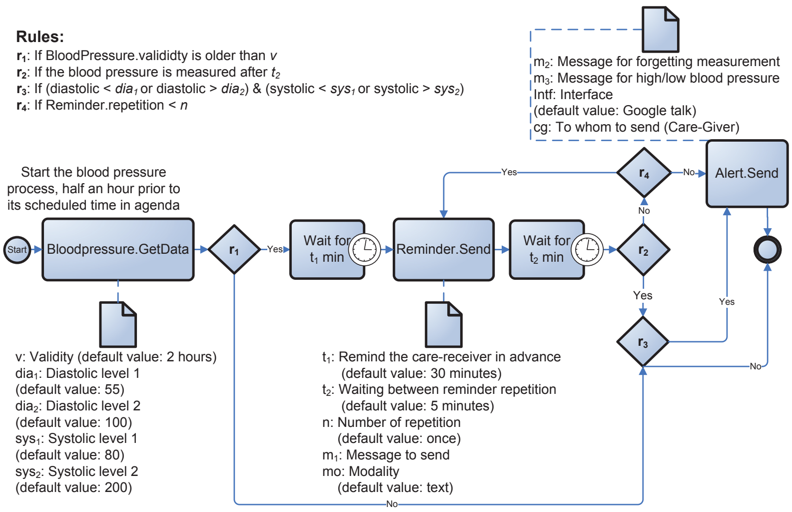

Figure 3-2 : Treatment pattern for the blood pressure monitoring task

The pattern also has four decision points to specify run-time behaviour using the values of configuration parameters, as configured by care-givers. These parameters are: r_1 indicates that if the latest measured blood pressure is older than v (validity) hours, then the care-receiver needs to measure this care-receiver's blood pressure, otherwise the latest value is still valid. If not valid, a reminder with m_1 message will be sent to the care-receiver, t_1 minute in advance (since we have agreed to start the processes only, at most, half an hour before its scheduled time, a reminder can be sent 30 minutes prior to its time).

If the care-receiver ignores the first reminder, r_4 indicates that the reminder should be repeated n times, and r_2 indicates that the waiting time between each reminder repetition is t_2 minutes. Note that the values of t_1, t_2, m_1, and n could be different for different care-receivers. They can also be different for a given care-receiver in various contexts. For example, considering location as context information, if a care-receiver is outside of his home, he should be able to receive the first reminder earlier and the number of repetitions should be higher comparing with the situation when he is inside his home. Again we preferred to exclude such information in the pattern for clarity; however, we discussed these situations with the interviewees and agreed to include them in the design of the tailoring platform and its GUI. Moreover, the reminder message could be sent via different modalities (such as, text, audio, or video) depending on the care-receiver's abilities or his contexts. For instance, he may prefer to have a textual message rather than a voice message when he has a company.

Rule r_3 is a specific decision point for the blood pressure monitoring task where an alert should be sent to a care-giver based on the measured blood pressure and predefined $dia1$, $dia2$, $sys1$, and $sys2$ threshold levels. There is a second type of alert, when the care-receiver ignores the reminder and does not measure his blood pressure. Therefore, the $Alert$ service has two different messages as configuration parameters: m_2 when forgetting measuring and m_3 for high/low blood pressure. An alert message can be conveyed through different interfaces such as SMS, Google talk, or phone depending on the care-giver's preferences.

Figures *3-3* and *3-4* present the treatment patterns of the $oxygen$ $saturation$ monitoring and $weight$ monitoring tasks respectively. These patterns are very similar to the $blood$ $pressure$ monitoring's pattern, except that the oxygen saturation monitoring's pattern has an $Oxygensaturation$ service and weight monitoring's pattern has a $weight$ basic service instead of $Bloodpressure$ service. They also have different rules r_3 as a decision point for detecting hazard situations when the vital signs are too high/low.

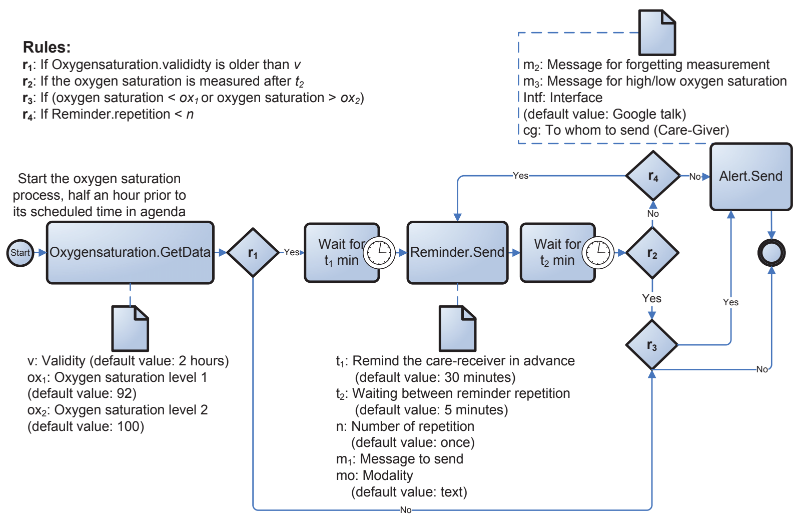

Figure 3-3 : Treatment pattern for the oxygen saturation monitoring task

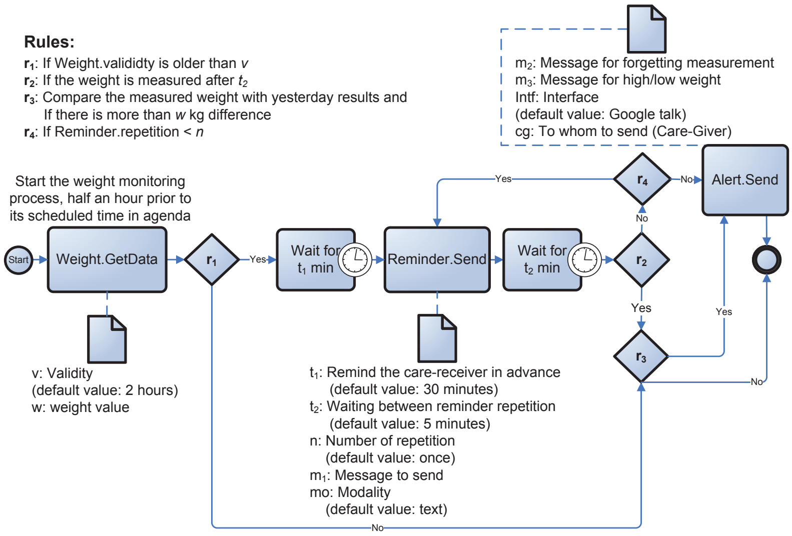

Figure 3-4 : Treatment pattern for the weight monitoring task

Fig. *3-5* illustrates the treatment pattern of the *medication intake* support task. This is also similar to the aforementioned patterns, however, there is no measurement activity and the task only checks if the care-receiver has taken his medication or not. Therefore, it only rise an alert when the care-receiver ignores the reminder and forgets to take his medicine. Moreover, before sending the first reminder, it enables a dispenser so the care-receiver can take his medicine on time. This prevents taking medication at incorrect times.

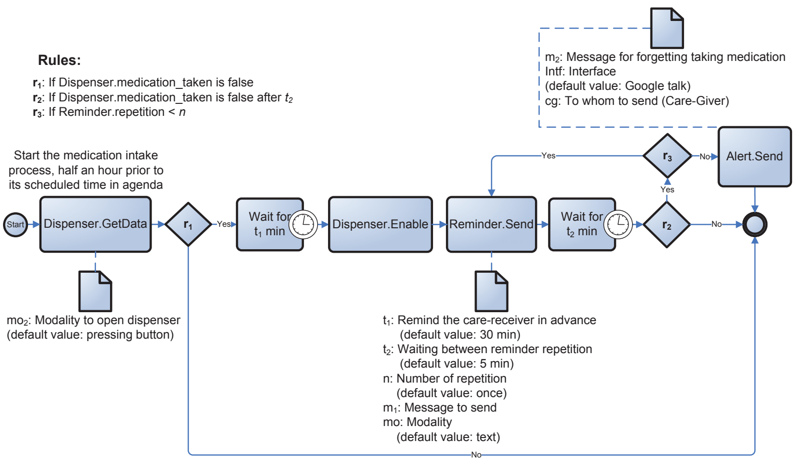

Figure 3-5 : Medication intake support treatment pattern

3.4 Summary

In this chapter, first, we described the case study that is used in this work. Then, we reflected on the results of the first two interviews that we have conducted in the case study with professional nurses. As a results of the first interview, we identified different roles in the homecare domain. Moreover, we identified a number of common homecare tasks performed in the homecare domain. Later, we have selected four of the identified tasks, and implemented the corresponding care services to be evaluated in the case study.

Further, based on the results of the first interview and the medical protocols, we defined the treatment patterns of each identified task. We make use of the treatment patterns as a starting point for the tailoring process. Finally, after defining the treatment patterns in the lab, we conducted the second interview with the same care-givers. We showed them our proposed treatment patterns for the identified tasks and asked them to refine these patterns.

Chapter 4

Risk Driven Requirements Specification (RiDeRS) *

"Change is not made without inconvenience."
— Samuel Johnson

Use of IT in providing homecare services to elderly people is expected to enable the care-givers to take care of more care-receivers at the same time. It is also expected that use of IT will increase the quality of services by providing services round-the-clock and will support independent living of the elderly. However, IT-based care systems can also introduce new types of risks and safety challenges, such as those related to availability and accountability. This could potentially lead to a decline in use of IT-based homecare systems in practice. In order to prevent this, we propose a method to identify potential risks of using such a system, and to specify additional requirements for the system in order to mitigate or prevent these risks. We validate the usability

*This chapter is based on the following papers:

– Mohammad Zarifi Eslami, Brahmananda Sapkota, Alireza Zarghami, Eelco Vriezekolk, Marten van Sinderen, and Roel Wieringa. Risk Identification of Tailorable Context-aware Systems: a Case Study and Lessons Learned. In The Proceedings of The CAiSE'12 Forum at The 24th International Conference on Advanced Information Systems Engineering (CAiSE), volume 855 of CEUR Workshop Proceedings, pages 40-49. CEUR-WS.org, 2012 ([173]).
– Mohammad Zarifi Eslami, Brahmananda Sapkota, Andrea Herrmann, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Risk Driven Requirements Specification (RiDeRS) of IT-based Homecare Systems. In The Proceedings of The CAiSE'13 Forum at The 25th International Conference on Advanced Information Systems Engineering (CAiSE), 2013.

and potential utility of our approach by three experiments using the case study in the homecare domain. We discuss whether the proposed approach can be generalized for use in the wider class of adaptive critical systems.

In Section 4.1, we present the motivation for the proposed method. In Section 4.2, we present some of the existing risks in the current situation of the homecare domain and discuss whether an IT-based system can mitigate them. In Section 4.3, we describe the concepts and process of the proposed method and in Section 4.4, we present the results obtained from applying the method in the case study. In Section 4.5, we present the initial validation of RiDeRS and in Section 4.6 we discuss the lessons learned and generalizable knowledge. In Section 4.7, we discuss related work and finally, in Section 4.8, we conclude the chapter.

4.1 Background

The use of IT-based homecare systems can have several benefits, such as (1) increasing productivity of care-givers, which is needed in the face of the aging population in the coming years, and (2) improve the quality of care and the quality of life of the elderly. The European Council recognises improvement of patient safety as one of the benefits of using eHealth systems [153]. However, IT-based care services can also introduce new types of risks, as they (partially) reduce the involvement of the care-givers in the process of providing care to the elderly.

In the literature, the concept of risk has been defined differently in different domains. For example, it has been defined in disaster management as “the combination of the probability of an event and its negative consequences” [155], in IT security as “the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization” [78], and in toxicology as “expected frequency of occurrence of a harmful event (death, injury, or loss) arising from exposure to a chemical or physical agent under specific conditions” [41]. Nevertheless, there is a common understanding in all these definitions: “risk is a combination of the likelihood that a threat will occur and the severity of the resulting impact”.

In the context of this thesis, we use the simple dictionary definition of *risk* as “the possibility of loss or injury”[1]. We do not assume that probability, loss or injury are quantifiable, since whether they are quantifiable strongly dependant on the environment. However, we do

[1] http://www.merriam-webster.com

assume that the system under consideration has users who can be hurt, and so there are risks of loss or injury. Users are biological or legal persons who can gain or lose something by the actions of the system, and include patients, their family, nurses, and other health care actors.

This means that the concept of risk is user-dependent: what hurts one user may not hurt another one. And what one user would consider an injury may be considered as acceptable pain by another. This means that risk assessments are context-sensitive, and are not transferable from context to context, but must be repeated for every context. If the users of the system change or their desires or vulnerabilities change, we may have to repeat the risk identification process.

Earlier work [157, 30, 109, 101, 119, 68, 50, 100, 44] mainly focuses on privacy and security risks of using an IT-system. We argue that an IT-system is not only subject to privacy and security risks, but also availability, safety, and accountability risks [10]. *Availability* is the readiness for correct service [10], or more elaborately, accessibility and usability when needed by an authorized entity [129]. We define *availability risk* as "the possibility of the system or its services not being available when needed". Not being available when needed has further consequences, which can cause safety risks. We define *safety risk* as "a possible threat to the safety of a user" (where safety is the absence of loss or injury).

Accountability is the availability and integrity of the record of the identity of the person who performed an operation [10]. In our case, we require that every performance or non-performance of an action by the system is attributable to a person, who is accountable for this action or inaction. Then, we define *accountability* as "the allocation of responsibility in case of materialization of a safety risk".

For example, in a voice-based notification message service, an availability incidence arises when the user accidentally touches the *switch off* button instead of the *acknowledge* button because of his poor eyesight. The health consequences that might arise if the service not being available could lead to a safety risk. Accordingly, either the system designer, the implementor, or the user should be assigned accountable for the consequences of this action. In this case, it is not clear who is accountable for this incident and this introduces an accountability risk.

Materialization of these risks could degrade the quality of life of individual users and can even prevent the adoption of such a system in real-life. Assessment of these risks must be performed up-front, as part of requirements engineering, and must continue during the lifetime of the system to discover and mitigate unknown or unperceived risks during early requirements engineering [131]. We propose a Risk Driven

Requirements Specification (RiDeRS) approach to elicit requirements of a system by identifying the risks of using such a system as a first step in a requirements engineering process.

Risk-based requirements engineering complements requirements engineering aimed at positive goals. RiDeRS can be used in early requirements engineering, but we see no reason why it could not be used at any point in the system's life cycle to assess risks and to elicit additional requirements.

There are few approaches to identify risks of a system during the requirements engineering process. These approaches are mainly goal-oriented, i.e., these approaches elicit requirements of a system by identifying the user's goals and by navigating through a goal tree from higher level to lower level goals [29]. However, identifying those requirements from a user's goals is quite a challenge, because for many users it is often difficult to express what their requirements are [126]. Moreover, the requirements specified based on a users' goals are usually expressed vaguely (e.g., the system has to be easy to use). Such requirements are neither detailed enough for being implemented by a developer nor specific enough for being verified by a tester, nor are they useful for cost estimation [66].

The proposed approach is adapted from the MOQARE approach defined by Herrmann and Paech [66] in which quality requirements are explored to identify business damages (via misuse cases) and their countermeasures. However, MOQARE is business environment oriented and only considers business goals. In contrast, we are focussing on an environment of users (e.g., the elderly and nurses) and want to include risks due to assumptions about these human users. These assumptions about the users, other systems, and even forces of nature should be considered in order to sufficiently identify risks and determine countermeasures.

In RiDeRS, we consider users' properties in addition to their goals in order to identify a list of possible risks and to specify the requirements which could prevent or mitigate these risks. This also helps other stakeholders (e.g., care centers) to perform risk-benefit analysis and to decide whether or not to use the system.

What is new in RiDeRS is that it is a goal-oriented method that aims to balance the assumptions about the environment with the requirements on the system. It focusses on safety and it is simple to use in practice, as we will show in our case study. We apply RiDeRS in the case study to illustrate its usability and utility. RiDeRS was applied before the end-users actually used the tailorable IT-based homecare system, i.e., the end-users were not aware of any risks resulting from the introduction and use of the system.

We performed the third series of interviews using RiDeRS with the same care-givers who participated in the previous two interviews to identify and analyse the risks of using our IT-based homecare system and consequently to identify further requirements on the system. The analysis of the results of this interview will be the basis of our hypothesis in Section 4.6 that RiDeRS is generalizable to a wider class of systems.

4.2 Existing Risks

Before explaining the RiDeRS method and discussing new risks that might be introduced due to use of an IT-based homecare system, in this section, we discuss the risks in the current situation and determine whether these can possibly be decreased using the system. We identified these existing risks while we interviewed care-givers to identify the common tasks generally performed in the homecare domain (the interview and its results explained in Chapter 3).

- **Forgetting to Treat Patients:** The care-givers usually follow a routine schedule in providing care services based on the medical protocols and doctors advice, for example measuring blood pressure every morning right after the care-receiver wakes up. However, it is likely that a care-giver forgets to measure blood pressure for a specific care-receiver or measures it late which might result in unreliable readings. An IT-based system can be used to remind a care-receiver to attach the measurement tool whenever it is required.

- **Observation Error**: The care-givers give reports about the situation of the care-receivers to the doctors, family members, pharmacy, etc., based on which the authorised stakeholders take appropriate actions. For example, a doctor can prescribe new medication based on the current situation of the care-receivers, then the pharmacy can provide it, and the care-giver can give the medicine to the care-receivers. However, while measuring the care-receiver's vital signs such as blood pressure, the care-givers can make mistakes in reading/writing of values, which can affect the diagnosis made by the doctor. An IT-based system can automatically create a correct report based on the measured data which is directly accessible by a doctor.

- **Action Error**: The care-givers can make a mistake in providing care services to the care-receivers. For example, a care-giver can provide an incorrect medicine to a care-receiver. An IT-based system can be used to provide medication through a digital dispenser filled with medicine by a pharmacy based on a doctor's prescription. We should take to account that there is still the possibility of the pharmacy making a mistake when filling the medicine dispenser.

- **Overlooking the Medical Protocols**: Nurses should follow the medical protocols in providing care services, and are examined yearly to prove that they still recall them. However, a care-giver may overlook these protocols and make an incorrect decision. An IT-based system ensures the conformance with medical protocols, because those protocols are embodied in the treatment patterns.

- **Conflict in Medication/Treatment**: Care-receivers typically receive multiple medications which are prescribed by different specialists. A doctor may prescribe a medicine without considering other prescribed medications that can have negative effects on other diseases/medicines. It is also common that for a specific disease, a doctor will prescribe a new medicine which has better effect, but without stopping the previously prescribed medicine(s). It is also possible to have a conflict in treatment(s). For example, a care-receiver can utilize advise or treatment from different professionals, such as a family doctor, hospital doctor, and physiotherapist, which each may in itself be correct, but not optimal when used in combination. This risk can be easily detected by an IT-based system, if there are predefined rules for conflicting medicines/treatments. The rules are derived from medical protocols or care providers recommendations.

4.3 RiDeRS

One of the key motivations for replacing manual activities with automatic IT-based systems is *reduction of human error*. However, many practical experiences have shown that in reality, automation may produce new sources and types of errors [162]. This is also true when an IT-based homecare system is used to provide personalized IT-based homecare services.

Critical systems such as IT-based home care systems have stringent availability and safety requirements and a failure can threaten human life [143]. Because of these threats, users may lack confidence in using or adopting such systems. Therefore, it is necessary to build the confidence of the users well before introducing such systems in practice. Requirements engineering is one means to achieve this. To build confidence among users, the users must be shown that the system matches their needs.

For the proper functioning of a system, the properties of the environment (including its users) also needs to be considered. For example, if the system is designed to interact with a user through a touch panel and the user is an elderly person with Parkinson's disease, then the user may not be able to touch the panel in the right positions. Consequently, the system will behave differently than desired. To prevent such undesirable system behaviours, we introduce a risk based approach to identify additional requirements for the system in order to ensure the safety of end-users while offering ease of use. In the remaining of this section, we describe RiDeRS concepts and the process that was used.

4.3.1 RiDeRS Concepts

As shown in Fig. *4-1*, we assume that there is a *system* which works in an *environment* (including the end-users of the system). The system provides *services* which are consumed by *end-users*. The designer of the system makes *assumptions* about the environment of the system, and specifically about the end-users. These assumptions guided the design and thus underlie the services. Services have associated *goals*, as foreseen by the designer, which can be achieved **if** the assumptions made by the designer are correct.

End-users also have goals, which they try to achieve by consuming the services provided by the system. However, if the assumptions made by the designer are incorrect, an end-user may not be able to achieve his/her goals. Therefore, assumptions imply a *risk*, namely that they do not match the actual properties of end-users, thus preventing end-users from achieving their goals.

Such risks may be identified at any point in the lifetime of the system, and suitable *countermeasures* may be proposed to mitigate the risks. Countermeasures are subsequently implemented in the system or in the environment (possibly in both), resulting in a modified system with a changed specification or a modified environment with changed properties.

The designer makes certain assumptions regarding the use of the system. Whether these assumptions are correct or not for a given instance of use, determines the suitability of the system and its services in that instance. With respect to the users of the system, we classify the assumptions into six categories:

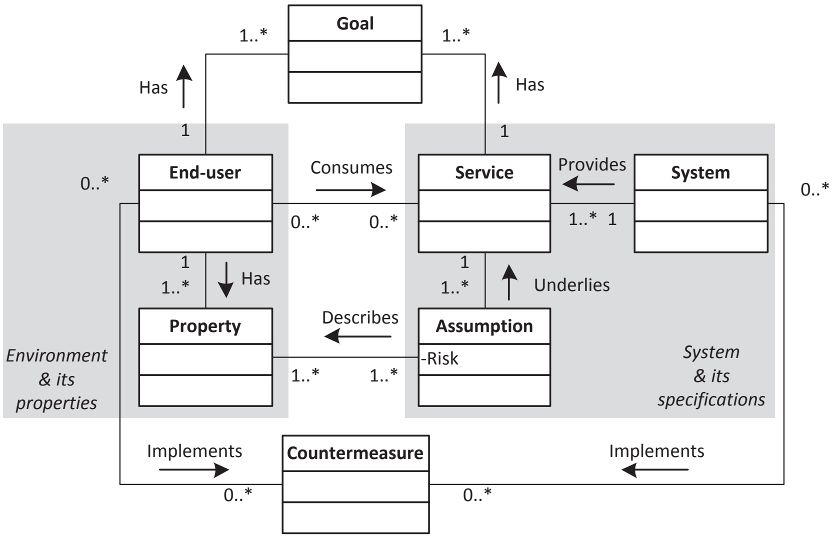

Figure 4-1 : Conceptual model RiDeRS approach

– **Capability assumptions** relate to the capabilities (and limitations) of the end-user. We are only interested in capabilities that are needed to access a service through the provided interface. For example, in the case of a medicine dispenser service, the designer may assume that the end-user can operate a touch panel to control the dispenser. This assumption is not true if the user has Parkinson's disease. The risk is then that the user may not be able to touch the panel in the right positions. Consequently, the system will behave differently than desired.

– **Procedure assumptions** relate to the procedures followed by the end-user to access and consume a service. The designer of a system assumes a specific procedure for each service. If a user is to use the system (to achieve some goal), then he/she should follow the procedure assumed by the designer. This assumption is risky because the user may/could not follow this procedure when using the system. For instance, in the case of a blood pressure measurement service, the designer may assume that the end-user follows the proper procedure for blood pressure measurement, namely that the measurement instrument is placed around the end-user's left arm and he is in a sitting position. This assumption is not true if the user is an elderly person who forgets the correct procedure and attaches the blood pressure cuff to his wrist. The risk is then that the measured values are not precise or reliable.

- **Location assumptions** relate to the location of the end-user carrying a device with sensors that measures a location parameter for a context-aware service. For example, the designer assumes that the location of the user can be determined with a GPS receiver in the user's mobile phone. This assumption is not true if the user does not carry a suitable phone. The risk is then that a location-based service is applied to the wrong location, and the user is not able to consume the service or consumes a service that is not optimal or even appropriate for his needs/goal.

- **Identity assumptions** relate to the identity of the end-user who uses a device with sensors that measures the necessary user context parameters. In the blood pressure example, the designer assumes that the measured values from a specific blood pressure meter are in fact those for the intended care-receiver. This assumption is not true if the user takes an instrument which does not belong to him (e.g., when an elderly couples live together in the same house). The risk is then that the reported blood pressure values are not correct and could result in incorrect health related decisions.

- **Needs assumptions** relate to the end-user having needs matching the user context in which the context-aware service is being offered. For example, the designer assumes that the user needs an audio reminder to take a medicine every five minutes until the medicine is taken. This assumption is not true if the user has not forgotten to take the medicine, but he has an urgent task to finish before he can go to the medicine dispenser. The risk is then that the user will turn off the reminder service because it irritates him, and subsequently forgets to take his medicine.

- **Timing assumptions** relate to the availability of the end-user (at a specific place/time). For any user-triggered service, the user should be available to initiate the service and to consume the offered service; and for any context-triggered service, the user should be available to accept and consume the offered service (when and where it is offered). For example, the designer assumes that the user can react to a reminder in say 5 minutes to measure their blood pressure and after 3 reminder repetitions the system raises an alert assuming that the user ignored the reminders. This assumption is not true if the user received the first reminder correctly and acted accordingly, however, because he is far from the place of the measurement, it takes him longer than expected to act. In this case, the system raises an unnecessary alert that irritates the care-givers.

Timing assumptions are only apply to user-triggered services, if the services are also time-scheduled. If services do not have a time schedule (and are also not event-triggered), then there are no timing assumptions.

To summarise, the system has been designed under certain assumptions that can be classified as stated above. Violation of one or more of these assumptions would introduce risks of not achieving the intended benefits of the system to the users. RiDeRS was developed to check and balance exactly such risks. It helps in checking, whether such assumptions are in fact true with respect to specific users of the system, identifying the consequences (risks) of false assumptions, and proposing countermeasures (additional requirements on the system) that would minimize adverse consequences w.r.t. goal achievement for the end-users.

4.3.2 RiDeRS Process

We define RiDeRS based on the existing quality requirements approach MOQARE [66]. In MOQARE, quality requirements of a system are specified based on identification of misuse cases. In RiDeRS, we also specify quality requirements by identifying risks; however, we specialize as well as extend MOQARE. We consider misuse cases which are caused due to incorrect assumptions by the designer about the users (and the use) of the system. Fig. *4-2* illustrates the RiDeRS process steps in BPMN notation.

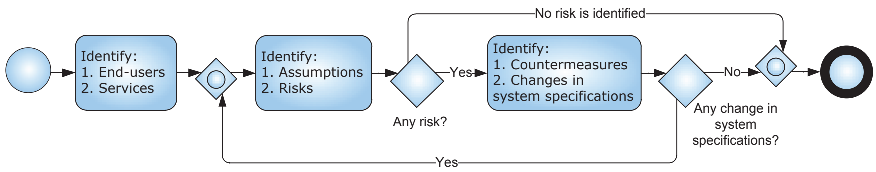

Figure 4-2 : RiDeRS process

The process starts by identifying the users of the system and, for each user, the services which a user wants to consume. Then it iterates over identifying, for each user, the assumptions underlying the services, the risks caused by these assumptions, the countermeasures for mitigating these risks, and the changes to the system specifications due to these countermeasures.

To help the requirements engineer, we formulated a list of questions that the requirements engineer should ask to check whether the assumptions are in fact true with respect to specific users of the system:

- What is the minimum capability that users must have to use this service?
- What procedure must users follow to use this service?
- What does the service assume about the physical location of the user?
- What does the service assume about the identity of the user?
- What assumptions does the service make about the real needs of the user?
- What does the service assume about the time of service consumption?

Regarding each assumption, further questions can be raised. For example:

- When would these assumptions be violated?
- Can the system discover these assumptions are violated?
- Can the system still be used when these assumptions are violated?
- Is the system still useful to use if these assumptions are violated?
- What if these assumptions are violated? Who is accountable for those hazards that materialize?

After specifying proper countermeasure for each risk, if the countermeasure change system specifications, then the system needs to be analyzed again for new sources of risks. In this regards, we suggest the following "good practices" that we have adopted in our case study:

- a) For each risk, identify a countermeasure which is implemented by the system in addition to a countermeasure which is implemented by the users. This promotes the consideration of trade-offs between countermeasures to be implemented by the system and by the users.
- b) After the identification of countermeasures, reduce the number of different countermeasures by considering their overlap and factor out common aspects.

We present the results of RiDeRS in the form of a table (Table *4-1* gives an example of such a table), in which the columns represent different services and their assumptions, and the rows represent the user's properties. The text in each cell represents the risks (violated assumptions resulting from the combination of assumptions and properties) and their countermeasures. If a risk is identified, we expect at least one countermeasure to exist.

We limit the scope of RiDeRS as follows:

– Traditional requirement engineering approaches have already been applied and system requirements and services have been specified.

– We do not consider financial risks of introducing and using IT-based critical systems (in our case homecare systems).

– In this work, we only discuss assumptions about the end-users and do not consider assumptions associated with other entities or events in the system's environment, such as communication infrastructures.

– The system technically works as designed, so that the risks due to the malfunctioning of internal components of the system (risks caused by interactive complexity [97]) are out of the scope of this work. We even assume that the system/services are correctly designed under the given set of assumptions and requirements. The risks we consider come from incorrect assumptions and incomplete statements of requirements. More specifically, we are interested in the risks related to the use (or non-use) of services offered by the system to its users.

RiDeRS is a general method which can be applied to any system embedded in a domain where the assumptions about users can generate risks. To extend our experience regarding RiDeRS, and to evaluate if using RiDeRS we can identify risks and countermeasures in practice, we apply it in the case study. In the next sections, we present the applications of RiDeRS to the case study, and then we return to general observations and motivations for RiDeRS in the light of our case study experiences.

4.4 Applying RiDeRS to the Case Study

We used RiDeRS to elicit possible risks in the scenarios of the case study. The result of this application of RiDeRS is presented in the following subsections.

4.4.1 Identifying End-users

The homecare domain is complex and involves various stakeholders with diverse interests (e.g., insurance companies, government, etc.). As indicated in Section 4.3, we include in the analysis those stakeholders who will directly interact with the IT-based homecare system. To analyse the risks, we begin with the identification of users in a homecare domain, where care services are provided both with and without using the IT system. We identified three main stakeholders before introducing an IT system in the homecare domain: *care-givers*, *care-receivers*, and *care centers* as depicted in Fig. *4-3*.

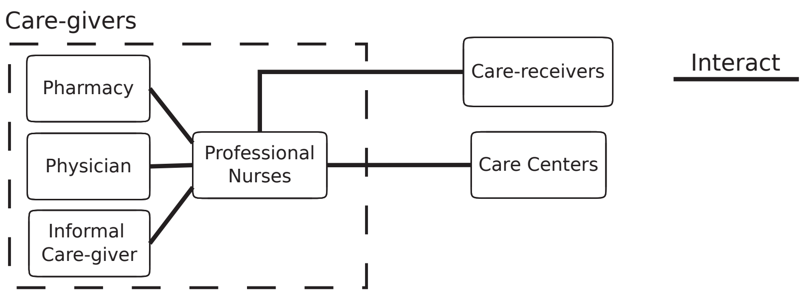

Figure 4-3 : A care center context diagram before introducing an IT system

Anyone involved in providing care to the elderly (*care-receivers*), is considered a *care-giver*. Those who can provide care or interact with the elderly include: professional nurses, informal care-givers, and physicians. *Care centers* are institutions who pay the care-givers and provide facilities to take care of the care-receivers. Care centers define medical protocols providing guidelines for taking care of care-receivers, which should comply with the national medical protocols as defined by the relevant government. When carrying out homecare tasks, care-givers must follow these medical protocols.

As shown in Fig. *4-4*, after introduction of the IT system, four new types of stakeholders are introduced to the system and are considered when analysing risks (as discussed in this chapter). These new types of stakeholders are *IT specialists*, *third-party service providers*, *infrastructure providers*, and *hackers*.

An *IT specialist* is a person who can install, test, operate, and maintain the IT-based homecare system. IT specialists are

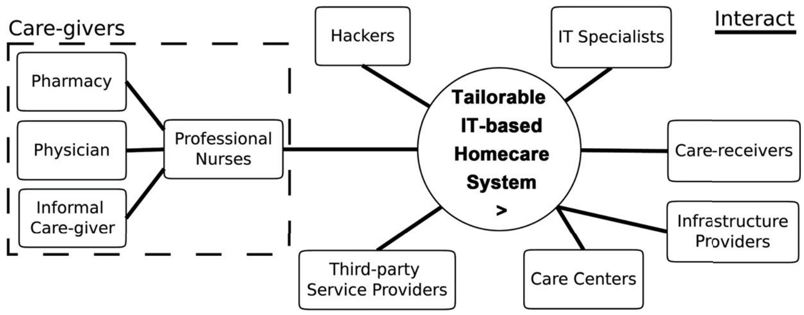

Figure 4-4 : A context diagram of a care center which is equipped with an IT-based homecare system

responsible for defining the treatment patterns based on existing medical protocols and care-givers recommendations and refining them based on operational experiences and test results.

Third-party service providers own and manage services (such as blood pressure measurement, location determination, and medication dispensing services) which can be used and composed by the provisioning system to provide desired services to the care-receivers. These providers are located outside the care center and their services are accessible through the Internet. *Infrastructure providers* are responsible for providing the necessary infrastructure to realise the system such as Internet connecting and power. *Hackers* are individuals or organizations who break into the system or its network and violate its function or extract information.

4.4.2 Identifying Goals and Desired Services

The system provides various services to different users. For example, it provides a tailoring service for the care-givers which enables them to create a service plan for a specific task (e.g., measuring blood pressure) and individual care-receivers by specifying values for configuration parameters, e.g., via a GUI. This service plan, for example for the blood pressure measurement task, indicates when a care-receiver should measure their blood pressure, how many times and when a reminder should be sent, when and to whom an alert should be sent in case of the care-receiver or care-giver ignoring reminder messages or alerts for high/low blood pressure values. The system provides reminder and measurement services to the care-receivers and an alert service to the care-givers.

Among the identified users we choose the *care-givers* as an example for this step. A care-giver intends to achieve two goals by using the system. *Goal 1*: Define personalized services for specific care-receivers using the tailoring platform. *Goal 2*: Provide safe and appropriate services to care-receivers and generate alerts if care-receivers are in a hazardous situation. To achieve these goals, a care-giver interacts with a tailoring platform of the system through a GUI to create a personalized service. They also use a smart-phone to receive alerts. To achieve these goals, two services should be used by the care-giver: *Service 1*: A tailoring service and *Service 2*: An alert service.

4.4.3 Listing Assumptions

For the tailoring service, a care-giver should have the following capabilities: she should have enough time to create a personalized service, she also needs to have some IT knowledge to interact with the GUI and enough domain knowledge (i.e., information about the situation of the care-receiver and blood pressure characteristics) to enter proper values for configuration parameters. She has to follow the following procedure: through a GUI, she should first select the name of the care-receiver from a menu, then a specific task (e.g., blood pressure measurement) and select or enter values for all required configuration parameters. As context assumptions, she has to create the service at least half an hour before its intended execution time.

For the alert service, as capabilities, she should know how to use a smart phone (e.g., increase/decrease its volume, drag to see the alerts, etc.). As procedure, she should confirm receiving alert messages by pressing a button on the smart phone. As context, she should carry the intended alert device, enable its alert beeper, and be close-by to react to the alert when it is needed (if a care-giver receives an alert, she should be close enough to arrive at the care-receiver's location within the couple of minutes).

4.4.4 Identifying Risks and Countermeasures

We use the goals, services, three types of assumptions, and user's capabilities identified based on the case scenario and described in the previous subsection to identify risks and countermeasures. We acknowledge that one can think of more goals, services, and assumptions and as a result derive a more comprehensive list of risks and countermeasures. However, to show the usability of our approach, we use two services (tailoring and alert services) and two care-givers' capabilities: *Capability 1*: Be able to take care of a limited number of care-receivers, and *Capability 2*: Have only limited IT knowledge.

Since we use two services and two capabilities for the care-givers, we could expect at least four risks and their countermeasures. However, this does not mean that these are the only possible risks and countermeasures. There can be more than one risk for the combination of services and capabilities, and each risk can have more than one countermeasure. Furthermore, the same countermeasure could be used to mitigate more than one risk. In the following paragraph, we explain how we arrive at and what we mean by the illustrative risks and countermeasures we entered in the cells of Table *4-1*. We used and show only capabilities in this table, other assumptions can also be similarly used to analyse related risk.

If the care-giver takes care of a limited number of care-receivers and wants to provide user specific services to these care-receivers, there is a risk of *forgetting to create a personalized service for a care-receiver for a specific task*. This risk can be mitigated by the countermeasure which requires the *system to check for missing personalized services and generate a remind if one should be created*. Similarly, we identify *Risk 2*, *Risk 3*, and *Risk 4*, and their associated countermeasures. Here we do not repeat the process, to check whether the countermeasures introduced new risks (the feedback loop in Fig *4-2*).

Using RiDeRS tables enables us to easily identify risks. In the following, we present some of the potential risks for each stakeholder that interacts with the system. A viable homecare system should mitigate these risks:

– **Care-givers**

Incorrect Configurations: Setting incorrect values for the configuration parameters, e.g., setting higher/lower values for the blood pressure threshold. This type of risks occurs due to the tailorability of the system. It is important to consider what can be tailored and what not. This risk happens when a user enters a configuration value which is not in the context model. For example, suppose that the location of a care-receiver is modeled as only *inside the home* and *outside the home*. A care-giver may insert a value *at park* as the location value. This type of risk can easily be prevented by limiting the possible configuration values, e.g., by checking the value against the model, decreasing the likelihood of this type of risk. Limiting the possible values and allowing a care-giver to only select among those values also eases the tailoring for the care-giver. However, at the same time this can also limit the the possibility of tailoring, for example, if the possible values for the diastolic level is limited to a number between 170-220 by the developer, it is not possible to set the diastolic threshold value to 230 for a specific care-receiver who has usually high blood pressure.

Table 4-1 : Applying RiDeRS in the case study with care-giver as a user

Required Services to Achieve Goals	User Capabilities (violated system capability assumptions)		
	Capability 1: Be able to take care of a limited number of care-receivers	**Capability 2**: Have only limited IT knowledge	...
Service 1: Tailoring service	**Risk 1**: Forgetting to create a personalized service for a specific task and care-receiver	**Risk 2**: Spending more time than expected tailoring the services, making the use of the system counterproductive	...
	Countermeasure 1: System should check for missing personalized services and inform the user if one should be created	**Countermeasure 2**: System should provide a user friendly interface and a simple method for tailoring the services	...
Service 2: Alert service	**Risk 3**: Forgetting to take the smart-phone with her or ignoring the alert	**Risk 4**: Cannot increase volume or change the alert sound in case they cannot hear alerts	...
	Countermeasure 3: System should provide alternative type of alerts and generate alerts to other care-givers in case of no response to the first alert to the designated care-giver	**Countermeasure 4**: System should support different modalities for alert (e.g., sound, vibration, or both) and increase the alert volume if she ignore it for longer than a couple of seconds	...
...	...	...	...
	...	...	...

Missing Services: Since care-givers have limited time to provide care services, they may forget to tailor a service for a care-receiver for a specific task. This type of risks can easily be mitigated by checking tailored services against the list of care-receivers and possible tasks, and reminding care-givers if it is necessary to create and tailor a service. The drawback of this countermeasure could be disturbing the care-givers with too many notification messages.

Conflict in a Task with Multiple Context Information: This type of risk occurs due to bad reasoning of the system regarding a task with multiple instances of context information. The system should decide what to do based on context information, however there will be a conflict if there are two different actions for different context information. For example, a reminder should be sent to a care-receiver's mobile phone when he is outside of the home, and he should not receive any reminder message on his phone when he is with somebody. So if he is outside the home and is accompanied by someone, there will be a conflict regarding sending the reminder message. Since this conflict occurs in one task (for example, sending a reminder), the system can be designed to detect such a conflicts and inform the care-giver in advance. This type of risks can be prevented by prioritizing the rules, i.e., saying which rule has higher priority.

Conflict of Different Tasks: This type of risk occurs because performing different tasks would result in a conflict. An elderly person typically suffers from a combination of diseases and hence, a care-giver may ignore a suggestion from the system and potentially create conflicting services for different tasks of the same care-receiver. For example, in one task, a care-receiver is asked to take his blood pressure at 8:00 AM while in another he is asked to take a walk outside at the same time because of his COPD problem. Conflicts of important tasks (such as monitoring vital signs) should be prevented by the system, however conflicts of unessential tasks such as social activity services could be ignored so a care-receiver could decide which event to participate in.

Too Little Information: Using IT-based system could decrease the face-to-face communication between care-givers and care-receivers, thus the care-giver's knowledge about their care-receivers' situations will be more limited.

– **Care-receivers**
Cheating with the System: Lying to the system whenever a confirmation is needed, for example, a care-receiver may lie about taking their medicine.

Increased Loneliness: Feeling lonely is a common issue among elderly people. This could be worsened by using a system such that face-to-face communication is limited only to hazardous situations.

– **IT Specialists**
Inappropriate Treatment Patterns: Incorrectly translating the medical protocols to treatment patterns or providing incomplete patterns. The possibility of this risk occurring is very low, because this a one-time activity (with only minor changes per year) and the incompleteness or incorrectness of the patterns can be detected during a testing phase.

– **Third-party Service Providers**
Third-parties Service Failure: Service Oriented Architecture (SOA) is becoming popular for designing IT-based systems. A SOA-based system may utilize services offered by different providers. Services provided by the third-party service providers may not function or function improperly. There are usually Service Level Agreements (SLAs) among the partners to assure the availability of the services with the required quality of service. However, in some domains such homecare, which is a safety critical domain, relying on the SLAs may not fully compensate for the risks that occur.

– **Infrastructure Providers:**
Communication/Power Network Failures: The communication or power network may go down.

– **Hackers**
Malicious Action: The care-receiver's data or created services may be altered/stolen. In fact, we do not consider hackers as a new source of risk, since thieves can do similar damage in the existing situation. However, various security methods such as encrypted logs and tunneling of all data traffic should be use in homecare systems to mitigate this type of risk.

4.5 Initial Validations of the Method

To test RiDeRS, we applied it to the scenarios of the case study three times in increasingly realistic ways. In this section we describe the case study and the applications.

4.5.1 From MOQARE to RiDeRS

We started our requirements engineering by applying the MOQARE method defined by Herrmann and Paech [66] to our case, *without* interacting with domain experts, in order to test whether this method was usable in our case study. We slightly adapted MOQARE to our case study by starting from end-user goals rather than business goals, and by focussing on risks rather than misuse cases. This approach was used earlier in the RiskRep method [65], also based on MOQARE. In the resulting adaptation of MOQARE, we start with the identification of end-users' goals and constrains, and then we identify risks and countermeasures.

We learned that MOQARE could **not** be applied as is and required two changes. First, MOQARE represents user goals, misuse cases, and countermeasures in a tree. This tree became too large to be usable, and we replaced it by a one-dimensional tabular representation in which the first column contained user goals and the second one contained user constraints. The tabular format was more convenient than the graphical one in terms of RiDeRS's purposes. Second, after analyzing the results, we could see that the countermeasures themselves could be the source of new risks and we should iterate the process of risk identification and countermeasure generation. So we included this loop in the process. The resulting method was the first version of RiDeRS.

4.5.2 Lab Test of RiDeRS

We next validated the usability of the RiDeRS approach with eleven IT experts who acted as if they were domain experts. The purpose of this experiment were (1) to see whether our claim was valid and users can indeed find risks by using RiDeRS and (2) to improve RiDeRS based on IT-experts' feedback before using it with real domain experts. We acted as consultants and first explained to the subjects the objective of applying RiDeRS. Then we explained how to apply RiDeRS by showing them an example where we apply RiDeRS to the *care-givers* as a user and presented them with the corresponding table (Table *4-1*). Next, we provided them a table to fill the possible risks and countermeasures considering *care-receivers* as a user. The table was pre-filled with the labels of goals and constraints of care-receivers. The task of the participants was to identify possible risks and their countermeasures. In addition to risks and countermeasures, the participants were asked to add more constraints to the table (and to derive further risks) which they deem plausible.

The experiment lasted approximately 30 minutes and all participants succeed in filling in the tables with possible risks. Most of the countermeasures identified by IT experts were ones that should be provided by the system. For instance, we gave "lack of IT knowledge and lack of interest in interacting with the system" as an example of a constraint imposed by care-receivers. Based on this constraint, an IT expert then identified a risk of "not being able or not willing to confirm to the system of receiving services whenever it is needed". He provided a countermeasure, which is: "the system should support some other ways of checking (e.g., video observation) for receiving the services by care-receivers".

We took away two lessons from this experiment. First, the IT experts conceptualized the risk analysis in terms of capabilities and limitations of end users, and we replaced the concept of a constraint imposed by users, with that of (possibly limited) capabilities of users. Second, our tabular representation of risks and countermeasures was still not usable and we replaced it with a two-dimensional table format such as the one shown in Table. *4-2*. The table lists one or more system services vertically, and for each service lists the assumptions that the environment should satisfy for the system service to be consumed by an end user. Horizontally, the table lists properties some users actually have. In the table entries, we list whether this property is a risk w.r.t. the assumption and if so, what the possible countermeasures could mitigate or eliminate this risk.

Finally, because the method now focussed on user capabilities, we restructured the method to follow the assumption-based reasoning from KAOS [157] and Jackson's problem-frame approach [80]: If the environment satisfies assumption A and the system satisfies requirement R, then the system will contribute to goal G. Our assumptions are assumptions about (limited) end-users capabilities, the requirements are countermeasures to mitigate risks associated with these limited capabilities, and our goals are end-user goals to perform some health-related tasks. This updated method was used in the field test of RiDeRS, described next.

4.5.3 Field Test of RiDeRS

We validated the usability and usefulness of this updated version of RiDeRS in the case study with care-givers as domain experts. As with the earlier IT-experts, we explained to the care-givers the application of RiDeRS and how to apply it by showing them the *care-giver* example, but this time we provided them with an improved lay-out derived from the previous experiment. We asked them to fill in possible risks and countermeasures considering the *care-receiver* as a user. All four care-givers of the case study participated in the experiment for an hour.

We made two observations based on this field test. First, it turned out that care-givers could not imagine any risks when the goals and user capabilities were stated abstractly. For example, as a goal, we initially specified “the care-receiver use vital sign monitoring services” and as a capability “he has some disabilities”. We needed to make this concrete and specific, as in for example “the care-receiver uses a medication dispenser service which requires pressing a button” and the capability “he has vision impairment”, so that the care-givers could identify risks and also suggest countermeasures.

The second observation is that in comparison with IT-experts, the care-givers usually provided user-based countermeasures rather than system-based countermeasures. For example, for the capability “lack of IT knowledge and lack of interest in interacting with the system”, a care-giver identified the risk of “fear of using the system”, she identified as countermeasure “training the care-receivers of using a tablet PC and encouraging them to use it by providing some card game applications to play on the tablet PC”. Thus, working with real domain experts in the field yielded not only countermeasures that were new IT system requirements, but also countermeasures that were requirements that end-users must satisfy.

Our conclusion from this first field test is that RiDeRS was usable (by us) in at least one case, and that it was useful, as we identified risks and countermeasures that neither we nor the domain experts had thought of in advance.

4.5.4 Further Extensions of RiDeRS and an Illustration

The field test provided yet another lesson, namely that there are more assumptions relevant for a risk assessment than just the capabilities and limitations of end-users. Analyzing our data, we identified the assumptions listed in Section 4.3.1. We now illustrate these with a fragment of the risk assessment that we did with the domain experts. We should say at the outset that no risk assessment can be complete, and we do not claim completeness for this one either. However, we do claim that using RiDeRS made it possible to identify more risks than would otherwise have been possible, although empirical comparison with other risk assessment methods remains to be done.

Table *4-2* shows a risk assessment for the blood pressure monitoring services for care-receivers as users. If the care-receiver, as an example, *has Parkinson's disease* and wants to *use the blood pressure service himself*, he can introduce *Risk 1*. If *Risk 1* is present, it can be mitigated by *countermeasure 1*. Similarly, we identify *Risks 2-7* and their associated countermeasures. Here we do not repeat the process of checking whether the countermeasures introduced new risks (the feedback loop in Fig. *4-2*). Note that one countermeasure (e.g., *countermeasure 2*) can be relevant for more than one risk, something that can be easily represented in this table.

User properties are non-compositional, i.e., there might be situations in which a combination of two or more properties are applicable to one user and they collectively create a risk that none of them creates on its own. For example, *Risk 5* exists because of *properties 2* and *3*. Another interesting point is that materialization of some risks could cause other risks. For example, in the case of *Risk 7*, the care-givers after some time may ignore the alert messages for this care-receiver, assuming that it is not an actual alert, while the care-receiver might be in real danger.

Classification of the assumptions is indeed useful, when a requirements engineer identifies the assumptions of provided services and fills them into RiDeRS tables. Then together with domain experts they can further fill the table with user properties and accordingly analyze the risks and identify additional system requirements. One can argue that by using RiDeRS, the problem of finding risks is shifted to finding properties instead of solving the problem. However, our experience with the domain experts shows that finding and perceiving risks is not easy for them, but they can easily name a couple of user properties.

A risk assessment consists of risk identification, analysis, and evaluation [76]. In RiDeRS, we only perform risk identification and analysis. To evaluate the risks we could use the categories proposed by u.s. FDA in [156] which are (a) *Major*: death or serious injury of the patient or operator is possible, (b) *Moderate*: minor injury is possible, and (c) *Minor*: no injury to the patient or operator is possible. These categories focus on the goal "survival and health", other goals are considered to be less important. It turns out that care-givers do evaluate risks this way, even without recording their evaluation explicitly in a table.

Table 4-2 : Applying RiDeRS in the case study with care-receiver as a user

Required Services and their Assumptions		User Properties				
		Property 1: Have Parkinson's disease	**Property 2:** Have amnesia and forget things	**Property 3:** Is living with a partner who also use blood pressure service	**Property 4:** Too many reminder messages irritates him	…
Service: Measuring blood pressure (using a blood pressure meter and a smart phone to send the measured values to the back-end servers). In addition to blood pressure service, the reminder service is also used to remind the user to measure his blood pressure and an alert service is used to inform care-givers in case of high/low values.	**Capability Assumption**: Can use the blood pressure meter himself	**Risk 1**: Cannot use the service alone, making the use of the system counterproductive **Countermeasure 1**: System should provide a user friendly interface and with multiple modalities	**Risk 1** **Countermeasure 2**: measure the blood pressure with the help of his partner	Not Applicable	Not Applicable	…
	Procedure Assumption: Can follow the proper procedure	Not Applicable	**Risk 2**: Forgets the correct procedure, and thus the measured values are not precise and reliable **Countermeasure 3**: System should play a video whenever care-receiver activates blood pressure service, illustrating the correct procedure	Not Applicable	Not Applicable	…
	Location Assumption: The location of the phone is the location of care-receiver.	Not Applicable	**Risk 3**: Forgets phone at home while he is outside, thus system sends reminder late assuming user is at home and nearby **Countermeasure 4**: System should check the location by means of other sensors	**Risk 4**: Taking his partner's mobile phone by mistake, thus the location may not be correct **Countermeasure 4**	Not Applicable	…
	Identity Assumption: The measured values are for the intended care-receiver	Not Applicable	Not Applicable	**Risk 5**: Taking his partner's mobile phone by mistake, thus the measured values will be reported as his partner's blood pressure **Countermeasure 2**	Not Applicable	…
	Needs Assumption: Needs several reminders until the blood pressure is measured	Not Applicable	Not Applicable	Not Applicable	**Risk 6**: Turns off the audio of the reminder service when the user is at home and nearby **Countermeasure 6**: System should use customized reminders	…
	Timing Assumption: Can react to reminder messages in 5 minutes	**Risk 7**: Cannot react to reminder on- time, thus too many fake alerts will be send to care-givers **Countermeasure 7**: Care-giver should tailors his reminder service and increase the reminder interval	Not Applicable	Not Applicable	Not Applicable	…
…	…	…	…	…	…	…

4.6 Discussion

RiDeRS combines a number of known ideas into a simple and usable method. We started from the ideas of MOQARE of misuse-case driven risk-driven quality requirements specification, and later adopted the idea from RiskRep to let risk be the driving factor. Later we included assumption-based requirements specification known from goal-oriented requirements engineering. The original contribution of RiDeRS is the simplicity of the method and the classification of assumptions which have been shown to be useful in the health care domain.

We performed an initial validation of usability and applicability of the RiDeRS approach with IT experts and domain experts. The validation shows that not only IT experts, but also care-givers can identify risks and countermeasures, and also using RiDeRS, increases their knowledge about the system's specifications.

At this point, the desired functionality to be used in the field test had been implemented in the system, and we needed to do a risk assessment of using this system in the field with volunteers. We consider our field test of RiDeRS to be similar to real-world requirements engineering that would take place if the system were transferred to the market, and a consultant would configure this system to be used by a particular homecare provider. It is similar, because some real-world aspects are present (real care-receivers, real institution, real nurses, and up-front requirements engineering), but it is not identical, because other aspects are absent (no commercial pressure, only a single homecare institution, few users, and a researcher-consultant). Another similarity is that RiDeRS was applied before the end-users actually used the system, i.e., the care-givers were not aware of any risks resulting from the introduction and use of the system.

RiDeRS only considers users and their properties when doing a risk assessment. We speculate that this makes RiDeRS potentially useful for any critical system in which user's capabilities and limitations could lead to risks. For example, in a smart home system that is used to report emergency situation (by pressing an alert button), if the user does not have all the required capabilities (e.g., could not press the button), he can not use the system (report an emergency situation) and this obviously can create a risk. Examples like these suggest that it would be useful to test the usability and usefulness of RiDeRS in such cases too.

We do not consider assumptions about other systems in the environment of the system, such as the electricity system or the cooling system. We speculate that a RiDeRS table can be easily extended to include other assumptions about the system's environment, if needed, but this still remains be shown in a future case study.

We started by pointing out that accountability, availability, and safety risks are important for IT systems in general, and for IT support in health care in particular. Availability is taken care of because we actually have focussed on availability of the system and the safety of care-receivers. Accountability is taken care of because countermeasures are allocated to an actor, such as an IT system, a care-receiver, or a care-giver.

In RiDeRS, identification of a correct and complete list of assumptions is a crucial step for eliciting the system's requirements specification. We experimented with only one case and even in this case we may not have found all users' assumptions. There is a possibility that we have forgotten important assumptions and accordingly have forgotten important risks. Moreover, in other domains, users may be more difficult to identify than in the current case, they may be less easily accessible, and they may have conflicting goals as well as assumptions. This suggests that for those domains, RiDeRS will have to be supplemented with other techniques to deal with these phenomena.

As a future work, we will observe the system's behavior in the field test and seek to identify the actual risks. Then we can do reverse engineering, by writing those risks in a RiDeRS table, try to identify their associated service assumptions and user properties. We should also try to understand why we had not identified these risks. Based on these results, we can further identify where and why the RiDeRS approach requires further refinement. Other future work includes an experimental comparison of the risks identified by RiDeRS and the risks identified by other methods such as Leveson's STAMP method [96].

4.7 Related work

There are many requirements elicitation methods which use risks to elicit risk-reducing requirements (countermeasures). There is consensus among the methods that the analysis starts with what the system must achieve. This can be system *goals* or *functionalities*. Then, *risks* are identified which lead to *additional system requirements*. While these three types of concepts are shared by most methods, some methods are more detailed in one or the other areas. What is new about RiDeRS is that it takes into account the properties of the system's environment and the system's interaction with this environment more systematically than the existing methods. A system developer, designs the system making some assumptions about its environment, i.e., users' properties. We use an assumption-guarantee reasoning method, to list those violated assumptions which can cause risks. Using these concepts for discussing risks and countermeasures was a useful trigger for requirements identification in the case study presented in this thesis.

In this section, we compare the meta models of other methods with RiDeRS and present them in Tables *4-3*, *4-4*, and *4-5*. We distinguish the following three kinds of methods:

– *Vulnerability-oriented methods* - Some methods, such as MOQARE [66] from which RiDeRS was derived, consider the system environment only in terms of "vulnerabilities", also called obstacles or causes. Vulnerabilities are properties of the system, stakeholders, or other actors in the environment which can cause risks or increase their probability or extent of damage. Calling them all "vulnerabilities" means treating them jointly, as one group of concepts, although they are not. These vulnerabilities stem from different sources (technical system properties, stakeholders, and legislation) and might also be elicited from different interviewees. When treating vulnerabilities as one category, the question about vulnerabilities in the elicitation process plays the role of a very vague trigger, while in RiDeRS the sub-categories offer more guidance for identifying specific vulnerabilities. Some other methods that model vulnerabilities are: van Lamsweerde et al.'s [157] extension of the KAOS framework, Lin et al.'s [100] problem frames variant, Firesmith's templates for reusable security requirements [50, 51, 52], the method of the Object Management Group [124], the vulnerability-centric requirements engineering framework of Elahi et al. [44] (which is an i* extension of Elahi and Yu's method [43]) and RiskREP [65].

  A special role is played by vulnerabilities in the framework of Liu, Yu, and Mylopoulos's [101] based on i*. This framework models interactions between the system and its environment or different parts of the system, including stakeholders. This method can represent the RiDeRS concepts of system and stakeholder capabilities and limitations, and also regulatory limitations and other environmental factors. However, Liu et al. do not distinguish these concepts explicitly and the i* diagrams can become quite big even for a small model which complicates the discussion of the model with stakeholders with no prior knowledge of i*. Because of this, in RiDeRS, we switched to a simple tabular representation after using a graph presentation, at the former is more suitable for a discussion with stakeholders and requires no prior training.

Table 4-3: Vulnerability-oriented methods

Method	Goal	Risk	Countermeasure	Environment Property	System Property
RiDeRS	Goal	Risk	Countermeasure	User's capability, context and interaction	System's capability, context and procedure assumptions
MOQARE [66]	Business goal, quality goal	Business damage, quality deficiency, Misuse Case, threat	Countermeasure	Misuser, misuser property	Vulnerability
van Lamsweerde et al.'s [157] extension of the KAOS framework	(soft) Goal	Obstacle	Requirement	Agent, expectation, object plus attributes, domain properties	Vulnerability, object plus attributes
i*-based framework of Liu, Yu and Mylopoulos's [101]	Goal, task	Threat	Countermeasure	Attacker (subcategory of actor), Malicious intent	Dependency, vulnerability
Lin et al.'s [100] problem frames variant	Security (confidentiality, integrity, availability) of the asset	Threat	Security requirement	Attacker	Vulnerability
Firesmith's templates for reusable security requirements [50, 51, 52]	Security goal, asset	Security risk, threat, attack, harm	Policy, security requirement, quality factor, security mechanism	Attacker, (anti-)goal	Vulnerability
Method of the Object Management Group [124]	Asset (target of evaluation), QoS category, asset value	Unwanted incident sub-model, risk model, threat	Treatment	Context sub-model, Threat Agent	SWOT analysis, Vulnerability, weakness
Vulnerability-centric requirements engineering framework of Elahi et al. [44]	Asset, (soft) goal, task	Threat, attack, effect, security impact, malicious task	Countermeasure	Actor, malicious (soft) goal	Vulnerability, dependency, decomposition
Elahi and Yu's security extended i* method [43]	Asset, Design goal and objective, task, soft goal	Threat, attack	Security mechanism	Actor, attacker, malicious goal	Vulnerability, resource
RiskREP [65]	Business goal, Quality goal	Business damage, quality deficiency, misuse case, threat	Countermeasure	Misuser	Vulnerability
Common Criteria [30, 31, 32]	Asset, Target of Evaluation (TOE), evaluation assurance level (EAL)	Threat, risk	Countermeasure	Threat agent, evaluation context and audience to which the evaluation criteria are addressed	Vulnerability, cause of vulnerability
Threat Modeling extension of the NFR Framework [42]	Claim soft goal, asset	Threat, attack (pattern)	Architectural mechanisms, countermeasure	Attacker	Vulnerability

Table 4-4: Methods which allow modeling vulnerabilities

Method	Goal	Risk	Countermeasure	Environment Property	System Property
RiDeRS	Goal	Risk	Countermeasure	User's capability, context and interaction	System's capability, context and procedure assumptions
ATAM [87, 88]	Scenario	Scenario	System architecture	Stakeholder (customer, maintainer and developer), stimulus	Trade-off point, Sensitivity Point in the architecture
Misuse Case templates of Sindre [138, 137] and Opdahl [139]	-	Misuse Case	-	Trigger, related business rule, assumption, precondition, post-condition, stakeholder, misuse profile	Assumption, precondition, post-condition
Attack patterns of Moore, Ellison and Linger [113]	Goal	Attack pattern: goal, precondition, steps of attack, postcondition	-	Attacker, precondition, postcondition, attack tree consisting of attack goals and sub-goals	Precondition, postcondition
UMLsec [81, 82, 83]	Performance, value/security level	Risk, crash, critical behavior	Guarantee, safe behavior, containment, error handling	Constraint	Constraint, safe link, safe dependency
FMEA [147]	-	Failure, effect	Action	Cause	Cause
BSI Standards [19, 20, 21]	Assets: data and application	Threats with respect to privacy, integrity and availability	Data security concept	Processes of security management, including personnel, data security concept, outsourcing, home office, server rooms and buildings	Processes of security management, including personnel, data security concept, outsourcing, home office, server rooms and buildings
GSRM [74, 75]	Goal (model)	Risk factor, risk event	Treatment, agent, task	Risk factor	Risk factor

Table 4-5: User-oriented methods

Method	Goal	Risk	Countermeasure	Environment Property	System Property
RiDeRS	Goal	Risk	Countermeasure	User's capability, context and interaction	System's capability, context and procedure assumptions
Sutcliffe and Minocha's scenario templates [150]	Non-functional requirement	Expected failure, scenario, damage	Countermeasure	Agent and motivation	-
Abuse case model of McDermott and Fox [109]	-	Exploit, harm, Abuse Case	Use Case	Actor, resources, skills, objectives	-
NFR Framework of Chung et al. [25, 26, 27]	Soft goal	Contribution relation	Sub-goal, operationalization	-	-
Tropos [22]	(soft) Goal, task	-	Functional and non-functional requirement	Actor, objective	Architectural component, resource
Secure Tropos [119]	(soft) Goal, task, secure goal	-	Functional and non-functional requirement	Actor, objective, security constraint	Architectural component, resource, security constraint
EMPRESS Quality Models [154, 79]	Quality attribute	Damage	Means	Stakeholder	-
SecReq [68]	System objective, functional requirement, security objective, security target, target of evaluation	-	Protection profile, security requirement	-	-
Tropos goal-risk framework of Asnar et al. [8]	Goal, asset, business object, task	Event, risk	Treatment	Agent and agent goal	Resource

– *Methods which allow modeling vulnerabilities* - The following methods contain concepts which are equivalent to vulnerabilities, although they call them otherwise. ATAM (Architecture Trade-off Analysis Method) [87, 88] identifies Sensitivity Points in an architecture which are critical for the system's quality. The Misuse Case templates of Sindre and Opdahl [138, 137, 139] contain three fields which can contain vulnerabilities, i.e., assumptions, preconditions, and related business rules. The attack patterns of Moore, Ellison, and Linger [113] contain preconditions for the misuse cases. UMLsec models constraints [81, 82, 83]. FMEA (Failure Mode and Effect Analysis) [147] uses tabular templates to analyse the potential failures in a system or process, along with each failure's causes and effects. The threat modeling extension of the NFR Framework [42] includes both vulnerabilities and access points. The BSI Standards [19, 20, 21] model the complete processes of security management, including personnel, data security concept, outsourcing, home office, server rooms, and buildings, and guide the search for vulnerabilities. The Common Criteria [30, 31, 32] not only identify vulnerabilities, but also causes of vulnerabilities and documents the evaluation context. The goal-driven software development risk management model GSRM of Islam [74] and Islam and Houmb [75] contains obstacles.

– *User-oriented methods* - Some other methods do not consider the environment's role, but rather focus on the direct interaction between users and the system. These consequently mainly identify risks caused by the users. Such methods are: Sutcliffe and Minocha's scenario templates [150], the abuse case model of McDermott and Fox [109], the NFR Framework of Chung et al. [25, 26, 27], Tropos [22] and Secure Tropos [119], the EMPRESS Quality Models [154, 79], SecReq [68], and the goal-risk model of Asnar et al. [8].

4.8 Summary

This chapter presented the conceptual framework RiDeRS and a method for systematic elicitation of documentation of risk-driven system requirements in a tabular form. The RiDeRS approach is based on elicitation of risks that arise from a mismatch of assumptions made by the designer of the system concerning the environment in which the system will be used (especially, concerning the end-users who will interact with the system and use its services). Application of RiDeRS leads to the identification of countermeasures which can prevent or mitigate these risks. For the domain of homecare systems, we focus on availability and accountability risks arising from the interaction of users with the system. RiDeRS results in a table which presents a user's *used services* to achieve *goals*, violated system's *assumptions*, *risks* arising due to those assumptions, and *countermeasures* to eliminate or mitigate these risks. The table is not only useful to identify risks and system requirements, but also helps (especially for non-technical) users to be creative and to think of goals and assumptions.

RiDeRS was demonstrated by applying it to a homecare system in an illustrative case study. IT-based homecare systems are critical systems in which malfunctions or incorrect usage can threaten the health and/or life of elderly persons. Moreover, users of the homecare domain, such as care-givers and care-receivers, have specific capabilities and limitations. If these do not match with the assumptions under which the homecare system was designed, this can be a source of risks, and therefore a risk-based requirement engineering method, such as that proposed by RiDeRS, must be utilized. It is important to note that such a method should explicitly and systematically elicit risks and countermeasures.

We show one example scenario in the case study and then use RiDeRS to identify risks and countermeasures. We also provide an initial validation of the usability and applicability of the RiDeRS approach with IT experts and domain experts (care-givers). This validation showed that not only IT experts, but also care-givers could identify risks and countermeasures using RiDeRS.

Chapter 5

Service Tailoring Platform and Process *

"Old fashions please me best; I am not so nice. To change true rules for odd inventions."
— William Shakespeare

Service tailoring, which is the process of creating a new service to meet individual requirements may be achieved in a cost-effective and time-efficient manner *if* new services can be configured and composed from already deployed services. In this chapter, we propose an effective service tailoring architecture (presented in Section 5.1) and process (presented in Section 5.2) to personalize homecare services according to the individual care-receiver's needs.

The proposed service tailoring process requires minimal technical knowledge and skills from the care-giver. Our approach envisions a care-giver who can use her domain knowledge to tailor the services within a short time period (typically during a home visit).

---

*This chapter is based on the following papers:

- Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Flexible Homecare Application Personalization and Integration Using Pattern-Based Service Tailoring: Supporting Independent Living of Elderly with IT. In The Proceedings of The 11th IEEE International Conference on Computer and Information Technology (CIT), pages 467-474. IEEE Computer Society, 2011 ([176]).
- Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service Tailoring: Towards Personalized Homecare Services. In The Proceedings of The 4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing (ACT4SOC), pages 109-121. SciTePress, 2010 ([175]).

The outcome of a service tailoring process is called a *service plan*, which represents a composite service tailored to the specific needs of a specific elderly person as understood by the care-giver. The service plan is created based on a treatment pattern that corresponds to the homecare task for which automated support is available and needed. A service plan contains enough information to allow automated transformation to a complete implementation that can be deployed on a target execution platform. Section 5.3 describes a service plan and its ingredients. In Section 5.4, we discuss related work and finally, in Section 5.5, we conclude the chapter.

5.1 Service Tailoring Platform

The objective of service tailoring is to create a user-specific service plan which can be executed by a provisioning platform at runtime in order to satisfy the individual needs and preference of a care-receiver. This section focuses on the tailoring platform and starts by defining its components. The detailed explanation of how the outcome of the tailoring platform is executed by the provisioning platform is out of the scope of this thesis. Interested readers are referred to [172, 171].

Fig. *5-1* depicts a simple version a of homecare service tailoring and provisioning platform including the users and components. The tailoring platform is responsible for enhancing the creation and tailoring of the service plans by providing a Graphical User Interface *(GUI)* to the care-givers. Since the care-givers generally do not have deep IT knowledge, the technical details of the service tailoring process should be abstracted as much as possible. We aim to achieve this abstraction by using the concept of a service building block *(SBB)*. Each SBB represents one or possibly several concrete services which abstract the underlying technical details. Each SBB has *configuration* parameters for specifying behavior constraints.

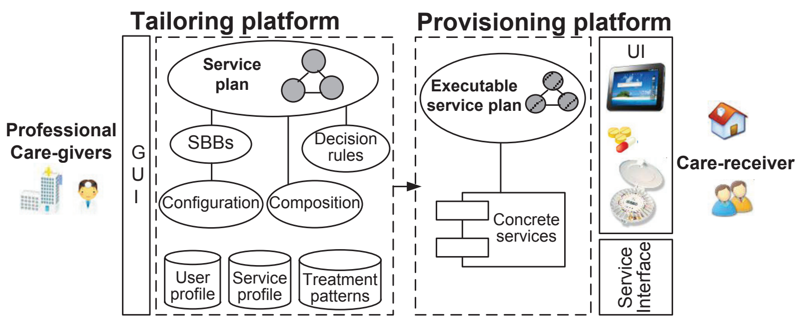

Figure 5-1 : Homecare service tailoring and provisioning platform

To simplify the creation of the service plan, we propose the use of *treatment patterns*, where a pattern can be provided by the service tailoring platform based on a selection of applicable homecare task. The care-giver selects from a menu of common homecare tasks (for example, the medication intake support task). A treatment pattern is composed of a set of SBBs. This pattern is customized for a specific generic homecare task. To personalize the selected treatment pattern for a specific care-receiver, the service tailoring platform configures the SBBs used in this treatment pattern using the predefined *user profile* of that care-receiver. The care-giver can further refine the configuration parameters until it satisfies the medical requirements of the care-receiver.

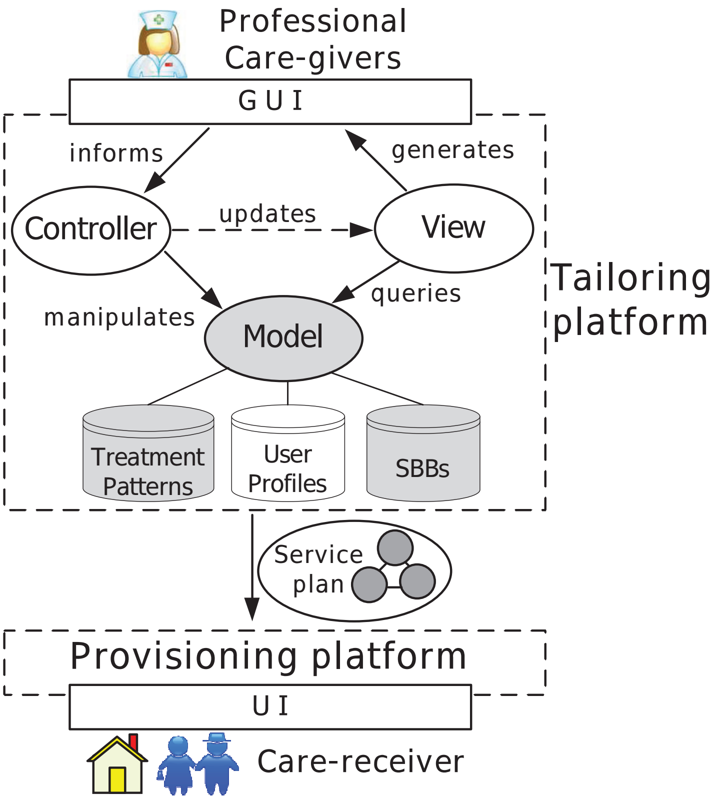

Figure 5-2 : Homecare service tailoring platform which is defined based on the Model-View-Controller pattern

Fig. *5-2* depicts a simple version of homecare service tailoring platform which is defined based on the Model-View-Controller pattern. In our platform, the *Model* manages the service plan through its various stages of refinement. The *View* presents the current service plan to the care-giver. Besides a view-presentation of the service plan, there are also additional views to assist the care-giver (e.g., a list of available homecare tasks and their corresponding treatment patterns, where the pattern is the initial version of a service plan). The *View* queries the model in order to generate the GUI. The care-giver interacts through the GUI to inform the *Controller* of her inputs and *Controller* interprets these inputs so as to update the *Model*. The *Model* also accesses the (relatively) static information that is relevant for the tailoring, namely: the set of treatment patterns, the set of service abstractions (SBBs), and the set of user profiles.

5.2 Service Tailoring Process

Our proposed service tailoring process consists of four different steps. These four steps and their constituent activities are explained below.

Step 1: Select the user for whom to create a service plan The service tailoring process starts with choosing the name of the intended care-receiver with the care-giver to specify for whom the service is being created. Moreover, it allows the tailoring platform to exploit the information about that care-receiver which is stored in the user profile and to annotate the service plan with user specific information, such as the abilities which the care-receiver needs to interact with or use the services. The user profile contains information with respect to the health conditions, disabilities, and preferences of the care-receiver (user profile will be discussed in Chapter 6).

Step 2: Select a task The tailoring process continues by the care-giver selecting a homecare task from a list of common homecare tasks. The proposed list of tasks can be limited by the tailoring platform based on the needs of selected care-receiver (the care-giver could specify the user profile, and the platform can use the user profile in order to limit the list of tasks that are appropriate for the care-receivers). Steps 1 and 2 are straightforward and intuitive, and hence easy to perform by the care-givers.

Step 3: Select a treatment pattern Based on the selected task, the tailoring platform proposes appropriate treatment pattern(s) to the care-giver. Creating a new integrated service from existing SBBs by a nontechnical user is a difficult task, since the user needs to know which existing individual services do what. By employing patterns, the care-giver does not need to select the desired SBBs herself. Instead, once she selects a task, the tailoring platform proposes an associated pattern with that task which already include required SBBs to satisfy the requirements of that task. In this manner, the care-givers express what they want (i.e., what SBBs they need) at a higher abstraction level.

For each task, a couple of alternative treatment patterns can be proposed. The purpose of having several treatment patterns for each task is to have different orchestration schemes (composition of services) for the same task. This is helpful, because the care-giver can choose a pattern that is appropriate for a specific care-receiver from the existing patterns and there is no need for re-composition of patterns by the care-giver. For example, for the medication intake support task described in Section 3.3, we can have two patterns. In one of them the medicine dispenser is enabled *before* sending the reminder while in the other pattern a reminder is sent *before* enabling the dispenser. As another example, for the weight monitoring task, in the case of too great a weight difference, one pattern would immediately inform the care-givers (because of the care-receiver's health condition and the measured weight value) and the other pattern could recommend the care-receiver to take a medicine without sending alert to the care-givers.

Step 4: Configure the selected treatment pattern After selecting one of the suggested patterns, the care-giver can easily configure the pattern based on her knowledge and consideration of the care-receiver's situation. The pattern selected in the previous step may be presented to the user based on default configuration values derived from the target care-receiver's user profile. However, the care-giver can also refine the pattern by specifying appropriate values to the configuration parameters of each SBB.

The service tailoring process ends with the generation of a care-receiver's specific homecare service specification, i.e., the service plan which constrains the behavior of the homecare services at runtime according to the needs and preferences of the care-receiver. The service plan should foresee all the requirements and preferences of a care-receiver and determine the corresponding desired behavior of the services. The service plan thus created, if confirmed by the care-giver, is deployed to the provisioning platform to be executed.

Since a care-receivers' situation could change, a created service plan may need to be refined at a later time. To re-tailor the previously created service plans, is similar to the initial service tailoring process, thus a care-giver performs steps 1 and 2 to choose a service plan, and performs steps 3 and 4 to re-tailor it. The only difference from the first time tailoring is that all possible configuration parameters have been filled with earlier plan's the values and the care-givers may need to change some of them to satisfy new requirements of the care-receivers.

5.3 Service Plan

We call the treatment pattern configured to satisfy the requirements of a specific care-receiver a service plan. A service plan consists of two parts: internal and external representations. The external representation of a service plan is to be understood and configured by the care-givers. The internal representation of the service plan is to be understood and executed by the provisioning platform. The internal representation of the plan should be enough detailed that it can be executed by the provisioning platform at runtime. As such, a SBB and its configuration parameters might be mapped to several concrete services and their input parameters at the provisioning platform.

We specify the external representation of the service plan by using workflow-based techniques. In the proposed approach, each service plan consists of several activities referring to the use of one or a composition of several *SBBs* and their configuration parameters (annotated as a data item in each activity). Beside activities, a service plan contains some *decision rule* points to specify the behaviour of the service plan at runtime. A care-giver can easily configure plans by specifying/modifying the configuration parameters' values. By composing and configuring the SBBs in a service plan, we will be able to provide the required homecare services to the care-receivers. In the following sub-sections, we describe SBBs, different types of decision rules and present two examples of service plans for two care-receivers with different requirements.

5.3.1 Service Building Blocks

The homecare support actions are represented as user-level service descriptions, and referred to as *Service building blocks* (SBBs). SBBs are defined when these actions have independently useful functionalities. Each SBB corresponds to functionality that has been implemented by a device and/or software application, and is available for use by the care-receiver. We have conducted a technology survey, constrained in scope by the usability and applicability of the technology for the identified homecare tasks and their patterns to define some of the required that might be useful SBBs for the homecare domain.

As shown in Fig. *5-3*, a treatment pattern consists of references to one or a composition of several SBBs. By composing and configuring these SBBs in a treatment pattern, we will be able to provide the required homecare services to the care-receivers. Since a SBB is an abstraction, several alternative implementations may exist that correspond to the same SBB. The tailoring platform is not aware of these alternatives, but configuration parameters of the SBB allow care-givers to indicate functionality which may only be provided by one specific SBB implementation. For instance, a SBB for a reminder service can have a configuration parameter which enables the care-givers to determine how much time in advance the provisioning platform should send a reminder to the care-receiver. To execute this, the provisioning platform employs a concrete service namely *waiting service* which takes the time it should suspend the process as input and resumes the process after this time has elapsed.

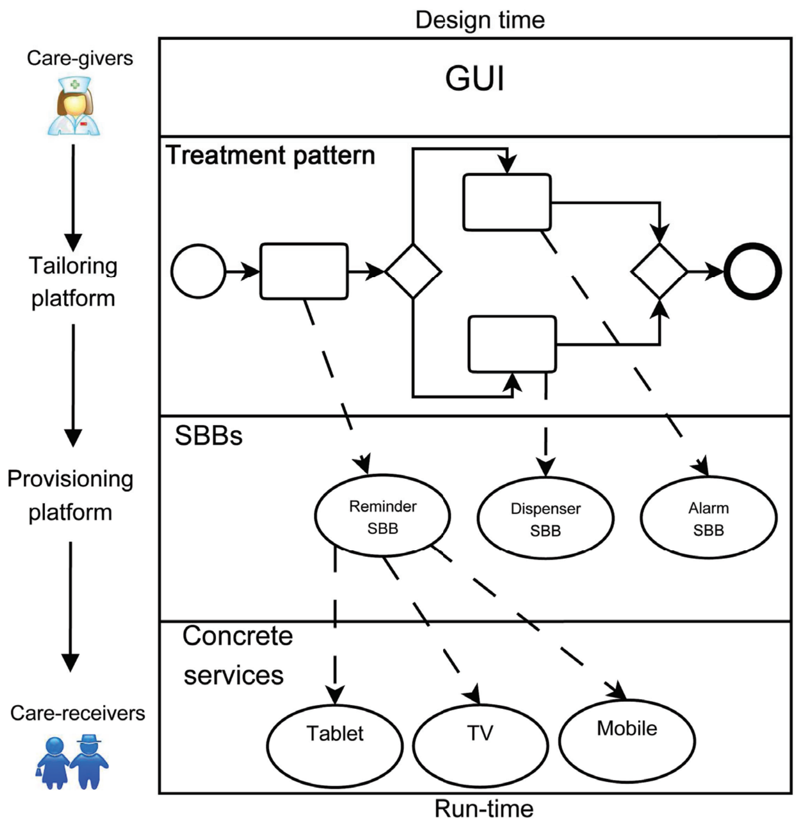

Figure 5-3 : Service Building Blocks

Configuration parameters of SBBs allow a care-giver to specify different aspects of the SBB such as service operations and user interface modalities. Below we present some of these SBBs and their configuration parameters in plain English. We use plain English solely for the purpose of this explanation.

1. Reminder: To notify a care-receiver to do something.
   Operation: Send Message()
   Configuration parameters: *string* Message: message to send to the care-receivers, *time* Timeout 1: how long before the scheduled event to send the first reminder, *time* Timeout 2: waiting time between each reminder repetition, *integer* Repetition: Number of reminder message repetitions, *list* Modality: via which device/modality to deliver the message.
   Some possible values for configuration parameters: Modality: audio, video, text, and vibration.

2. Alert: To inform a care-giver if there is a hazardous situation.
   Operation: Send Alert()
   Configuration parameters: *string* Message 1: message to send to the care-givers about ignored reminder messages, *string* Message 2: message to send to the care-givers about high/low vital sign values, *list* Interface: via which interface to show the alert, *person* care-giver(s): to whom send the alert.
   Some possible values for configuration parameters: Interface: phone call, SMS, Google talk, and email.

3. Medicine dispenser: To help a care-receiver to take the correct dosage of medicine(s).
   Operation: Enable(); to enable (disable) dispenser, so that care-receivers can (not) take their medicine.
   Configuration parameters: *list* Modality: the type of interaction between care-receiver and dispenser to take the medicine from it.
   Some possible values for configuration parameters: Modality: pushing a button, automatic.

4. Oxygen saturation: To measure the level of oxygen saturation of the body of a care-receiver.
   Operations: Set Saturation Level Threshold (); to set a threshold for sending a notification to a care-giver.
   Get Saturation Level (); to measure the current oxygen saturation level.
   Configuration parameters: *integer* Threshold 1: for measured values under this value, an alert message should be sent to the care-givers, *integer* Threshold 2: for measured values above this value, an alert message should be sent to the care-givers, *time* Validity: maximum validity of last measured oxygen saturation level.
   Some possible values for configuration parameters: Threshold 1: 92 and Threshold 2: 100.

5. Blood pressure: To measure the blood pressure of a care-receiver.
   Operations: Get Blood Pressure (); to measure the current blood pressure level.
   Configuration parameters: *integer* Diastolic 1, Diastolic 2: to set the threshold for diastolic level, *integer* Systolic 1, Systolic 2: to set the threshold for the systolic level, *time* Validity: maximum validity of last measured blood pressure.
   Some possible values for configuration parameters: Diastolic 1: 55, Diastolic 2: 100, Systolic 1: 80, Systolic 2: 200.

6. Weight: To measure the weight of a care-receiver.
   Operations: Get Weight amount (); to measure current weight.
   Configuration parameters: *integer* Compare point: compare the measured weight with this value, *integer* Weight max/min: if the the difference of the measured weight with the compare point is higher/lower than this value, an alert message should be sent to the care-givers, *time* Validity: maximum validity of last measured weight value.

7. Step counter: To measure the steps a care-receiver takes during a day.
   Operations: Set Steps Amount(); current day's step count.
   Configuration parameters: *integer* Steps: to set the threshold for steps which a care-receiver should take during a day.

8. Agenda: To set notification schedules of a care-receiver for various tasks (e.g., blood pressure measurement, medicine intake, appointments).
   Operations: Set Event(); to set an event in the care-receiver's agenda.
   Notify Event(); to notify the care-receivers or/and provisioning platform (to trigger a process) based on the scheduled time.
   Configuration parameters: *date* From: from which date to schedule, *date* To: until which date to schedule, *time* Time: for which time to schedule, *list* Repeat: how often this schedule will be occurred (repeated), *list* Location: where the event location is (this parameter is needed for some tasks, e.g., to notify a care-receiver about a social activity in a specific place).
   Some possible values for configuration parameters: Repeat: every day, once a week, once a month, or twice a day.

Some SBBs can provide additional operations which may not be relevant to the care-giver in tailoring the services, for example, a *reminder* SBB can provide a *Confirm Acknowledge()* operation through which a care-receiver can confirm that the reminder message was received or a *medicine dispenser* SBB can provide a *Medicine Taken()* operation through which a dispenser device can confirm that the medicine was taken from the dispenser. These operations are not discussed in this thesis, since they are not visible to the care-givers and do not play a role in service tailoring.

The configuration parameters can receive the configuration values from two different sources: care-givers and a user profile. For example, in the *reminder* SBB, the *modality* parameter can acquire the value from the user profile; whereas *message, timeout and repetition* parameter values can be set by the care-givers.

In order to make the SBBs reusable, different and sometimes conflicting aspects need to be balanced. One of these aspects is the proper level of SBB granularity [63]. Defining more or fewer operations and configuration parameters makes a SBB more or less generic. Having more generic SBBs make them more reusable, however it makes composition of these SBBs more complicated.

5.3.2 Decision Rules

Decision rules are used to specify how the system should behave when providing services. We use rules to support care-givers as they can specify constraints or values while defining the service plan. We identify four types of decision rules in the homecare domain. These rules are:

1. *Trigger Rules*: This type of rule is needed to specify when a process (i.e., service plan) should be started. We model this type of rule as an Event Condition Action (ECA) rule of the from *on* event *if* predefined condition is true *do* start the process.
   We identify three type of events which can trigger a service plan process:

   – *Time event* based on predefined schedule

   – *Context event* based on the change in the context of a care-receiver

   – *Chain event* based on being called from another processes (the condition part of this event is always true)

Three examples of trigger rules, used in two different service plans, are illustrated in Table *5-1*.

Table 5-1 : Examples of trigger rules

Rules	**Event**	**Condition**	**Action**	**Homecare Task**
R1	Agenda.Medicine	Medicine notTaken	After t min start the process	Medication intake support
R2	CareReceiver. NotEnoughWalked	Care-receiver Activity.Sitting	start the process	Monitoring Doing Training
R3	BloodPressure Process.Called	Always true	start the process	Monitoring Doing Training

2. *Configuration Rules*: In each service plan process, decision points are used to specify the control flow of the process to decide what to do based on the current run-time data. The necessary values comes from the configuration parameters of SBBs. The following rule is an example of the use of configuration rules, in the *medication intake support* task.

If Reminder.repetition $> n$

Then Alarm.send

3. *Mapping Rules*: This type of rule is used for mapping each SBB to available concrete services. The following rule is an example of mapping rules, in the *medication intake support* task for a care-receiver with visual impairments.

If Care-receiver.hasVisualImpairments

Then Reminder.Modality= Voice

4. *Safety Rules*: These types of rules are used to specify safety constraints and requirements because the homecare services are

classified as safety-critical systems [94]. A system malfunction could lead to the loss of a life. The following rules are examples of safety rules, in the *medication intake support* and *monitoring training activity* tasks.

If MedicationTaking.Process.Fails *Then* Call care.giver

If TrainingActivity.Process.Fails *Then* SendSMS to care-giver

5.3.3 Example Service Plans

We illustrate our discussion of service plans with two homecare tasks and their corresponding treatment patterns. The service plans were created by *Nancy* as a care-giver for *John* and *Mary* as was explained in the application scenario in Section 1.2.

Fig. *5-4* shows the treatment pattern for *blood pressure* monitoring task which is selected by *Nancy* to be configured for *John*. Some of the configuration parameters, shown in this example, have default values such as *visual modality* for the reminder message, which is obtained from the care-receiver's profile. Nancy could refine the pattern further by providing values for the configuration parameters based on the needs of *John*. While Fig. *5-5* shows the treatment pattern for *monitoring training activity* task that is selected by *Nancy* to be configured for *Mary*.

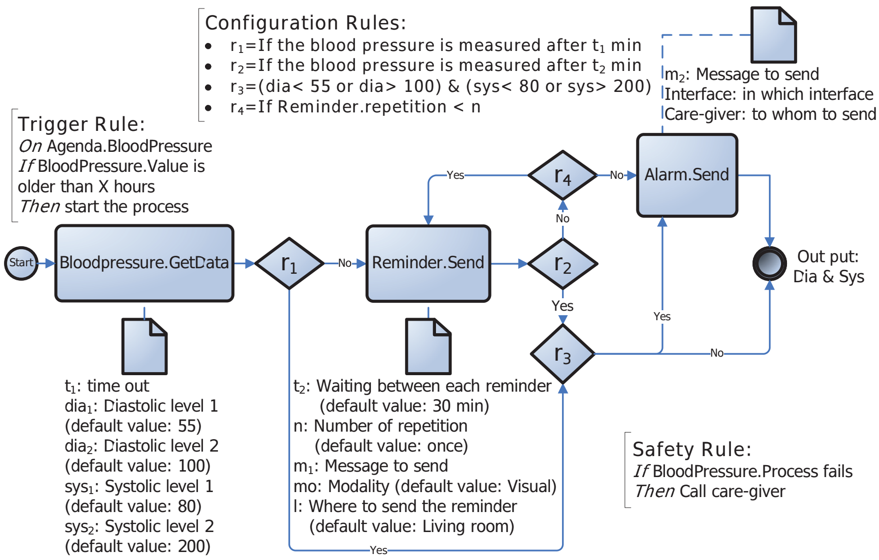

Figure 5-4 : *Blood pressure* monitoring treatment pattern selected for John

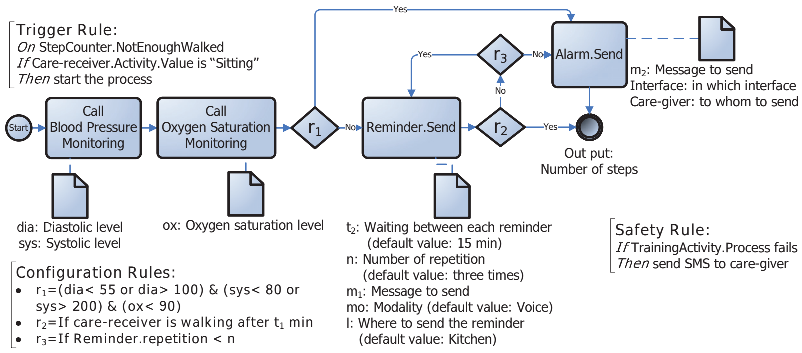

Figure 5-5 : *Monitoring training activity* treatment pattern selected for Mary

5.4 Related Work

There are several approaches advocating the use of a service plan with the goal of realizing application services with ease [23, 93, 108, 159, 6]. These approaches are defined based on the concept of graphs, workflows, rules, or the combination of these concepts. The scope of these approaches are also different, i.e., they are defined for resolving specific problems in different application domains. In the following, we discuss some of the relevant approaches to identify their strengths and weaknesses in comparison to the work presented in this thesis.

In other domains, enriching business processes by business rules to achieve better flexibility has been investigated [23, 93]. The main idea behind these approaches is to extract the highly dynamic decision rules from the process and manipulate them without affecting the processes themselves. This extraction seems interesting because with respect to current service provisioning technologies changing the business rules are simpler than changing business processes [18]. However, the existing approaches cannot be used as is because in the homecare domain the decision rules and configuration parameters should be configured by a non-technical care-giver. Moreover, the homecare-specific rules such as *safety rules* may not be fully supported by the existing approaches.

In other works, several approaches have been proposed to support computer-based modeling of medical protocols [35]. Rule-based medical protocols such as Arden Syntax for Medical Logic Modules (MLM), which is part of Health Level Seven (HL7) standard [15], has been employed to facilitate knowledge sharing among care-givers for instance for COPD treatment [144]. Workflow-based medical protocols (e.g., care-flow [130, 58]) have been proposed to define which task needs to be executed and in which order while providing care to a care-receiver. Unlike in our approach, these works neither consider the use of workflow patterns nor personalization. The use of workflow patterns allows non-technical care-givers to create and personalize the service plan for a specific care-receiver.

The service plan as proposed in the Match project [107] is based on a directed graph. This plan specifies how the interaction between different services can be mapped, and hence allows the user to specify multiple alternative compositions that can be used to satisfy a specific need. These alternative compositions are prioritized in the order of the care-receiver's preferences. However, creating such a service plan is beyond the capacity of non-technical users, such as care-givers. The work presented in [120] proposes using UML as a means to define a service plan. This approach is similar to ours, however they do not explicitly support decision rules in addition to their UML plans, thus, tailoring an existing service plan for an individual's needs and addressing the preferences of a care-receiver with their approach, is quite a time consuming task.

5.5 Summary

In this chapter, we discussed an approach through which a professional care-giver can create personalized plans for automated care support for elderly persons who want to live independently in their own private home. The personalized plan, called a service plan, is created through a GUI provided by a service tailoring platform. The creation of the service plan is facilitated by using treatment patterns, corresponding to common homecare tasks, as a starting point for the service plan. Since the needs of individual elderly persons slowly change with time, care-givers should also be able to re-tailor previously created service plans quickly and easily.

We defined service tailoring as a process which consists of activities that a care-giver needs to perform to personalize the services at design time, i.e., prior to provisioning homecare services. The service plan constrains the behaviour of the homecare services at runtime. The service plan is a composition of existing service building blocks (SBBs) for a specific common homecare task which are configured by the care-giver to provide appropriate homecare applications to the individual care-receiver. Creating such a service plan could be difficult and time consuming for a care-giver, hence the use of treatment patterns simplify the creation of a service plan. A care-giver does not have to create a service plan from scratch, but rather can configure and modify an existing pattern proposed by the tailoring platform. Because of the high similarity of the care services at an abstract level, we interviewed several care-givers to identify common homecare tasks and their corresponding treatment patterns. In this chapter, we further investigated the ingredients of each treatment pattern and described the required SBBs, their configuration parameters, and decision rules for the homecare domain. We could make a richer collection of homecare services by composing and configuring these SBBs.

The service tailoring concept can be applied at different levels in a homecare system. For example, treatment patterns can be tailored by adding or removing activities or decision rules; SBBs can be tailored by adding or removing operations or changing configuration parameters, or even by adding new SBBs. Different roles should have different level of access rights in the service tailoring process. As discussed earlier, professional nurses follow the medical protocols in providing specific care service and therefore they can not alter the treatment patterns. However, a physician together with the manager of a department of a care institute perhaps sit together once a year and decide to change some of their local medical protocols and as a result alter the predefined treatment patterns (this will affect the service plans and the care-giver should re-tailor them). So there is a need to provide a management interface which allows tailoring of the treatment patterns.

Chapter 6

User Profile *

"If you want truly to understand something, try to change it."
— Kurt Lewin

We introduced a service tailoring process in the previous chapter, which exploits the Service Oriented Architecture (SOA) paradigm [127] to create personalized composite services through configuration and composition of predefined service building blocks (SBB). The concept of a SBB is used to denote the smallest manageable unit of service functionality from the point of view of care-givers. Each SBB represents a concrete service or alternative concrete services, abstract the technical details that are irrelevant to the service tailoring (i.e., to the care-giver).

Unlike other domains, users of homecare services are subject to various impairments. This characteristic of elderly people is a strong determinant in service usage. Hence, services should be selected based on not only the user's preferences, but also based on their health situation. In this chapter, we describe an approach that employs predefined information about care-receivers, called a user profile, to hide most of the technical details from the care-givers who do the service tailoring.

In Section 6.1, we define the information to be included in a user profile and subsequently used by patterns. In Section 6.2, we define how the service tailoring process exploits information stored in the predefined user profiles, which subsequently guides the provisioning platform to select the proper instances of a service for various specific care-receivers. After that, in Section 6.3, we illustrate the approach with an example. In Section 6.4, we discuss related work and compare them with our approach. Finally, in Section 6.5, we conclude the chapter.

---

*This chapter is based on the following paper:

– Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service Tailoring: Towards Personalized Homecare Services. Procedia CS, 5:409-417, 2011 ([177]).

6.1 User profile and Treatment Patterns

Each SBB is the realization of one or possibly several concrete services (CS). SBBs abstract the technical details of CSs and provide a generic service specification and an interface to be used by care-givers in the creation of a service plan. Each SBB has configuration parameters for specifying behavior constraints. Among these configuration parameters, in this chapter, we focus on usage requirements and context-aware parameters. Usage requirements stipulate the abilities which the care-receiver needs to interact with or use the service; while, context-aware parameters are those parameters which express the desires of the care-receiver for different services in different situations. For instance, a reminder SBB can realize several concrete reminder services in a real world care environment to remind the care-receiver differently, for example, via showing a text message, playing a sound, or vibrating a device used by the care-receiver. At runtime, each SBB in the service plan will be instantiated as one of its possible alternative CSs based on the abilities and desires of the care-receiver that uses the service. We exclude here a discussion of the provisioning platform which maps the SBBs in the service plan to the CSs and generates executable code based on the annotated information of the service plan.

To simplify the creation of a service plan, we propose the use of treatment patterns, where a pattern can be provided by the service tailoring platform based on selecting a homecare task from a menu of common homecare tasks. To personalize the selected treatment pattern for a specific care-receiver, the service tailoring platform configures the SBBs used in the treatment pattern using a predefined user profile of the specific care-receiver. The care-giver can refine further the service plan until it satisfies the medical requirements of the care-receiver.

A user profile is typically filled in by a care-giver before the start of the service tailoring process. The information in a user profile helps to partially automate the generation of a service plan by suggesting and/or constraining certain choices. The user profile of a care-receiver may suggest usability options of selected SBBs (e.g., prescribing that the user interface should be voice-based, since the user profile indicates that the care-receiver has a visual impairment) and suggest decision rules with respect to alternative behavior (e.g., prescribing that voice messages should be played immediately unless company is present, since the user profile indicates that the care-receiver has a corresponding privacy preference). We discuss our motivation for the user's profile structure and show how a user profile can be used in the service tailoring process. We also discuss the impact of this use on other information structures such as the service plan and the description of SBBs.

This thesis does not specify the creation of a user profile instance nor how the information about the care-receiver is collected, rather it simply specifies the structure of a user profile and how this information can be exploited when building a service plan and its influence on the configuration of this service plan. In the remaining of this section, we explain user profiles and their role in the tailoring process.

We will utilize the application scenario presented in Section 1.2 on page 4 to illustrate the use of proposed user profile and motivate the need for it.

6.1.1 User Profile

We define a *user profile* as "a structured representation of a user's personal data, needs, and preferences that is used by the tailoring platform to configure SBBs of a service plan".

We identify three necessary items which should be contained in a user profile.

1. **Personal Information:** Represents the personal information of the care-receiver such as name, year of birth, weight (in kilograms), gender, and language. In some scenarios, this information can affect the configuration of SBBs, for example if $John$ speaks Dutch language, then the output message of the reminder service should be in Dutch. We identify five parameters for personal information, but this can be extended if other parameters are deemed necessary. The schema of the personal information is the following:

   $U_I = (Name : String, Year : Int, Weight : Int, Gender : String, Language : String)$

   For example, for $John$:
   $u_I = ("John", 1929, 71, "Male", "Dutch")$

2. **Abilities:** Represents the information about the care-receiver's physical and mental (dis)abilities. For example, if the care-receiver cannot hear well, he will not be able to use any services or devices which utilize sound as a means to convey messages to him. This information can be generalized based on the international classification of functioning, disability, and health (ICF) [170] to: hearing, seeing, touching, mobility, speaking, and memorizing. However, this can be extended if other parameters are deemed necessary. Abilities are stored as Boolean variables where 0 means absent or reduced ability:

$U_A = (Hearing, Seeing, Touching, Mobility, Speaking, Sensing, Memorizing)$

For example, for $John$:

$u_A = (0, 1, 1, 1, 0, 1, 0)$ means that he cannot hear but he can see and so on.

Instead of using either 0 or 1 as the value of an ability, we could use a more general form of these abilities. For example, one can completely be blind "0" , or he might have perfect vision "1" or something in between. In that case, we could assign a value in the range [0,1] to the ability parameters and adapt the service provisioning with regards to the value of the abilities (e.g., the lower the value of seeing ability, the larger the font size or the lower the value of hearing ability parameter, the louder the audio output). Without loss of generality, in this thesis, we use only the values 0 or 1, but these can be extended to range values.

3. **Preferences:** Indicates how the care-receiver wants the desired services to be delivered to him in different situations. For example, the care-receiver may want to be reminded through vibration of the cell-phone, instead of receiving a voice message when he has company.

   Preferences usually are defined as a strict partial order, such as *"I like A better than B"* [89]. Based on Dey's definition of context ("Context is any information that can be used to characterize the situation of an entity") [37], we add context information (such as location and time) to define preferences, i.e., *"I like A better than B in situation C"*. So, we define *user preferences* as "rules and conditions set by the user to characterize his/her desire in each context situation".

   A user preference consists of two parts:

   – The first part is the condition part in which the situation is specified.

   – The second part is the desired action for the situation specified in the first part.

The conditional part includes three parameters namely: *Location, Time, People around.* However, these parameters can be extended if other context information are deemed necessary. The possible values of the parameters are determined in consultancy with the care-givers. The second part relates to user abilities. Here are some examples of user preferences:

If (-, -, People around > 0) *Then* u_P =(x, 0, x, x, x, x, x)

Where ”0”, ”1” and ”x” denote ”not desire”, ”desire” and ”do not care” respectively. A dash (“-”) means there is no condition for the relevant context information. This preference means that when there is someone around the care-receiver, he prefers non visual messages.

If (Location= outside the house, -, -) *Then* u_P =(0, x, x, x, x, x, x)

This means that when the care-receiver is outside the house, he prefers non audio messages.

If (-, -, -) *Then* u_P =(x, 1, x, x, x, x, x)

This means that the care-receiver always prefers visual messages. If there is more than one preference, the care-receiver could specify their priorities maybe with the help of care-givers (the order in which the rules are specified determines their priority).

Fig. *6-1* summarizes our discussion about the user profile.

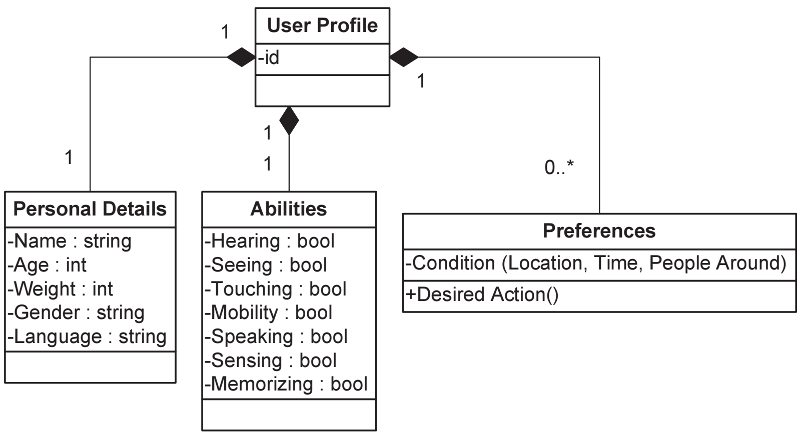

Figure 6-1 : User profile

6.1.2 An Example of Treatment Patterns

Treatment patterns were discussed in Section 3.3. One such a pattern is shown in Fig. *6-2*. This treatment pattern is represented by a BPMN like notation in which an action is represented by a rounded corner rectangle which at run-time is an SBB realized as a CS. The decision points are represented by diamonds. The treatment pattern shown in Fig. *6-2* is for reminding people to take their medicine(s). It consists of three actions: $dispenser$, $reminder$, and $alarm$. This service plan assists a care-receiver to take his medicine on time. If it is time to take medicine, the $dispenser$ is enabled and a $reminder$ message is sent to the care-receiver. If he ignores the reminder and does not take his medicine after t_2 minutes, the $reminder$ service sends another reminder message up to n times. If the care-receiver still has not taken his medicine, then an $alarm$ message is sent to the care-giver indicating that the care-receiver did not take his medicine.

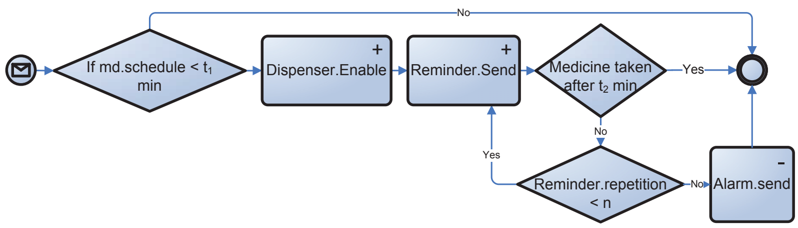

Figure 6-2 : Treatment pattern for $medication\ intake\ support$ task

The target users of the SBBs are either care-receivers (specified by a "+" sign in upper right corner of an action) or care-givers (specified by a "-" sign in upper right corner of an action). The target user of each SBB is indicated in its specification. Later, in the tailoring process, the user profile information is added only to those SBBs targeting care-receivers.

6.2 Creating a User-specific Service Plan

As already mentioned, we assume that there are already predefined treatment patterns for each care task in the tailoring platform. When a care-giver wants to make a specific service plan for a care-receiver she must select the appropriate homecare task. Then, the tailoring platform adds the necessary information excerpted from the user profile to the treatment pattern. Hence, a detailed service plan is a pattern annotated with the information of a specific care-receiver.

The care-giver may adapt the service plan generated automatically by the tailoring platform to suit the situation of the care-receiver. The annotated information of the service plan can be of two types: hard constraints and soft constraints.

6.2.1 Hard Constraints

The ability information in the user profile provides hard constraints which are used to specify the service requirements (S_R) those requirements which $must$ be met by the CSs. For example considering $John$'s profile, u_A = (0, 1, 1, 1, 0, 1, 0), thus all the SBBs in the service plan, with this care-receiver as the target user, will be annotated with the signature s_R = (0, x, x, x, 0, x, 0). This signature means that all those CSs of that SBB that do not need the first, fifth, and the last abilities can be selected in the executable service plan. However, those services may or may not need the other abilities.

6.2.2 Soft Constraints

The second type of information is added to service plans as soft constraints based on user preferences. These constrains (S'_R) are those requirements which $should$ be met by the CSs. For example, if we have the following user preference: If (-, -, People around $>$ 0) Then u_P =(x, 0, x, x, x, x, x). We define an alternative signature: s'_R =(x, 0, x, x, x, x, x). After applying the hard constraints if there is more than one CS which satisfies all the requirements of the care-giver, then we apply the soft constraints one by one in the order already defined, until either there is only one CS left or there is no additional user preference. Applying the hard constraints before the soft constraints prevents selection of inappropriate services if preferences are incompatible with the abilities of a care-receiver.

6.3 Example

We use the scenario described in Section 1.2 to illustrate our approach. Based on $John$'s profile, a user specific service plan will be generated by the tailoring platform as shown in Fig. *6-3*. As shown in the figure, the $reminder$ and the $dispenser$ are used by the care-receiver, while the $alarm$ SBB is used by the care-giver. Therefore, in the configured service plan only the $reminder$ and $dispenser$ SBBs have annotated information. Considering the abilities and preferences parts of the user profile, we assume that $John$'s user profile has the following values:

$$\begin{cases} u_A = (0, 1, 1, 1, 0, 1, 0) \\ If(-, -, \text{People around} > 0)\ Then\ \mathrm{u}_P = (x, 0, x, x, x, x, x) \end{cases}$$

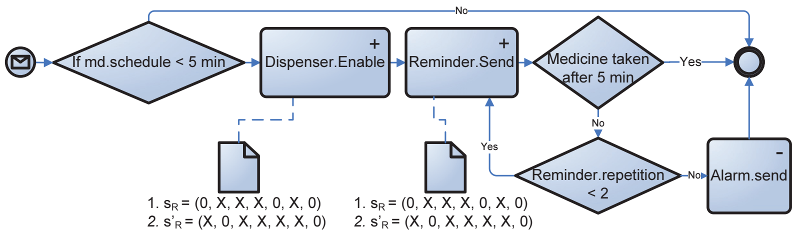

Figure 6-3 : Service Plan for John, Automatically Generated Using Corresponding Pattern and John's Profile

As mentioned in Section 6.1.1, we consider the care-receiver's abilities (hearing, seeing, touching, mobility, speaking, sensing, memorizing). The above values then indicate that *John* has impairments with respect to hearing, speaking, and memorizing, and that in the case of company *John* prefers not to receive visual messages.

The annotated information, represented as a data items, are attached to those SBBs used by *John* in order to guarantee their usability. For example, because *John* has a hearing problem the reminder SBB should suit this disability, i.e., it should **not** use voice to convey its message. To ensure this, the appropriate signature (0, x, x, x, 0, x, 0) is connected to the reminder action. *Nancy* can further configure the created service plan for *John*, for example, by assigning values to t_1, t_2, and n as 5 minutes, 5 minutes, and 2 times respectively.

As shown in Fig. *6-4*, the service plan for Mary based on her profile is different since:

$$\begin{cases} u_A = (1, 0, 0, 0, 1, 1, 1) \\ If(\text{Location= Outside the house, -, -})\ Then\ \mathrm{u}_P = (0, x, x, x, x, x, x) \end{cases}$$

As shown in Fig. *6-5*, *Nancy* can further configure the service plan for *Mary* by removing the *reminder* SBB and increase the waiting time after enabling the *dispenser* from 5 to 15 minutes, as *Mary* does not forget, but can not move quickly. *Mary* needs the assistance of dispenser to take the correct medicine as she has limited vision.

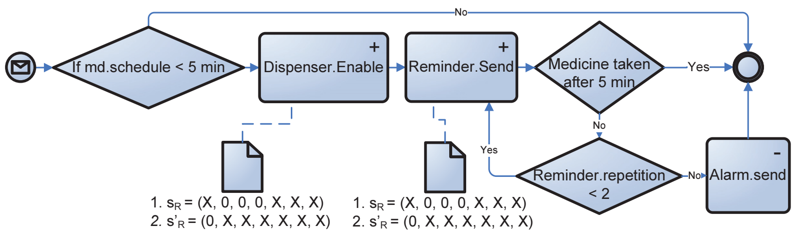

Figure 6-4 : Service plan for Mary, automatically generated using corresponding pattern and Mary's profile

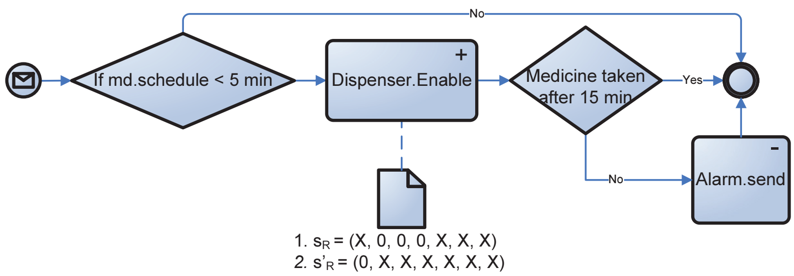

Figure 6-5 : Service plan for Mary, as configured by Nancy

In this way, the service tailoring platform generates the structure (i.e., the added data items), and any values that can be automatically derived from the user profile are filled in. The remaining values have to be provided by the care-giver via interaction with the service tailoring platform GUI.

6.4 Related Work

To decrease the effort required for personalization and to avoid manual development of business process models from scratch, the concept of *reference models* (patterns) and their configuration to satisfy the needs and preferences of target users seems to be a promising approach [61]. To adapt a reference model to individual requirements, either additional information can be added to fine tune the reference model, or some of the existing elements can be eliminated from the reference model to abstract the reference model [61, 132, 14].

In our case, reference models are treatment patterns, which should be understood, altered, and confirmed by non-technical developers, such as care-givers. Therefore, we use abstract patterns which are automatically annotated based on the target care-receiver's user profile and further refined by care-givers who adds extra information (see Fig. *6-3* and Fig. *6-5*) and possibly removes information (see Fig. *6-5*). Our service tailoring platform supports care-givers in adding/removing information.

Alternatively, flexibility and personalization of the process models may need to be supported at runtime. Exception handling approaches can be considered as one of the possibilities to achieve this flexibility [84]. However, this approach is unsuitable in the homecare domain as care-givers have to develop the service plans. We cannot expect care-givers to be well acquainted with exception handler these technologies and to have enough technical knowledge to create exception handlers the service execution at runtime. In our approach, we abstract these technical details and allow care-givers to specify the personalization parameters at design-time.

User profiles have been used in the healthcare domain to provide standards for sharing, managing, and retrieving of the care-receiver's physiological raw data (inter-operable electronic health records) among care-givers [141, 24, 90, 62]. Unlike our approach, these profiles are not used specifically for personalization of services. In other domains, user profiles and service catalogues have been used for personalized selection of services [12, 142, 149, 146], but such techniques are not applicable in the homecare domain as they do not take into account the physical and mental constraints of users. User abilities are considered and defined in the ETSI user profile standard [48]. However, they mainly provide some settings for the deployment of ICT services, as opposed to specifically selecting services based on user profile information as proposed in our approach.

In other work, several approaches have been proposed to support computer-based modeling of clinical guidelines and execution of these models [35]. Rule-based clinical guidelines such as the Arden Syntax for Medical Logic Modules (MLM), which is part of the Health Level Seven (HL7) standard [15], have been employed to facilitate knowledge sharing among care-givers (e.g., sharing knowledge for COPD [144] treatment). Workflow-based clinical guidelines (e.g., care-flow [130, 58]) are proposed to define which task needs to be executed in which order when providing care to a care-receiver. Unlike in our approach, these works neither consider the use of workflow patterns nor the personalization aspects. However, the use of workflow patterns would allow non-technical care-givers to create and personalize the service plan for a specific care-receiver. In [73], the implementation of an agent-based platform to execute clinical guidelines in an efficient way is proposed. In this work, the concept of a user profile to provide patient-centered services is used. The user profile, however, is mainly used for specifying users' preference parameters (e.g., on which day or at what time care-receivers prefer to go for clinical treatment). In our approach, we additionally consider care-receivers abilities when using services, which is important for selecting appropriate services.

6.5 Summary

In this chapter, we discussed a situation in which an elderly person receives care at home. Unlike other domains, users of homecare services are subject to various impairments. This characteristic of elderly people is determinant in service usage. Hence, there should be a selection process over the alternative services, not only based on user preferences, but also based on their health related situation.

To provide such services, we can use different approaches. For example, designing specific services for each particular user is an option. However, such a solution is undesirable because of the cost and time required to design services for each and every elderly person. Because of the high similarity of the care services at an abstract level, we proposed an approach to enable creation of a personalized homecare service based on SBBs which can be composed and configured easily by a care-giver. We can make a rich collection of homecare services using SBBs. One could argue that configuring and composing already deployed SBBs to create a service through tailoring is less expressive than developing the service from scratch. Indeed, some users may have very special needs or preferences which cannot be supported by adopting the properties of the SBBs nor selecting their composition. However, we still think that considering the cost, a service tailoring approach such as ours is the best compromise.

Even though homecare services are similar at the abstract level, the usage requirements of their concrete services are different. To select appropriate services which are consistent with the care-receiver's abilities and preferences, we used an approach which adds information derived from the user profile to the service plan. We think that an appropriate user profile in the homecare domain should contain the care-receiver's physical and mental abilities, and consequently we consider these (dis)abilities in realizing services.

Chapter 7

Experimental Prototype

"You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete."
— Richard Buckminster Fuller

In this chapter, we discuss the prototype implementation of the proposed tailoring platform. The prototype is a proof of concept to be validated using realistic scenarios in our case study. Before implementing the tailoring prototype and its underlying functionality, we designed mock-ups of four different representation of the tailoring platform and process (i.e., four different user interfaces) that helps care-givers to understand how the tailoring process works and to figure out what the tailoring platform should look like. The goal was to acquire feedback from care-givers in order to align the service tailoring functionality with the care-givers' needs and constraints; while avoiding the need to make expensive changes after implementing the platform. We have implemented the prototype in Java. For the pilot, we implemented four care monitoring services namely *blood pressure*, *oxygen saturation*, *weight*, and *medicine reminder*

In Section 7.1, we explain how we implemented the prototype based on the proposed tailoring architecture. In Section 7.2, we discuss several different possible tailoring user interfaces and we explain how we used the mockups to select the most acceptable interface. After selecting the interface, in Section 7.3, we illustrate the use of the user interface of the tailoring platform using the *oxygen saturation* monitoring task (as this was one of the tasks which was implemented for the case study). In Section 7.4, we discuss the structure of the database of the prototype and finally, in Section 7.5, we summarize the chapter.

7.1 From the Tailoring Architecture to a Prototype

As a proof of concept, we developed a prototype of our service tailoring platform, as part of the U-Care project. Fig. *7-1* presents the U-Care system. The U-Care system comprises three main platform components which are:

– A *tailoring platform* to enhance creation of the service plans by care-givers. The tailoring platform is the focus of this thesis.
– A *provisioning platform* [171] to execute the service plans, and to integrate and orchestrate the application services as required by the service plans.
– A *service repository platform* [158] that is a collection of application services. Some of these application services are implemented by the U-Care system such as reminder, calendar, and alert services, while others are implemented by third-party providers outside the U-Care system, such as blood pressure monitoring and medication dispensing services.

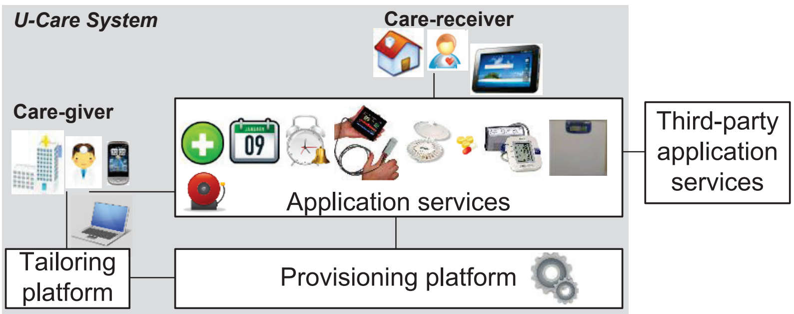

Figure 7-1 : U-Care system plus application providers

As discussed in Section 5.1, service tailoring is mainly about enabling end-users to configure the parameters of each SBB and to edit the composition of SBBs to create an integrated service. We do not ask care-givers to edit the composition of the SBBs, but instead we provide them with different alternative treatment patterns for each task, from which a care-giver can select a suitable treatment pattern according to the requirements of a specific care-receiver.

Different alternative treatment patterns are implemented and deployed in the provisioning platform using IBM's WebSphere Lombardi Edition [72]. For each individual care-receiver, care-givers choose the treatment pattern that best suits the patient. For a detailed explanation regarding implementing the treatment patterns in the provisioning platform, interested readers are referred to [171].

Within the tailoring platform, we designed a GUI to acquire values for the configuration parameters of each SBB from the care-givers. After the care-giver enters all of these required values and creates the service plan, the service tailoring platform calls a web service of the provisioning platform to inform it that there is a new service plan for a pair of care-receiver and task, and also to pass the configuration values to the provisioning platform. The tailoring platform also stores the configuration values in a database so that they can be retrieved later if the care-giver wants to re-tailor the previously created service plan. In the remainder of this chapter, we explain how we designed the user interface for service tailoring, illustrate the implemented user interface using the *oxygen saturation* monitoring task as an example, and, we describe the database structure of the prototype.

7.2 Several Representations

We have used mockups and shown different user interfaces for service tailoring to the care-givers in our case study. In computer science, mockups are mainly used by designers to acquire feedback from users. Mockups are a way of designing user interfaces on paper or as computer images without implementing application's underlying functionality. We have shown four different interfaces to the care-givers and we selected the one which received the most positive feedback from the care-givers to implement. We do not claim that the selected one is in general the most suitable user interface for service tailoring in the homecare domain, if there is such a thing, since we had only one case study and a limited number of care-givers. Nevertheless, the selected user interface was used by domain experts, care-givers, to tailor the four services. They evaluated it as usable. In the following, the four different interfaces are described:

– **Table-like Interface:** The first proposal was to provide care-givers with a table (Excel-like spreadsheet). This table showed all the configuration parameters and decision rules together. Then, we asked the care-givers to configure them for a specific care-receiver and task by selecting appropriate values from the drop-down menu or by entering text values into cells. However, the table eventually became too complex as it had a lot of rows, columns, and data fields which made the manipulating of the table difficult for the care-givers. So we dropped this interface.

– **Rule-like Interface:** For this interface, we did more than a mockup and actually implemented the *blood pressure* monitoring task. We used WebSphere ILOG JRules [71] to model our decision rules and configuration parameters. This enables us to define the rules in plain text, which we thought would be easy to understand by the care-givers. The data model of the rules (SBBs), called an eXecutable Object Model (XOM), can be used by its corresponding Business Object Model (BOM) and rules templates in the rule editor to modify the behaviour of the application.

  As shown in Fig. *7-2*, to tailor a blood pressure task for *Jan* as an example, a care-giver can decrease the maximum systolic level by changing the rules' parameters (printed in green color) in the rule editor. In this example, the diastolic and systolic blood pressures are the business objects and *"the diastolic blood pressure is more than"* is a rule template. By clicking on the plus/minus symbols, the care-giver can change the objects or their values. It can be specified what part of the rules the care-givers can/cannot edit. However, the care-givers found it very complex to work with the WebSphere ILOG JRules editor, so we also dropped this interface.

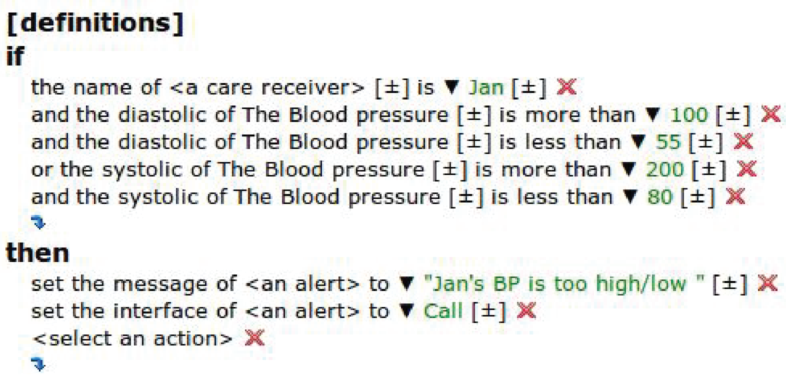

Figure 7-2 : An example of Rule-like interface for the blood pressure monitoring task

– **Process-like Interface:** As shown in the Fig. *7-3*, the third alternative interface provided a care-giver with a graphical interface of a treatment pattern in BPMN-like notation, whenever a task and care-receiver were selected. Then, the care-giver could click on the data items of each activity to enter the required values for the configuration parameters of each SBB. Since during the interview, we had used similar diagrams to identify the treatment patterns, we thought this interface would be usable by them. However, compared to the wizard-based interface explained below, the process-like interface was mostly rejected by the care-givers. We concluded that the care-givers prefer a wizard-like interface, because this resembles the interface of the applications they work with in other parts of their work.

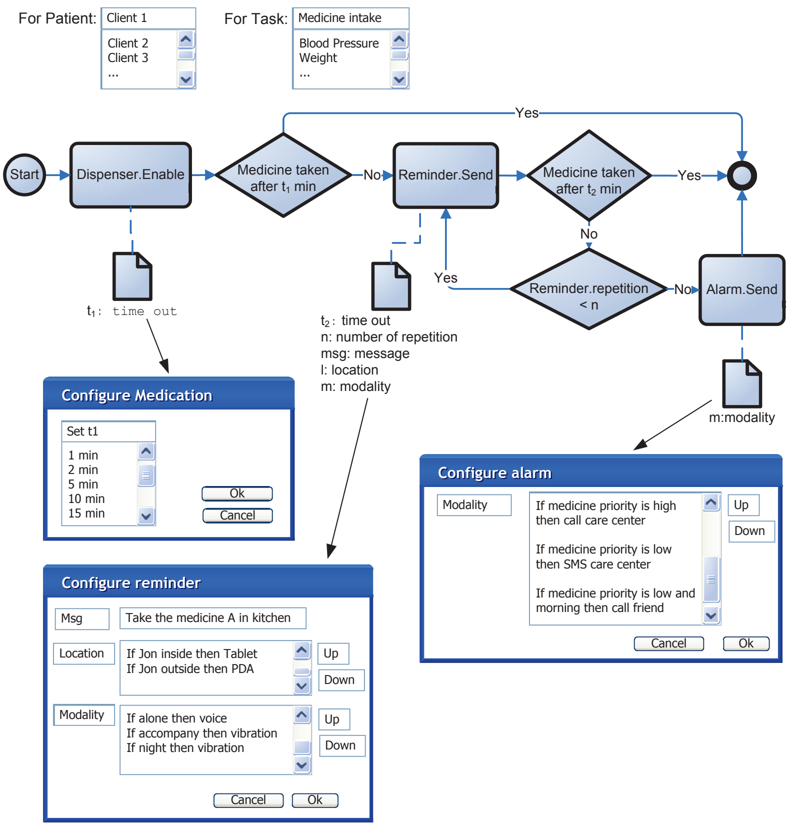

Figure 7-3 : A mockup of process-like interface for the medication intake support task

– **Wizard-like Interface:** The wizard-like interface, shown in Fig. *7-4*, received the most positive feedback from the care-givers, thus we implemented it in our prototype. This interface requires the care-givers only to configure one activity (SBB) of the service plan at a time in different steps. This interface is explained in the next section.

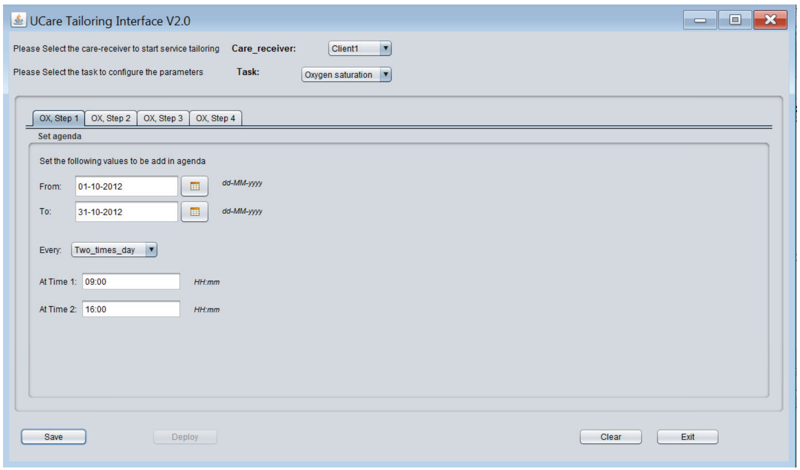

Figure 7-4 : An example of Wizard-like interface for the oxygen saturation task

7.3 Implemented GUI for the Prototype

As shown in Fig. *7-5*, service tailoring process starts when the care-giver selects the name of the intended care-receiver. For privacy reasons, we have used pseudonyms, client 1 to client 8, in the prototype. Only the care-givers were aware of the real name of the care-receivers. We entered these names in database and they were retrieved when the care-giver tailored the services. In a real setting, the care-receiver's names could be entered by the care-giver while she is entering the values for the care-receiver's user profiles. Moreover, in a real setting, each care-giver is responsible for approximately 8-10 care-receivers. The tailoring platform could show only the names of relevant care-receivers for whom the care-giver is responsible.

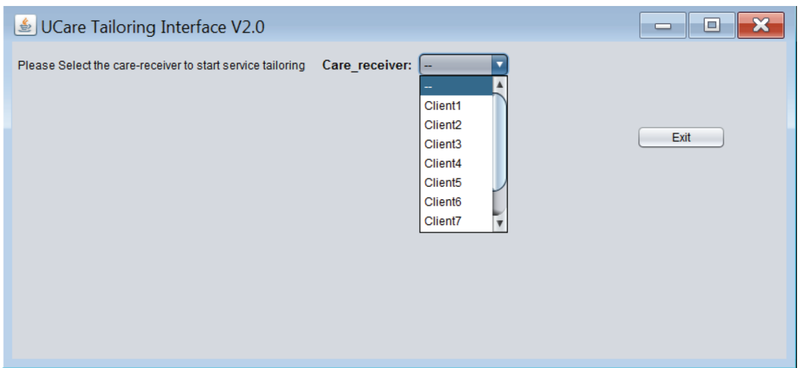

Figure 7-5 : Tailoring GUI, selecting a care-receiver

After selecting the name of the care-receiver from a drop-down menu, as presented in Fig. *7-6*, the care-giver should select for which task she is going to create a service plan. For the case study, we implemented four care services (namely *blood pressure monitoring*, *oxygen saturation monitoring*, *weight monitoring* and *medication intake support*). In a real setting, the list of tasks would be less than 10 tasks in most cases, and those tasks could be filtered based on the care-receiver's needs indicated in his user profile.

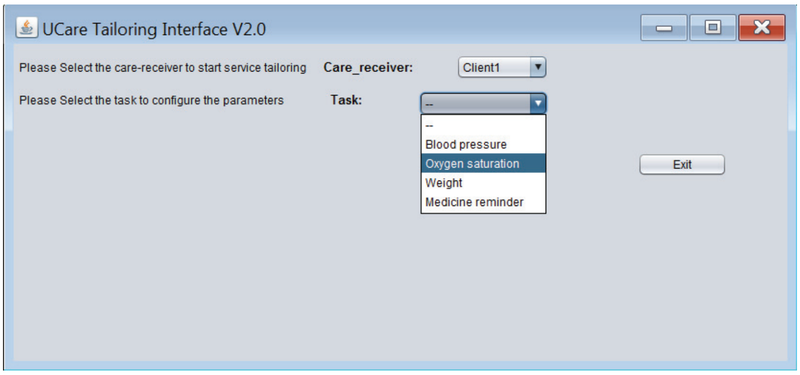

Figure 7-6 : Tailoring GUI, selecting a task

After selecting a task from the drop-down menu, in the lower part of the page, we can see different tabs (steps) for each task (see Fig. *7-7*). Each time the care-giver tailors only one task per care-receiver. If she wants to create a service plan for another task or another care-receiver, she need to start the process from the beginning. Another alternative could be a service plan that covers more than one task. For example, by giving an option by which a care-giver could select more than a task when creating a service plan, then the different steps of each selected task could be presented one after each other.

Until this pint, the steps are the same for all tasks, but from now forward, each task has its own steps, nevertheless, these steps are quite similar among different tasks (this is because the same SBBs are used for each task). In the remainder of this section, we only show and explain the steps of the *oxygen saturation* monitoring service.

As shown in Fig. *7-7*, the oxygen saturation monitoring task has four steps. Three of these steps (*Step1*, *Step 2*, and *Step 4*) are mandatory which means the care-giver has to provide some inputs to them before being able to deploy the plan, but *Step 3* is optional and she can leave this step out. In the following, we explain the purpose of each step and its required configuration parameters.

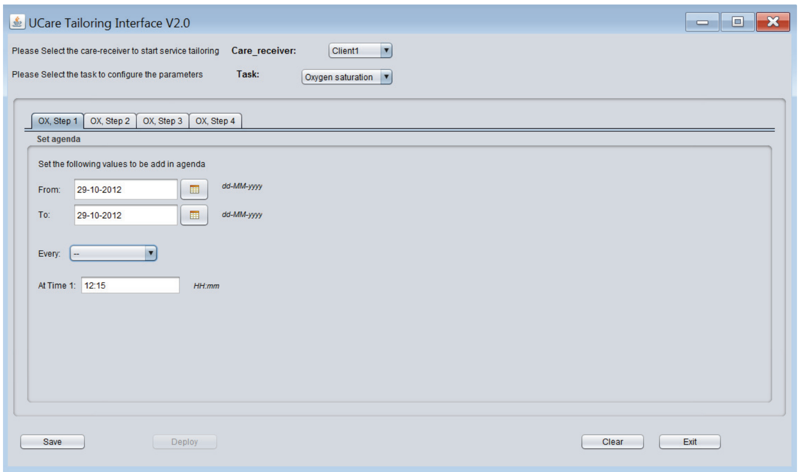

Figure 7-7 : Tailoring GUI, steps of creating a service plan for oxygen saturation monitoring service

Step 1

This step is for setting an agenda. As shown in Fig. *7-8*, in this step, the care-giver should provide configuration values for agenda SBB. She should fill starting date (From, in the format of dd-MM-yyyy), ending date (To, in the format of dd-MM-yyyy), time (At, in the format of HH:mm) and the frequency (Every, with inputs of every day, every two-days, every week, every month, two-times per day, three-times per day, and four-times per day). After she fills in these four parameters, she can press the *Save* button. Note that the *Deploy* button still is inactive and this means that there are other parameters in other steps which the care-giver needs to fill in before being able to deploy the plan.

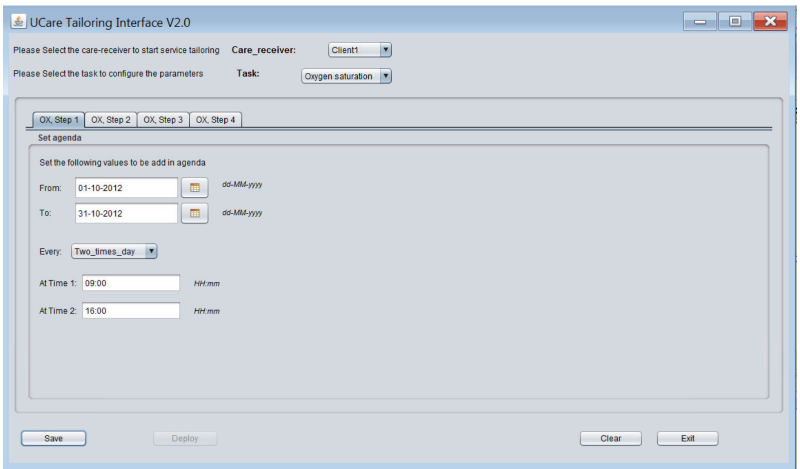

Figure 7-8 : Tailoring GUI for oxygen saturation monitoring service plan, setting the agenda parameters

Step 2

This step is for setting the validity of last measured oxygen saturation level and creating a reminder. As you can see in Fig. *7-9*, in this step, the care-giver should provide input for the reminder SBB's parameters. First, she should specify the validity of the last measured oxygen level. If she chooses 2 hours, this means that if the care-receiver has measured his oxygen saturation level within last 2 hours of the scheduled time, the measured values are valid and there is no need for sending a reminder to (re)measure it.

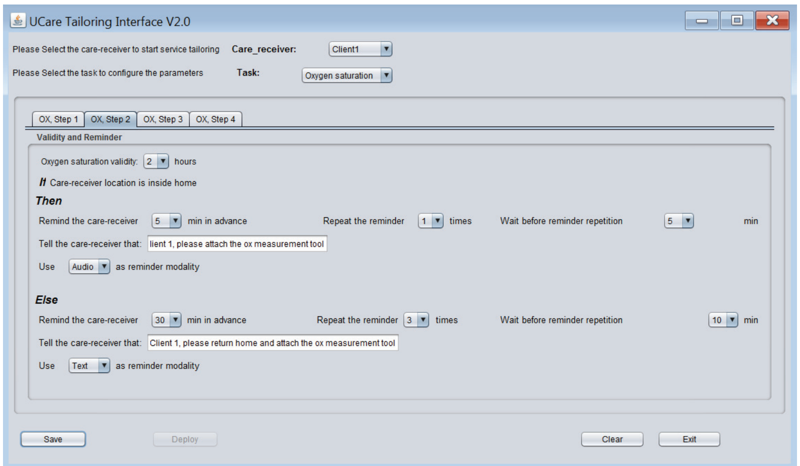

Figure 7-9 : Tailoring GUI for oxygen saturation monitoring service plan, setting the validity and reminder parameters

To complete the rest of this step, the care-giver needs to provide inputs for the reminder SBB in two possible situations: when the care-receiver is inside or outside their home. In the case study, we only consider *time* and *location* as context parameters. In each of these situations, the care-giver uses a drop-down menu need to specify how much time in advance the care-receiver should be reminded, how many times the reminder should be repeated, how much time to wait between reminder repetitions, and via which modality to send the reminder (such as text, audio, or video). She should also specify the reminder message in a text box. Note that if the care-receiver is inside his home, probably he does not need to be reminded too much in advance and the number of messages can be few, while when he is outside, he needs more time to go back to the home, so he should be reminded earlier and he may need more repetitions. In the same way, the message and the modality could also be different for inside and outside the home.

Step 3

This step is for specifying an appropriate treatment pattern. As explained before, to support different possible orchestrations of SBBs, we provide alternative treatment patterns for a task. In the beginning, we thought that a care-giver could choose a suitable pattern, if we showed her the alternative patterns. However, this proved to a difficult and time consuming task, since the care-giver needs to compare them, distinguish their differences, and select the proper one. For these reasons, we add some explanations beside each pattern to make the selection easier. Later, we learned that we did not need actually to show them the patterns as this could be easily covered using questions in a step.

For example, for the oxygen saturation monitoring task, there are two patterns. In one of them when the oxygen saturation is high/low but it is not at a dangerous level, then a care-receiver can control this level himself by taking medicine. In the other pattern (which is the default if the first pattern is not specified by the care-giver), because of the care-receiver's situation, the care-giver prefers to receive an immediate alert when the latest measurement is high/low. As you can see in the Fig. *7-10* and Fig. *7-11*, *step 3* is designed for this purpose and it is an optional step. The care-giver specifies the pattern with the possibility of taking medicine by pressing the check box: "*Ask care-receiver to take medicine for high/low oxygen saturation?* ". The care-giver should provide values for high/low oxygen saturation levels and enter the message which will be shown to the care-receiver to take medicine.

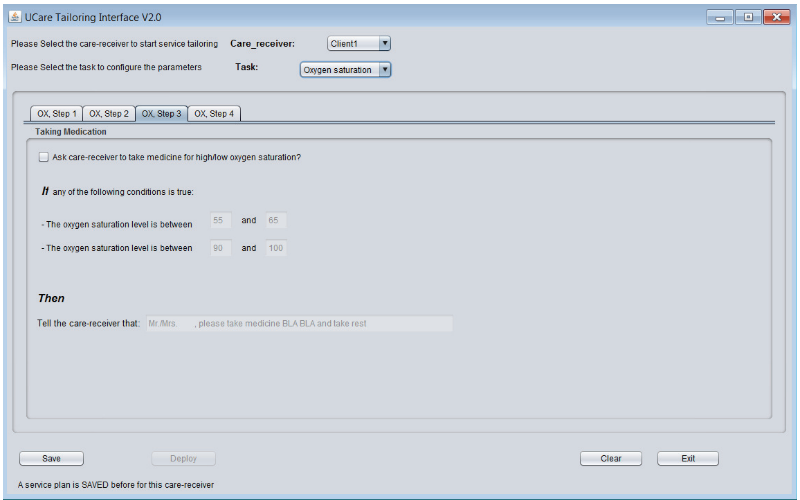

Figure 7-10 : Tailoring GUI for oxygen saturation monitoring service plan, specifying alternative patterns

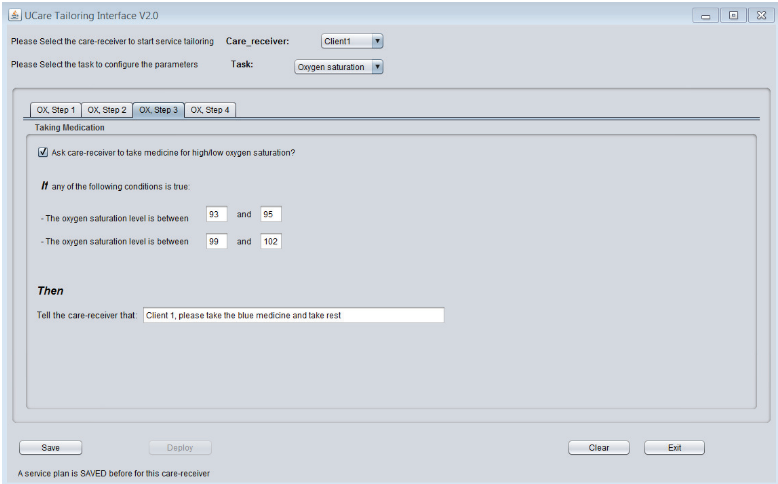

Figure 7-11 : Tailoring GUI for oxygen saturation monitoring service plan, setting the taking medication parameters

Step 4

Step 4 is the final step in the oxygen saturation monitoring task. This step is presented in Fig. *7-12*. In this step, the care-giver should provide input of the alert parameters. First, she should fill in the threshold levels for high/low oxygen saturation levels. This means that if the care-receiver's oxygen level is outside of the provided range, then an alert message will be send to the care-giver.

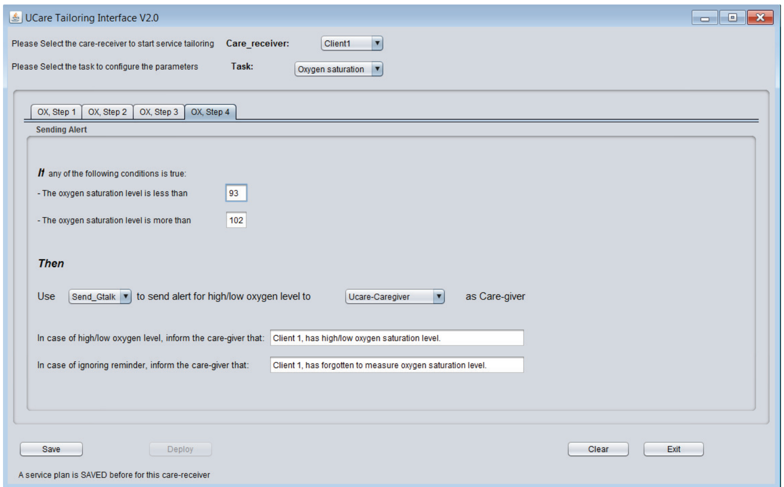

Figure 7-12 : Tailoring GUI for oxygen saturation monitoring service plan, setting the alert parameters

We limit the low value to two digits and high value to three digits. This is because if the care-giver by mistake enters fewer or more digits, the software will remove the value automatically and the care-giver should re-enter the values. This way we could prevent entering some of the wrong values by mistake. The care-giver should also specify in which way (through Gtalk, phone, sending SMS, or Skype) they want to receive the alert messages. During the case study we only used Gtalk. Finally, the care-giver should write the alert messages and specify to whom these messages should be sent. There are two alert messages: when the care-receiver has high/low oxygen saturation level and when he ignores a reminder and forgets to measure his oxygen saturation level after the valid period ends.

If the care-giver provides all the values required for configuration parameters in the steps, the *Deploy* button will be enabled. This button will deploy the created service plan to the provisioning platform. As can be seen in Fig. *7-13*, after pressing the *Deploy* button, a new window will pop-up which summarizes the entered values for the created service plan. This final check is done before the deployment. If the care-giver confirms it, the service plan will be deployed and the user interface will return to the selection of the care-receiver step, ready for next care-receiver/task service plan creation.

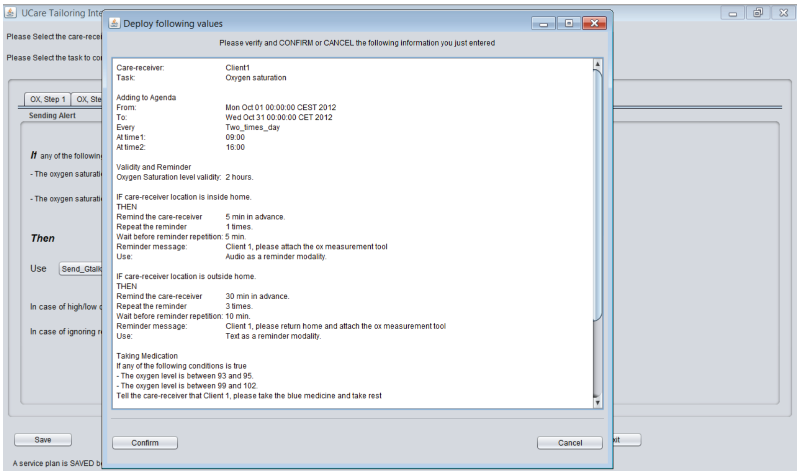

Figure 7-13 : Tailoring GUI for oxygen saturation monitoring service plan, checking the inserted values and confirming it

7.4 Database Structure

Fig. *7-14* presents the conceptual model of the database structure of the service tailoring prototype. It consists of 13 tables. As discussed earlier, a service plan is created per the pair: *Care-receiver* and a (homecare) *Task*. We have 6 tables to store configuration parameters of the SBBs. Three of these SBBs, *Calendar*, *Reminder*, and *Alert* are task *independent* SBBs which can be utilized in any task and the other three, *BP*, *WT*, and *OX* are task specific (as they store the configuration parameters of blood pressure monitoring, weight monitoring and oxygen saturation monitoring tasks, respectively). Note that for the medication intake support task, we do not define any task specific table and we could store all the required configuration parameters in the task independent tables.

The *Care-receiver* table is used to store the care-receiver's personal information and the *Care-giver* table is used to store the contact information of the care-givers who could receive alerts. Moreover, the *Treatment Pattern* table is used to store the treatment patterns.

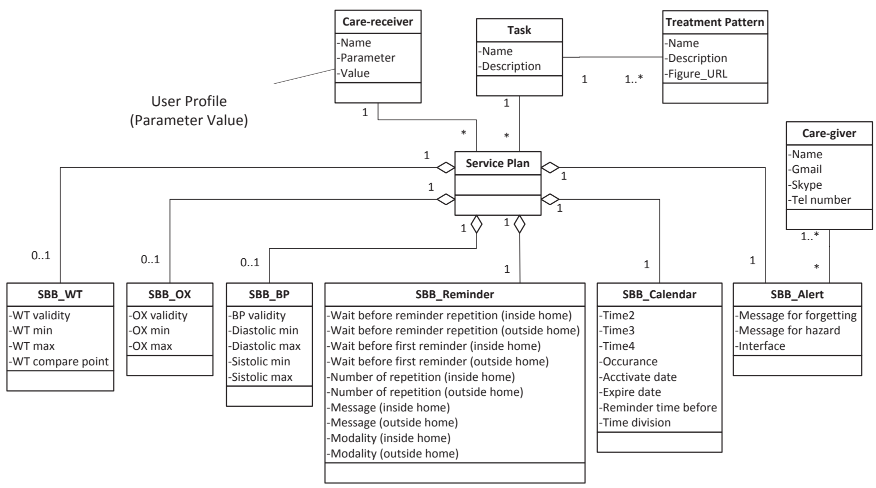

Figure 7-14 : Conceptual model of the service tailoring database

Fig. *7-15* presents how the database is implemented in DB2 [70]. Once a service plan is created, a *SERVICE_ PLAN_ ID* is associated with the *SERVICE_ PLAN*'s table. We distinguish between saved, deployed, and old service plans (previously deployed). If the care-giver only creates and saves a service plan to be deployed later, then its *PATTERN_ ACTIVE* entry has the value of "0", and if she deploys it, its *PATTERN_ ACTIVE* entry has the value of "1" indicating that this is a active service plan, and if a previously a service plan is deployed for that care-receiver and task, its $PATTERN\_ACTIVE$ entry has the value of “2” indicating that this is an old service plan. This way, we do not remove service plans from the database and we keep the history of them for future use.

There are other tables to store information about the date and time stamps of creation and modification of the service plans. Different steps of creating service plan are also logged. We used this information for debugging and analyzing purposes. These tables are excluded in Fig. *7-15*.

We did not store the identity of a care-giver who creates and/or modifies the service plans and any care-giver could tailor the services. However, in real setting, there would be more than one care-giver at different times of a day and on different dates who takes care of the same care-receiver. In the user profile of a care-receiver, the responsible care-givers could be specified to have access to the care-receiver information. Thus, there should be a logging step in the beginning of the service tailoring process that requires an authorized care-giver to enter her user name and password to log in. This way, the identity of the care-giver who creates and/or modifies the service plans can be controlled and stored. This is also necessary to keep track of changes made by different authorized care-givers for the accountability aspects (which is discussed in Chapter 4.

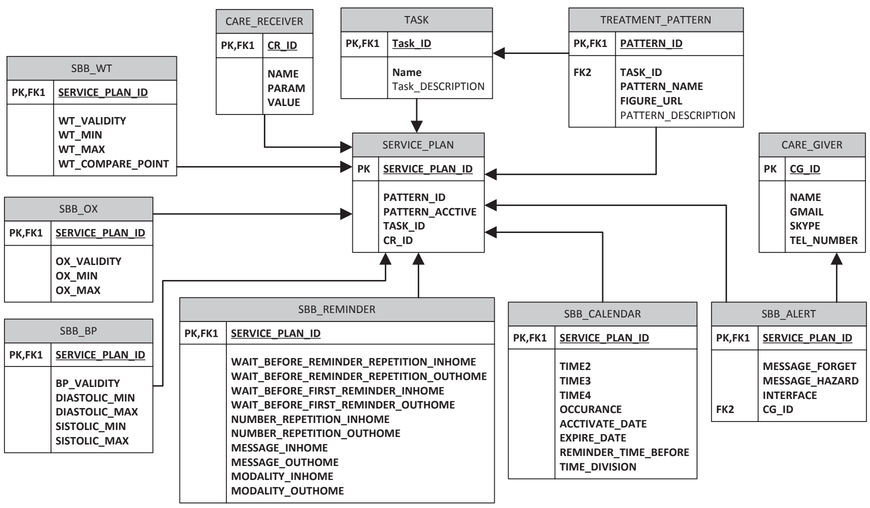

Figure 7-15 : Entity relationship diagram of tailoring database

7.5 Summary

In this chapter, we have described the prototype of our proposed solution. We have implemented four care monitoring services to be evaluated in the case study. We have also implemented another prototype for tailoring social activity services. The reason for separating the tailoring of care monitoring and social activity services is that the social activity services are tailored by other type of care-givers, namely social carers who are not allowed to decide about medical activities. We did not include a description of the prototype of the tailoring of social activities services, since these type of services are not the main focus of the case study.

We described how we designed the user interface for service tailoring by utilizing several mockups of possible user interfaces and presenting them to the care-givers. We have explained the implemented user interface by presenting the different steps that a care-giver should follow to deploy a service plan. We also explained the information that a service plan needs to contain in order to be deployed to the provisioning platform. This service plan is also stored locally in a database.

The prototype is a proof of concept for the proposed service tailoring in the homecare domain. It shows that such service tailoring can be implemented. In the next chapter, we validate it in a real life case study to assess its usability and usefulness in practice.

Chapter 8

Validation: Experiments and Results *

"They always say that time changes things, but you actually have to change them yourself."
— Andy Warhol

As discussed in the previous chapter, as a proof of concept, we have developed a prototype of the service tailoring platform. The prototype was subsequently used in two series of experiments in a case study to validate the properties of the approach and to improve the service tailoring platform. The goal of these experiments was to study the usability of the approach in terms of (a) *effectiveness*, i.e., to see whether the care-givers can use the tailoring platform to create service plans, (b) *efficiency*, i.e., to see whether the care-givers can tailor the services within a short time period, (c) *learnability*, i.e., to see whether the care-givers need less time to tailor services as they gain more experience with the tailoring service, and (d) *satisfaction*, i.e., to see whether the care-givers find the tailoring platform usable and meets their expectations. For satisfaction, we asked the care-givers about the care-receivers' opinion about using the U-Care services. Furthermore, we investigate possible improvements based on the care-givers' feedback.

*This chapter is based on the following paper:

– Mohammad Zarifi Eslami, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Care-giver Tailoring of IT-based Healthcare Services for Elderly at Home: A Field Test and its Results. In The Proceedings of The 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013.

We define an evaluation strategy, presented in Section 8.1, which we followed to evaluate the service tailoring platform. Section 8.2 presents the details of the experiments' setup and their organization under which the evaluation is carried out. Section 8.3 presents the results of the first series of experiments and the list of changes that could be introduced in the tailoring platform. We implemented those requested changes in the tailoring platform for the second series of experiments in which we conducted a similar set of evaluations as in the initial experiments. Section 8.4 presents the results of the second series of the experiments, and finally, Section 8.5 summaries this chapter.

8.1 Validation Criteria

The goal of the experiments of the field test was to evaluate the *usability* of the approach. Since only a few subjects participated in the experiments, we cannot generalize statistically from the results, and we regard this simply as an experimental case study. We will use these measurements from the experiments to identify possible improvements to our approach, but will not generalize as to the set of (all) possible applications of the current version of the approach. For this reason, we explain our observations as this allowed us to understand which parts of our approach needed improvement.

Measuring usability is a particularly difficult problem, because usability is not one-dimensional, but emerges as a multidimensional characteristic in the context of users performing tasks with a system in a specific environment [17, 135]. Most usability evaluation methods gather both subjective and objective quantitative data. Subjective data are measures of participants' opinions or attitudes concerning their perception of usability. Objective data are measures of participants' performance, such as scenario completion time and successful scenario completion rate.

To evaluate the usability of our approach, we use the usability criteria of the *NISTIR 7432* standard [123]. It defines usability (in compliance with ISO 9241-11 [1]) as: "The extent to which a product can be used by specified users to achieve specified goals (an intended outcome) with *effectiveness*, *efficiency*, and *satisfaction* in a specified context of use". The standard provides guidelines to measure effectiveness and efficiency, which results in objective data, and to measure satisfaction, which delivers subjective data. Furthermore, as another aspect of the usability, we measured the *learnability* of the service tailoring user interface.

8.1.1 Effectiveness

Common measures of effectiveness include *task completion rate*, *frequency of errors*, and *frequency of assists* to the participant from the testers. The effectiveness measurement indicates the accuracy and completeness with which users achieve specified goals (here viewed as task completion by users). It does not take into account of how the goals were achieved, only the extent to which they were achieved. Effectiveness can be scored on a scale of 0 to 100% based on specified criteria.

8.1.2 Efficiency

Efficiency relates the level of effectiveness achieved to the quantity of resources expended. Efficiency is mainly assessed by the mean time taken to achieve a task. It may also relate to use of other resources (e.g. total cost of usage). Task time values are useful when making comparisons between systems.

In order to measure efficiency, we measure the task completion time of end-users (care-givers) and we compare this with the task completion time of an 'expert' (someone who is familiar with the system and the technology used, but not a domain expert). The comparison will give an indication of whether or not the technology used is a hindrance for using the system. This is similar to *relative user efficiency* as defined in the literature, i.e., how long a user takes in comparison with an expert [123]. However, here the values obtained from the experiments have no statistical relevance due to the small number of participants.

8.1.3 Learnability

According to ISO/IEC 9126 [77], learnability is the capability of a software product to enable the user to learn how to use it. Learnability is an important aspect of usability, i.e., if users cannot easily learn to use a system, for example, by following simple instructions and/or just by trying out, they will simply ignore the system. In this thesis, we want to measure the degree to which the user interface of the service tailoring platform can be learned quickly and effectively. To do so, first we analyze the learning time where we measure the service plan creation time by the care-givers during the first series of the experiments and we observed if this time is decreased during the experiments for same or similar tasks. Second, we perform the second series of the experiments two month after the first one, and we observed if the care-givers still remember how to create service plans *without* our assistance.

8.1.4 Satisfaction

Satisfaction describes a user's subjective response when using the system (expression of perceived usability). The satisfaction measurement shows if the experience was freed from discomfort and gives positive/negative evaluation of the experience of using the system.

In this thesis, the satisfaction analysis allows us to learn the caregivers' perception of the comfort and acceptability of the tailoring platform. In order to measure user satisfaction, we used a *questionnaire* method. We prepared a questionnaire based on the *Computer System Usability Questionnaire* [98].

The questionnaire we prepared (shown in Figures *A-1* and *A-2*) contains two types of questions: close-ended questions directly related to the subjects and open-ended attitudinal questions to uncover people's beliefs and thoughts on a subject. The close-ended part includes 19 questions and each question can be answered using a 7-point scales, anchored at the end points with the terms "Strongly disagree" for 1 and "Strongly agree" for 7, and a Not Applicable (N/A) point outside the scale. Thus, higher numbers are used to represent higher usability of the system.

Following the guidelines in Lewis [98], the results from user satisfaction are summarized into the following factors:

– *(perceived) Overall system usability (OVERALL)* - Questions 1-19 are used to measure the overall perceived usability. One interesting point is that questions 1 & 19 both ask about the overall usability with different words (one in the beginning of the questionnaire and the other at the end), to see if the user provides consistent answers before and after answering other questions.

– *(perceived) System usefulness (SYSUSE)* - Questions 1-8 are used to measure perceived system usefulness. These questions actually concern the perceived effectiveness and efficiency from the user's perspective.

– *(perceived) Information quality (INFOQUAL)* - Questions 9-15 are used to measure the perceived information quality, which is the quality of information provided by the system and/or by the developer(s) of the system to guide and assist a user in using the system.

– *(perceived) Interface quality (INTERQUAL)* - Questions 16-18 are used to measure the perceived quality of only the user interface (rather than the whole system).

The *OVERALL*, *SYSUSE*, *INFOQUAL*, and *INTERQUAL* factors are reported as mean values, following the guidelines in Lewis [98].

There were three questions for the open-ended part, where we ask the care-givers for the positive and negative points, as well as any suggested improvement that they see possible in the tailoring platform and the overall system.

8.2 Setup of the Experiments

As presented in Fig. *8-1*, the U-Care system was used in a field test with two series of experiments to validate the properties of the approach and to improve it. We follow the guidelines described by Wieringa in [165] to perform the experiments systematically.

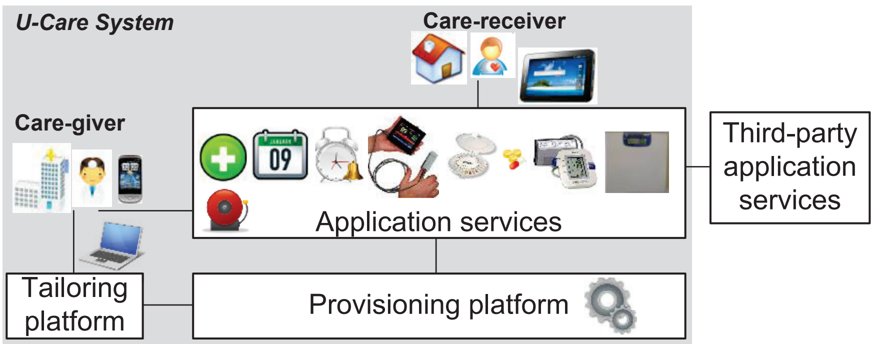

Figure 8-1 : U-Care system plus application providers

Each series of experiments was conducted in a near real-world setting in a care institute in the Netherlands and each series lasted for two months. The setting of the experiments is *near* real-world, because some real-world aspects are present, such as real care-receivers, a real care institution, real nurses, and realistic scenarios, but some other aspects are absent, such as only a single homecare institution, a limited number of end-users, and no use of real medicines. In this section, we first explain which actors participate in the experiment and the role of each one, then we describe the scenarios that were used in the experiments, we explain how we collect data, and finally we describe which instruments and services were used in the experiments.

8.2.1 Actors

Several actors have participated to cover the range of expertise required and to provide the facilities and environments needed to perform the experiments. The participants consists of one research partner, Centre for Telematics and Information Technology of the University of Twente (UT-CTIT), a healthcare partner, Orbis[1], and three information technology partners (IBM[2], MobiHealth[3], and Innospense[4]).

UT-CTIT, the research partner, includes three research groups: Information Systems (Computer Science department), Remote Monitoring and Treatment (Electrical Engineering department), and Industrial Engineering & Business Information Systems (Management and Governance department). These three groups studied the goals of the project in terms of four different aspects: service tailoring, service provisioning, user interfacing, and business feasibility.

The information technology partners provided knowledge of IT solutions, application services, and industry standards. They also have experience applying such solutions in practical settings, including healthcare. For example, MobiHealth provides IT services for vital signs monitoring. Their services were used in the experiments to measure blood pressure, oxygen saturation, and weight. Innospense provides an electronic medicine dispenser device and service which were used in the experiments to guide a person in taking the proper medicine at the correct time and in the correct amount. There exist simple medicine dispensers that only function as a pill sorting box (we refer to this as a manual medicine dispenser). Also dispensers exist which can automatically dispense medication to persons and can give notifications at specified times (we refer to this as an automatic medicine dispenser). Our experiments employed both manual and automatic medicine dispensers. Integrating these third party services demonstrates that the U-Care system enables decentralized applications, i.e., utilizing distributed components and services from independent service providers.

Orbis, the healthcare partner, owns residential blocks where the elderly can live and receive care services provided through professional care-givers. The aim of this institution is to provide round the clock services to their care-receivers and at the same time to enable them to live an independent life as much and as long as possible. For performing the experiments, Orbis provided the application context and a test-bed for the research. They also participate in the development of scenarios

[1] http://www.orbisconcern.nl/
[2] http://www.ibm.com/nl/nl/
[3] http://www.mobihealth.com/home/en/home.php
[4] http://www.innospense.com/index.html

and the derivation of user requirements, as well as in evaluation of the usability of the prototype. Eight care-receivers (identified as Client 1, Client 2, ..., Client 8) and four care-givers volunteered to use the U-Care system in two series of experiments. None of these care-receivers had ever used any IT-based system (e.g., computer, laptop, Tablet-PC, or smart phone) previously.

8.2.2 Scenarios

We considered two types of homecare services, *care services* and *social activity services*, in the validation experiments. We consider four different care services: blood pressure monitoring (BP), oxygen saturation monitoring (OX), weight monitoring (WT), and medication intake support (MD). Together with the care-givers we defined the following scenarios to be experimented with Clients 1 to 8. We involved the care-givers in defining the scenarios in order to make them as realistic as possible. The motivation for using these specific scenarios in the experiments are the individual needs of Clients 1 to 8; the scenarios combine different use of the configured services to meet the needs of the care-receivers.

– *Manual MD + BP + WT*: In this scenario, a manual medicine dispenser is used; the care-receivers are asked to take their medicines (in the experiments, we used candies instead of medicines) using manual MD, and to measure their blood pressure and body weight using BP and WT, respectively, based on the plan created by the care-givers.

– *Manual MD + BP*: Same as previous scenario but only supporting medicine intake and monitoring blood pressure using MD and BP, respectively.

– *BP + WT*: Same as previous scenario but only monitoring blood pressure and body weight using MD and WT, respectively.

– *Automatic MD + OX + BP*: In this scenario, an electronic medicine dispenser is used (only for routine medication and not for medicines in hazardous situations); the care-receivers are asked to take their medicines (candies instead of medicines) using Automatic MD, and to measure their oxygen saturation and blood pressure using OX and BP, respectively.

8.2.3 Measurement Instruments

To collect data, we installed a screen capture software package, and asked the care-givers to run this software whenever they want to create a service plan. Capturing the screen helped us to record the care-givers' activities and behavior while creating service plans, so we could analyze and see in which part of the application they have difficulty understanding and performing the required actions. We also used screen capturing (with time recording) to measure the amount of time spent creating a service plan.

Furthermore, after each series of the experiments, we interviewed the care-givers and care-receivers who participated in the experiments to collect their opinion about the service tailoring, which could complement or explain the collected results and be useful in improving the U-Care system.

8.2.4 Devices and Services Used in the Experiments

The care-givers received a laptop with the installed service tailoring software on it which they use to create service plans. The created service plans are sent through the Internet to a tailoring server, and deployed and executed in the provisioning server (both servers were located at the Computer Science department of UT-CTIT). The care-givers also had a smart phone on which they received alert messages.

The care-receivers received three type of sensors to measure their own vital signs (blood pressure, oxygen saturation, and/or weight), one medicine dispenser, one smart phone running Android operating system, and one Tablet-PC running Android operating system. The measured vital signs' values are transmitted from the sensors to the smart phone via Bluetooth and then the smart phone transmits this data through the Internet to MobiHealth's local server. The MobiHealth server pushes these values (without storing them) to the U-Care provisioning server. Then, after successfully receiving the values, the provisioning server analyzed the values to make decisions about (not) sending alerts based on the threshold values (indicated in the service plans). Finally, the provisioning server sends the values to an application server (located at the Electrical Engineering department of UT) and this sever presents the values via the care-receiver's Tablet-PC.

The communications channel among the care-givers' laptop, the tailoring server, the provisioning server, the MobiHealth server, the application server, and the care-receivers' Tablet-PCs is the Internet and used Web-services technology. Note that in practice, the tailoring, provisioning, and application servers could be run in one physical server and could reside in one physical location, e.g., in the care center or on the service provider's premises.

As mentioned earlier, MobiHealth, as a third party service provider, provides IT services required for measuring a patient's blood pressure, oxygen saturation, and weight along with the necessary sensors. Blood pressure is commonly described in terms of upper pressure (Systolic) and lower pressure (Diastolic) levels. The blood pressure measurement devices come in different forms and types such as a finger-clip, a pulse cuff, and an upper arm cuff. As illustrated in Fig. *8-2*, for our experiments, we use an upper arm cuff which is clinically accepted as one of the most reliable devices. Some of blood pressure measurement devices have network capabilities and can transfer data to software applications using different wireless communication links, such as Bluetooth, Wi-Fi, etc. In our experiment, we connected the blood pressure measurement device to a smart phone via Bluetooth.

Figure 8-2 : Blood pressure meter

In order to measure oxygen saturation (the amount of oxygen in the blood) and pulse rate, a pulse oxymeter is used. Pulse-oxymeters are commonly used to safety-monitor the training process of COPD patients. As blood pressure measurement devices, pulse-oxymeters also come in different forms and types, such as finger clips/rubbers, ear clips, and forehead reflectors. They can also be finger-top, handheld-based, tabletop based, or wrist-worn devices. Some pulse oxymeters can also communicate with external software systems using different communication links, such as Bluetooth, Wi-Fi, etc. and some are equipped with internal memory and USB ports such that the measured data can be manually transferred to external systems. The measurement environment and the situation of the user determines which combination of sensors is best to use (e.g., an ear clip may be better than a finger clip in situations where users are doing major exercises). As illustrated in

Fig. *8-3*, for our experiments, we select a finger-clip pulse-oxymeter which can communicate over Bluetooth. We selected this device because the care-receivers in our experiments will only be using pulse-oxymeters when they are not doing major exercises.

Figure 8-3 : Oxygen saturation meter

The primary function of a personal weight scale is to measure the weight of a person. However, more advanced weight scales also measure BMI, fat percentage, muscle/bone mass, metabolic rates, and more. Weight is a common vital parameter often measured for COPD, obesities, and heart-problem patients. Some of the weight scales can be connected to a computer system via Bluetooth or Wi-Fi. For our experiments, we selected a simple weight scale, as shown in Fig. *8-4*, (which measures only the weight) that can connect to a computer system via Bluetooth.

Figure 8-4 : Weight meter

Fig. *8-5* illustrates the Android based (running version 2.3.6 of the Android OS), Google Nexus S, smart phone capable of running Java software. The smart phone receives the measured values of blood pressure, oxygen saturation, and the weight of the care-receiver through Bluetooth and transfers them to the Mobihealth server via the Internet using a Wi-Fi connection to a local Wi-Fi access point located in the care-receiver's home. This smart phone acts as a gateway between the measurement devices and the Mobihealth data collection/analysis server. We used an encrypted Bluetooth communication between the measurement devices and the smart phones (the data in the smart phone was stored temporarily until sending it to the Mobihealth server), whereas we use a secure HTTP connection between the smart phone and the third-party data collection server.

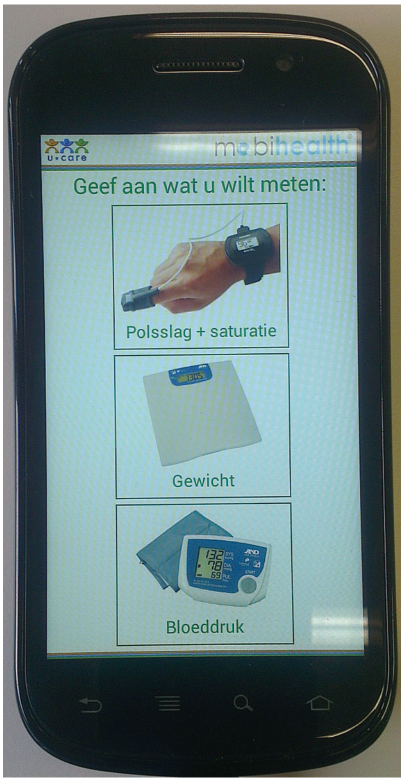

Figure 8-5 : Smart phone

We used an electronic medicine dispenser device and service from Innospense company (for the automatic medicine dispensing scenario). This device is illustrated in Fig. *8-6*. It is used to guide a person to take the proper medicine at the correct time and in the correct amount.

Figure 8-6 : Medicine dispenser

Fig. *8-7* illustrates elderly person's user interface application service, which was designed and developed by the Remote Monitoring and Treatment group of UT-CTIT. This interface is accessible via the web for the care-receivers. Using their Tablet-PC, care-receivers (and care-givers) can browse the functionalities of this application to see their reminder messages, measured values, and the scheduled events. Moreover, they can communicate with their family members, friends and care-givers through this web application.

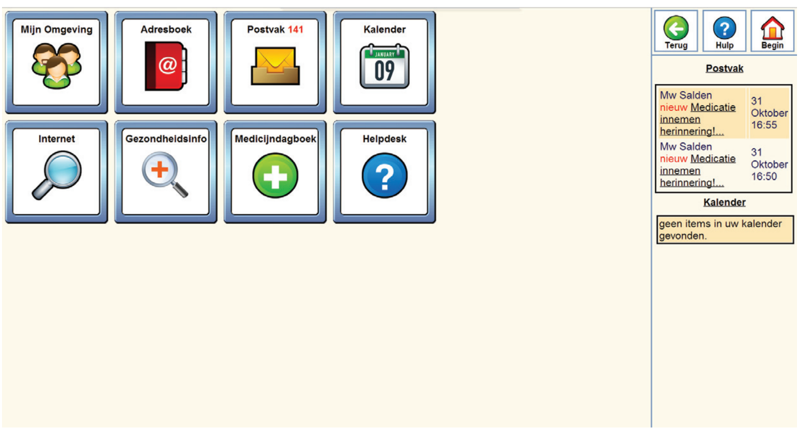

Figure 8-7 : Care-receiver's application interface

8.3 First Series of the Experiments and the Results

In the first series of the experiments, all eight clients participated in the social activity services, while five clients based on their needs and care-giver's recommendations, participated in the care services. The scenarios planned for in the vital sign monitoring application were the following:

- *Client 1* participated in scenario *Manual MD + BP*.
- *Client 4* participated in scenario *BP + WT*.
- *Client 6* participated in scenario *Manual MD + BP + WT*.
- *Client 2* participated in scenario *BP + WT*.
- *Client 3* participated in scenario *Automatic MD + OX + BP*.

As illustrated in Table *8-1*, due to the limited number of sensors and smart phones (i.e., two sets for each service), we could not test the scenarios in parallel and instead we scheduled them in three different time periods each lasting three weeks (one week of instructions and two weeks of usage). In period 1, *Client 1* & *Client 4*, in period 2 *Client 6* & *Client 2*, and finally in period 3, *Client 3* use the care services.

Table 8-1 : Participation of the care-receivers in the first series of the experiments

Time Periods	Clients				
	Client 1	**Client 4**	**Client 6**	**Client 2**	**Client 3**
Period 1	Manual MD + BP	BP + WT	-	-	-
Period 2	-	-	Manual MD + BP + WT	BP + WT	-
Period 3	-	-	-	-	[Automatic MD] + OX + BP

We could not integrate the automatic MD with the U-Care system for the first series of the experiments, thus *Client 3* participated only in the BP and the OX tasks and we postponed the use of automatic MD to the second series of the experiments.

Four care-givers participated in the experiments. Three of them are professional nurses who participated in the care services and one of them is a social carer who participated in the social activity services. Before the start of the experiments, we trained all four care-givers who participated in the experiments about how to use the overall U-Care system (including tailoring the services, measuring the vital signs, and checking the results using the care-receiver's application). The care-givers instructed the care-receivers how to measure their vital signs using the sensors and smart phones, and how to use the care-receiver's application using a Tablet-PC. This two stage method of training was used because of privacy issues and because care-receivers trust their.

The social carer created service plans for the social care services. Since a limited number of care-receivers participated in the experiments, only one professional nurse (we refer to this care-giver as *Care-giver 1*) was responsible for providing care activities (to those care-receivers) and she participated in creating the service plans for the different subsets of four care services. The other two professional nurses only created test plans for an imaginary care-receiver.

8.3.1 Usability Results

In the first series of the experiments, *Care-giver 1* created 11 service plans in total for 5 clients and covering four tasks: BP (5 plans), OX (1 plan), WT (3 plans), and MD (2 plans). In the remainder of this section, we present the usability results of the tailoring platform based on the validation criteria which were introduced in Section 8.1.

Effectiveness

To measure the *effectiveness*, we counted the number of the service plans created by the care-giver without our assistance. Only one service plan (for MD) out of 11 needed our assistance to be created. The care-giver thought that in order to schedule a medication task for two times per day, she should create two different service plans. We explained that the second service plan would overwrite the first plan and she could schedule the medication task for two times per a day with one service plan and specifying the twice daily base in the configuration parameters. This shows that care-givers should be properly instructed concerning this point. Also a warning from the tailoring platform about overriding the previous service plans could be useful to give an indication that there is an old service plans and this is useful for safety purpose. Note that this measurement of effectiveness is only based on the 11 plans which were created by one care-giver.

Efficiency

To evaluate efficiency, we measured the relative user efficiency when creating service plans. To do so, we asked a colleague who is a partner in the U-Care project to create a service plan for the BP task (we did not repeat it for the other tasks, since creating service plans for all tasks have similar steps). This choice was made to model the role of an IT-expert as this colleague was familiar with the underlying technology and the idea behind the services that were to be created. Then, we compared the ratio of time taken by the IT-expert with the care-givers when creating the same service plans. As presented in Table *8-2*, the IT-expert created the BP service plan for an imaginary care-receiver in 4:48 minutes, while the 3 care-givers participating in the experiments created the same service plan in: 9:23, 3:34, and 2:23 minutes respectively.

Table 8-2 : Service plan creation time for BP, by three care-givers and an IT-expert

Care-giver 1	Care-giver 2	Care-giver 3	IT-expert
3:34 minutes	9:23 minutes	2:23 minutes	4:48 minutes

The care-giver, who created the service plan in 2:23 minutes, is younger and has more experience using computers than the other two. And the care-giver, who created the service plan in 3:34 minutes, had previously created 11 service plans and thus, had the most experience with creating plans among our subjects. This indicates that experience with computers and/or with the tailoring interface are two important variables to measure if we want to predict the time needed to complete a service plan. This experiment showed that a care-giver who has general experience with using computers might create service plans faster than an IT expert with in-depth knowledge of the technology used in the system. We believe that in this case, domain knowledge is more important than IT knowledge. However, a care-giver with no computer experience at all needs more time than an IT expert (who lack domain knowledge). In this case having more domain knowledge does not compensate for the lack of IT knowledge (computer experience). Whether we can draw a conclusion from this concerning possible improvements of the system depends on the learnability of the system. If learnability (for people with no initial experience with using computers) is good, then there is no need for improvement. Our measurement of completion times of a few minutes for all plans and all nurses was encouraging, but cannot be generalized to other plans or other nurses.

Learnability

As illustrated in Fig. *8-8*, the time required for creating service plans decreased after the first two tries by our care-givers and then stayed roughly the same. Because this kind of interface was familiar to the care-givers, we think this learning behavior would happen with other care-givers too.

Service plans 5-9 were created two weeks after creating *service plans 1-4*, and *service plans 10-11* were created two weeks after creating *service plans 5-9*. Nevertheless, looking at the Fig. *8-8*, it is evident that the care-giver could remember the steps required in creating service plans even after two weeks, hence later plans took less time than the initial ones. Another interesting observation is that even though there was only one plan for the oxygen saturation task, *service plan 10* was created in a relatively short time due to the similarity of steps required to create service plans for the different tasks.

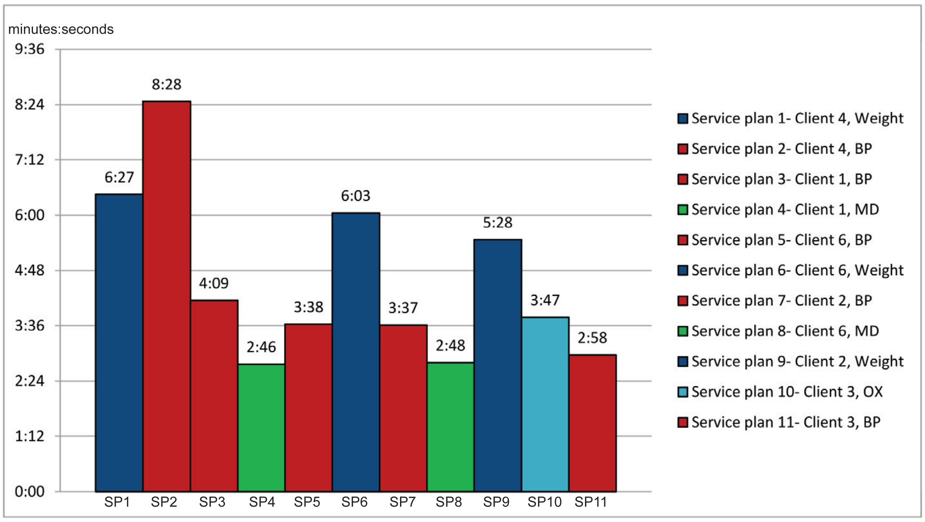

Figure 8-8 : Created service plans for different tasks and care-receivers in the first experiment

The service plans for the weight monitoring task took on average more time than other plans. This is because the care-giver created the plans for the weight monitoring task for Clients 5 & *2* (i.e., *service plans 6* & *9*) after creating the plans for the blood pressure monitoring task (i.e., *service plans 5* & *7*), and hence, she took some time to decide about the time of weight measurement so that it does not conflict with the blood pressure measurement time. This shows that the tailoring platform should provide some abstract information to a care-giver about previously created plans for a care-receiver while s(he) creates a new service plan, such that interdependencies can be tackled during the plan creation process. Another possible improvement is that a service plan could cover more than one task, allowing better handling of interdependencies between tasks. Which improvement is better has to be investigated in further experiments.

Satisfaction

After the execution of the first series of the experiments, we interviewed the care-givers and asked their opinions about the U-Care system in general and the service tailoring platform in particular. To measure *satisfaction*, we asked the care-givers to fill in the questionnaires, which were introduced in Section 8.1. They filled in these questionnaires three times: once for the whole U-Care system and two times only regarding the service tailoring (the professional nurses filled in the questionnaires for the vital signs services and the social carer filled it for the social activity services).

Table *8-3* summarizes the results for the System Usefulness (SYSUSE), Information Quality (INFOQUAL), and Interface Quality (INTERQUAL) of the whole system, and service tailoring platform for care services and service tailoring platform for social activity services (for the close-end questions). The values presented are the mean-values calculated from the values provided by the four care-givers.

Since the care-givers used the service tailoring platform themselves, the numbers reflect their opinion about the platform. While judging the whole system (care-receiver's application, third party services, and provisioning platform), they provided their answers from two perspectives: first, the amount of time and effort they needed to (re)train and assist the care-receivers in using the system, and second, the care-receiver's opinion about the system as perceived by the care-givers.

Looking at the Table *8-3*, we can see that the care-givers were more satisfied with the tailoring platform for creating care services plans, than for creating the social activity services plans and than with the whole system. The care-givers only interacted directly with the service tailoring platform. It had an interface of the kind they are familiar with in other hospital applications, and they felt comfortable with it. Perhaps when considering the whole system, they also assessed the interaction between the system and the patients. Moreover, looking at the satisfaction factors of the tailoring platform, we can see that *information quality* was judged low (5.17), because they could not get a list of service plans and parameters that they had already entered for a care-receiver. This relates to what has been said earlier when discussing learnability.

Table 8-3 : Satisfaction results for the first experiment

Score Name (1-7)	Average the Responses to	Whole System	Tailoring of Care Service	Tailoring of Social Activity Services
OVERALL	Items 1 through 19	4.75	5.67	4.84
SYSUSE	Items 1 through 8	4.37	6.00	5.37
INFOQUAL	Items 9 through 15	4.93	5.17	4.86
INTERQUAL	Items 16 through 18	5.32	5.67	3.33

Through subjective opinions from the second part of the questionnaire, we observed that the care-givers found the service tailoring interface and the process of configuring the care services easy and straight forward and it did not take too much of their time to create a service plan. They mention that “We create a plan in less than 5 minutes and it was enough for the whole period (of first series of the experiments)”. Moreover, they found the re-tailoring of previously created plans quite easy and quick. The re-tailoring was required if they monitored the behavior of the care-receivers and wanted to change some part of the created plan based on this experience and the care-receivers' usage behaviour.

They found the tailoring platform for the social activity services less optimal. The reason for this could be the main focus of the experiments was on the care services and we used the same design for the social activity services. Each plan for the care services was tailored for one care-receiver, while for the social activity services, the care-givers preferred to create group based services such as a “watching movie” activity. The reason for not considering the tailorability of the social activity services was that those services do not have too many tailoring possibilities and they only include scheduling different events and several activities and the care-receivers have a choice of participating/not participating in these events (we refer to this as On-Off tailoring). Thus, service tailoring these services was generally found to be unnecessary.

The care-givers found the user interface of elderly person's application not mature enough to be used by the care-receivers. For example, the care-givers say that “Client 4 has fear of using the system alone”. This took the care-givers extra time because they were frequently called to assist Client 4 in using the system, thus defeating the time-saving purpose of introducing the system. None of the care-receivers who participated in the experiments had ever used any IT-based system previously, and in a similar situation we expect similar extra calls to care-givers to be made. Designing a suitable user interface for the elderly remains a challenging task.

The care-receivers themselves, on average, have more positive opinions about the U-Care system. Client 1 (73 years old) and Client 6 (93 years old) were enthusiastic users of the U-Care services without the care-givers' help.

Client 6 was the volunteer to whom we gave a demo at the end of the first experiment, so we had a chance to meet the client in person and hear the client's opinions about the U-Care system. After giving a demo, where the client measured her blood pressure and weight using the system, and without the help of the care-givers, the client had the following reactions about different aspects of the system:

– *Independency* - The client indicated the feeling of being more independent, since for example, the client could measure her blood pressure every morning without needing to wait for a care-giver to come and measure the client's blood pressure. This could save the time of the client and indeed this could be considered as an important indication of increased *quality of life* because of using an IT-based homecare system.

– *History of vital signs values* - The client also mentioned that being able to compare the measured values with the values measured on the previous days was interesting because the client can see how her health condition was changing over time and could be aware of self health without needing to wait to talk with care-givers about it.

– *Calendar application* - The client liked the U-Care's calendar application, comparing with a paper-based calendar, where the client could see both personal events as well as appointments and vital signs and social activity services schedules, all in one place. The client also mentioned that she could choose to see everything per day, per week, etc.

– *Overall satisfaction* - When asked how much the client would pay for such a system, the client could not guess the monetary value of such a system, but indicated that it was a valuable system for quality-of-life. The client was interested to keep the U-Care system and use it, during the improvement period between the first and second series of experiments.

– *Other services* - When asked about what other services the client wished to have within the U-Care, the client was interested to learn and use the web. The client would like to have a search area in the U-Care application such that the client can find and read material on topics of interest.

We also observed that not all clients were satisfied; while Clients 1 and 6 were satisfied with the U-Care system, Client 4 (98 years old) found it difficult to use the system: "Too many steps (button pressings), not enough loud reminder sound to notify me, etc. I lived 98 years without using this types of services, I could survive the rest of my life without them". However, she liked using the Skype service to communicate with her family members (who filled in the contact lists), friends, and care-givers. Additionally, she liked reading E-books. In fact, she ignored reminders from the U-Care system because she wanted to continue reading an E-book.

8.3.2 Requested Changes

After the first series of the experiments, the care-givers requested a number of changes be implemented in the service tailoring platform. These changes were expected to be implemented before the second series of the experiments.

1. For the care services, they asked to add a feature that would give the possibility of scheduling the tasks to be executed at different times of a day. In the current prototype, they could add multiple times, but only by indicating time intervals between two events. For example, they could specify in the blood pressure service plan to create an event twice per day, starting from 9:00 in the morning and repeating it after 8 hours. However, they would like to specifically indicate the time of two or more (up to four) events per day.

2. For the weight monitoring task, the initial treatment pattern was to compare the measured value from today with that from yesterday, and if the difference exceeds a predefined threshold, the system should raise an alert. However, the care-givers mentioned that the measured value must be compared with a reference point (e.g., 85 Kg), which could be configured for each individual. They also preferred to have the tailoring interface in local language, which was Dutch in our experiment/validation environment.

3. As mentioned before, we captured the screen of the care-givers' laptop to monitor their behaviour while creating service plans. Based on the analysis of these screen shots, we identified another possible change (**not** based on the care-givers' feedback). We saw that most of the time, which is consumed by a care-giver to create a plan, was spent checking other created plans for that care-receiver. This can be eliminated by providing a list of already created service plans (for a care-receiver) and a summary of their parameters to a care-giver, while (s)he is creating a new service plan.

4. By analyzing the screen shots, we also saw that most of the time consumed by a care-giver to create a plan was writing the proper reminder and alert messages. Thus, we decided to use the messages that were written by the care-givers in the first experiment as the default values of the reminder and alert messages for the second series of the experiments. We expect that this could save time when creating plans. However, there could be some care-receivers who would be more motivated to use the services and the system if customized messages are delivered, and for these care-receivers the default messages are not suitable, hence, for those situations, there should be a pool of messages and each time one message could be selected to be sent.

5. For the social activity services, besides translating the interface to Dutch, the care-givers asked to add the possibility of creating new activity types by care-givers themselves (by choosing the *other* option in the activity type list and insert their own desired activity type).

The care-givers also requested some improvements regarding the user interface of the applications by the elderly. The improved version should have fewer buttons and options and require a minimum number of interactions with the care-receivers.

8.4 Second Series of the Experiments and the Results

Two months after the end of the first series of the experiments, the improved U-Care system based on the requested changes was validated in a second series of experiments. The same clients were supposed to participate in the same scenarios.

Unfortunately between the first and second series of the experiments, *Client 6*, who was very enthusiastic about using the system and appreciated it, had passed away. *Client 4* stopped using the system because of a lack of interest. Client 3 initially was willing to use the system for the second experiment, however, she lost her interest because of receiving bad news (she was diagnosed with cancer). For the second series of the experiments, Orbis introduced a new volunteer client (*Client 9*) to use the U-Care system. Thus, for the care services, only 3 clients in total participated in the following scenarios:

– *Client 1* participated in scenario *Manual MD + BP*.
– *Client 2* participated in scenario *BP + WT*.
– *Client 9* participated in scenario *BP + WT*.
– *Client 1* participated in scenario *Automatic MD*.

In the second series of the experiments, the automatic MD service was already integrated with the U-Care system. As in the case of the first series of the experiments, due to the limited number of sensors and smart phones, in the second series of the experiments the scenarios are also executed sequentially in three periods. As illustrated in Table *8-4*, in period 1, *Client 1*; in period 2, *Client 2* & *Client 9*; and finally in period 3, *Client 1* (for the second time in the second series of the experiments) used the care services. Before the start of these experiments, we informed the care-givers about the refinements made in different parts of the system based on their feedback.

Table 8-4 : Participation of the care-receivers in the second series of the experiments

	Clients		
Time Periods	**Client 1**	**Client 2**	**Client 9**
Period 1	Manual MD + BP	-	-
Period 2	-	BP + WT	BP + WT
Period 3	Automatic MD	-	-

8.4.1 Usability results

In the second series of the experiments, the care-givers created seven service plans in total for 3 clients and for three tasks: BP (3 plans), WT (2 plans), and MD (2 plans). The creation of service plans for automatic MD and manual MD have same number of steps and therefore, the care-givers saw no difference in creating their plans. This is one of the benefits of having a service tailoring platform and using the concept of SBBs for the different concrete services provided by various providers. This way, the care-givers did not have to learn different configuration options or different user interfaces of the different vendors, but only used the service tailoring interface that we proposed.

Out of the seven service plans, five were basically re-tailoring of the service plans previously created (for *Client 1* & *2*) during the first series of the experiments. Creating these five service plans was easy and each took less than half a minute to create. The care-givers only created 2 service plans for *Client 6* from scratch, of which the service plan for *BP* & *WT* were created in 2:38 and 2:43 minutes respectively.

Since only 2 service plans were created from scratch, measuring the *effectiveness* and the *efficiency* is not that meaningful. However, we could observe the *learnability* of the service tailoring platform and its user interface. The service creation time required by the care-givers in the second series of the experiments was compared to that in the first series of the experiments (see Fig. *8-8*). This comparison suggests that the learning curve is shorter, i.e., once trained, the care-givers not only could remember how to create the service plans for the second series of the experiments themselves and did so without expert assistance, but they also created the service plans faster.

After the completion of the second series of the experiments, we interviewed the care-givers to measure their *satisfaction*. In this final interview, we asked the care-givers to fill in a questionnaire. But unlike in the first series of the experiments, this time we used a different questionnaire which had only open-ended (descriptive) questions and had no close-ended (multiple choice) questions. The reason behind this was that after analyzing the results of the first series of experiments, we noticed that the open-ended questionnaires provide more information about the usability of the system than the close-ended questions. Asking these close-ended questions exhausted the care-givers before they could answer the second part of the questionnaire with open-ended questions. Thus, we design a new questionnaire to evaluate the usability of the U-Care system in general and the service tailoring platform in particular.

The new questionnaire is in the Appendix (see Figures *A-3* and *A-4*). This questionnaire has 15 questions. Q1-Q3 evaluate the functionality of the system, Q4-Q13 evaluate quality aspects, and Q14-Q15 ask about whether the care-giver is willing to use such a system in practice or not.

Another lesson learned from the interview during the first series of experiments was that most of the care-givers do not distinguish between the different platforms of the system (i.e., the service tailoring platform, the provisioning platform, or the application services), but rather they see the whole system as one unit. Thus, we decided not to ask them to fill in the questionnaire two times (for the whole system and for the service tailoring platform) and instead we evaluated which part of the system they are (or are not) satisfied with based on their answers.

Due to the unavailability of the care-givers, we could not talk with all the care-givers who used the U-Care system. Instead, we did the final interview with the four care-givers as follows:

– Two of them are *first responsible care-givers.* The one who created most of the service plans for the vital signs monitoring services and the other who created the service plans for the social activity services.

– The third care-giver is the *care coordinator* (“Zorgcoordinator in Dutch”), who did not create any service plan, but simply observed how the other nurses created the service plans and how the care-receivers used the system. This care-giver has more experiences of providing services to elderly people than other professional nurses, and coordinate other younger professional nurses.

– The last one was the human resource manager who was familiar with the U-Care project, but did not have direct contact with the care-receivers. It was useful to have him among interviewees, since he has more managerial and business perspectives than care perspective.

We asked them to fill in the questionnaires on a laptop computer instead of on paper. This way, the answers were readable and the care-givers also found it more convenient to type than to hand write. Filling in the questionnaires took almost an hour, and then after a short break, we discussed the questions and answers all together. During this discussion, we recorded our conversation. After the interview, we integrated and summarized the answers and sent the care-givers a copy for internal validation. The results of these interviews are listed below:

– **Usability of the Service Tailoring Platform**: The care-givers found the service tailoring platform as usable the second time as they did the first time. They could create service plans in about 3 minutes, and they did not feel a need for reducing this further.

  But the care-givers were not satisfied with the service tailoring of the social activity services, which wasted their time compared with providing manually scheduling in a paper format. The main reason is that the focus of the service tailoring was the care services, where during the requirement elicitation phase, its necessity was identified. But then later, in order to test the effect of social activities for the elderly, we made a service tailoring interface for social activities, which was related to scheduling different events and assigning care-receivers to those events (interested/not interested). We did not elicit the requirements of a service tailoring platform for such services but foolishly we used the same design as was used for the care services. This was not efficient, since, for the care services, we have service plans per care-receiver which unlike the case for social activity services where there a group of people could be interested in an activity.

– **Usability of UI of Applications for Elderly**: The care-givers still found the care-receiver interface not sufficiently usable for the elderly. They stated that there should be fewer options and buttons than the interface that we implemented, and they expressed a preference for a voice interface to the elderly. Not only the software, but also hardware should be designed specifically for elderly. Current Tablet-PCs technology was not considered convenient for elderly who are not familiar with concepts, such as scrolling, tapping, or sweeping, and it was difficult to teach them such concepts. The care-givers mentioned the automatic medicine dispenser (MD) as an example of a usable technology for the elderly in compare to the manual MD, where in order to get the medicine from the automatic MD, the care-receivers did not have to press any button on the Tablet-PC, but instead after receiving a reminder message, just pressed a big button on the dispenser and they could confirm receiving the medicine.

– **Saving Time**: The care-givers mentioned that if an IT-based system works correctly, indeed it could save their time, since they do not have to measure the vital signs of care-receivers or dispense their medication in person. However, if the elderly need the help of care-givers to operate the system, then systems like U-Care would create more work for care-givers rather than less work.

– **Quality of Care**: The care-givers believed an IT-based homecare system could increase and at the same time decrease the quality of care. It could increase the quality of care, when a care institution has too many clients or when the elderly live in their own home and receive care services at home. In those situations, using the IT-based systems could save care-givers' time and increase the quality of care by providing services 24/7. Some of the care-receivers found it nice to be aware of their own vital signs situations.

  On the other hand, for a care institution with a limited number of elderly, the quality of care could be decreased, because of *less attention* and *less physical contact* between care-givers and care-receivers. The physical contact is reassuring for elderly and reduces stress. Due to this less attention, the elderly may feel uncertain and keep measuring their vital signs repeatedly, when the values are too high/low. Furthermore, when there is a hazardous situation, the elderly do not get immediate help if there is a need. The care-givers believed that using a video communication service together with other care services could compensate for these negative effects of less attention and less physical contact. Interestingly, the video communication service was never mentioned by the care-givers during the requirement gathering phase of the project, and as a result of the experiments, they came to the conclusion that such a system is both necessary and helpful in the homecare domain.

– **Quality of Life**: The care-givers also believed that an IT-based system could increase as well as decrease the quality of life of the elderly. For some elderly, it gives *a sense of independence*, since they could measure their own vital signs without the help of care-givers. For other elderly persons, it has negative effects as the system restricts their behavior in the sense that they are afraid to leave their rooms because of the fear that they might not be able to measure their vital signs at the scheduled time. The care-givers indicated that the system is mobile (so mobility is an important factor) and the care-receivers could take the system with them for example, to the restaurants and measure their vital signs there. However, some elderly persons were scared that the devices fall and break (so another important factor is solidity and durability of the devices and using fewer devices, i.e., only a Tablet-PC or a smart phone and not the both).

– **Reliability of Homecare Systems**: The care-givers appreciated that the second version of the system was more reliable than the first version. Both the care-givers and care-receivers were stressed, whenever the system hangs and does not response. The distance between Orbis and the University of Twente caused some difficulties for the care-givers and made them sometimes upset when the system did not work properly. They believed that in practice, when an IT-based homecare system is used and there is a problem, immediate help should be provided by its service providers, otherwise the users will not trust the system anymore.

– **Target Group**: The care-givers believed an IT-based system is more useful in situations where the elderly still live in their own home rather than in a care institute. Moreover, some care-receivers, stopped using the system for the second series of the experiments, because of the fear of making mistakes when using the system and lack of familiarity. The care-givers indicated that they would prefer to test the system with the care-receivers in the range of 60-80 and for a longer period of time. This is a suggestion from the care-givers and future research should experiment the system with the elderly people in the age range of 60-80 years and for longer period to see if indeed these elderly people are better candidates for using such a system.

– **Integrated with their Current IT Systems**: Another interesting desire was that the system be integrated with their current IT system, such as their electronic patient file system, this would be more useful and have greater value than a separate stand alone system. This is because vital signs and other elderly data could be directly stored into the electronic file system without intervention from nurses, relieving them of some administrative tasks.

– **Other Possible Services**: Regarding other tasks which could be automated and provisioned using an IT-based system, the care-givers suggested the following: (a) a video communication service could be used together with other vital signs monitoring services, (b) temperature measurement, and (c) measuring blood sugars. The care-givers indicated that measuring blood sugar automatically would indeed save a lot of time of the care-givers.

8.5 Summary

In this chapter, we discussed a field test of our proposed service tailoring solution for IT-based homecare systems. The contribution of this chapter are:

– It describes the design and execution of the field test
– It presents the collected and analyzed data from the field test
– It reports on interesting results we obtained from the field test.

The field test was designed as two series of experiments to study the usability of the approach in terms of *effectiveness*, *efficiency*, *learnability*, and *satisfaction*. Because of the small size of the experiments (with a limited number of participants), the results that we obtained are primarily qualitative. However, we found them interesting, since they provided insights into the social and motivational mechanisms underlying the use of IT-based homecare services, which in turn could be used to further improve our approach.

The results of these experiments show that the service tailoring platform was usable for 4 care tasks and the care-givers found it useful and could save their time. However, this is very dependent on the usability of care-receiver's applications, which the care-givers should spend sometime training the care-receivers how to use the system. If the amount of time required to train the care-receivers to use the system is reduced, using the U-Care system generally could save the care-givers time and effort. The results also show that the whole U-Care system in general is useful and it could increase the quality of life of the elderly and increase the quality of care provided. The results in general provide a good motivation for using this type of technique to enable care-givers to create personalized care services and to provide the elderly with personalized care services. We did qualitative research via the real-world field test, but our conclusions are only propositions, thus quantitative research should be done to complemented this work to find empirical support for our results presented in Sections 8.3.1 and 8.4.1.

Chapter 9

Conclusions and Future Work

"Life isn't about finding yourself. Life is about creating yourself."
— George Bernard Shaw

This thesis focused on designing and developing a service tailoring platform to support personalized provisioning of IT-based homecare services that facilitate independent living of elderly people in their own home. We interviewed professional nurses (i.e., care-givers) in a care institute in the Netherlands and also employed a literature study of existing homecare systems during the requirements elicitation phase of our approach. Based on this findings of the requirements elicitation phase, we identified several common homecare tasks. To be able to provide the identified tasks, we designed supporting service building blocks (SBBs) and treatment patterns. By configuring the appropriate SBBs and using the correct treatment patterns, care-givers can create the required individual service plans for each specific care-receiver. The proposed service tailoring platform assists the care-givers in creating these service plans. A prototype implementation of the proposed platform has been developed and evaluated in a case study.

In this chapter, we discuss what we learned and then propose some challenges as potential future research topics. In Section 9.1, we summarize the contributions of the thesis by answering the research questions and discuss whether we achieved the research objectives. In Section 9.2, we discuss possible extensions to this work and identify possible future research, and finally, Section 9.3 summarizes the chapter.

9.1 Contributions

The approach proposed in this thesis had the following goals:

– Same or reduced costs of care provisioning to the elderly, which can be achieved by saving time of the care-givers.

– Better or same quality of care provided to the elderly, which can be achieved by personalizing the services.

– Improved or same quality of life experienced by the elderly, which can be achieved by supporting independent living of elderly.

Our aim is to make at least one of the above items better and none of them get worse. However, the results are subject to discussion. We believe, a hard criterion is costs: care provisioning should not become more expensive. But the other criteria might be violated to some extent (but preferably not) if the cost savings are substantial. Also the criteria are relative to the current situation. But the current situation will not last. The aging population may force changes: due to lack of personnel they will have less time to spend per care task per elderly person, and thus, the quality of care or quality of life will decreases, unless this can be prevented by new solutions, such as tailorable IT-based services as we proposed in this thesis.

In Section 1.4, we presented the main research questions to be addressed in this thesis. Now, we explain the contribution of the thesis by reflecting on the results and describe how we addressed these research questions.

– **RQ1**: *What are the common homecare tasks performed by care-givers and their corresponding IT-based homecare services?*
To answer this question, first, in Chapter 2, we performed a literature study to identify existing IT-based homecare systems and their challenges including the challenge of service tailoring.

In Chapter 3, we did an interview to identify common homecare tasks performed by care-givers in general and those tasks which can be automated in particular. We further identified several types of care-givers, which play different roles in the homecare domain. Among those, we selected the professional nurses to tailor the services, which became the target group for this thesis. Note they are only the targets of goal #1 Among the identified common tasks which could be automated by IT-based services, we selected four care monitoring tasks to implement and test in the case study: *blood pressure monitoring*, *medication intake monitoring*, *oxygen saturation monitoring*, and *weight monitoring*.

These health monitoring services are mainly demanded by the care-givers to monitor their patients. However, our field test showed that the care-receivers, by contrast prefer more *recreational* services such as: Skype, ebook, web browsing, and games. This shows that a successful IT-based homecare system should integrate recreational services with health monitoring services to motivate the elderly to utilize the system (gamification). Note that our case study describes a situation in which the elderly are institutionalized and live in a care institute. We speculate that the solution is generalizable to those elderly who would like to live independently in their own home under the surveillance of some health monitoring services.

– **RQ2**: *What is the (generic) service tailoring process?*
In Chapter 5, we introduced a service tailoring process to define the flow of actions, which should be followed by a care-giver to create a personalized homecare service. The service tailoring process starts by selecting a care-receiver and a homecare task and it eventually leads to the creation of a service plan. This service plan specifies the composition and configuration of the SBBs w.r.t. the requirements of an individual care-receiver as understood by a care-giver. The created service plan can be executed in a provisioning platform.

We have successfully employed the tailoring process in one case study with the four tasks mentioned earlier. The process is 'generic' enough to be used with minor or no adjustments in any homecare system comprising various care tasks. The field test provides some support for this claim but certainly it does not prove it.

– **RQ3**: *What are the components and entities, needed for service tailoring?*
In Chapter 5, we identified the required components for service tailoring, such as SBBs, their configuration parameters and treatment patterns. After that we defined a user profile to specify health related information as well as the preferences of the care-receivers in Chapter 6. Then, we described how the service tailoring platform could exploit this information (as stored in the user profiles). This guided the provisioning platform in choosing proper concrete services w.r.t. the abilities and preferences of the care-receivers.

Furthermore, in Chapter 7, we illustrated the implemented user interface and database structure that were used in the case study for creating, storing, and finally deploying a service plan by the care-givers.

- **RQ4**: *What can and cannot be automated in the service tailoring process?*
  The service tailoring process consists of three main steps, namely selection, composition, and configuration of SBBs. None of these steps is fully automated, nevertheless using some techniques we provide some level of automation for these steps. Using the treatment patterns we facilitate the selection and composition of the SBBs by abstracting away the technical details. Based on existing medical protocols, technicians together with the care-givers can define possible treatment patterns for each care task. The treatment patterns are defined to cover different scenarios by considering different types of users in various contexts. Then, during service plan creation, based on the selected homecare task, the service tailoring platform proposes appropriate treatment pattern(s) to the care-givers. There might still be more than one treatment pattern for a specific task; with each an alternative composition of services for that task. The advantage of this approach is twofold. On one hand, it helps the care-giver choose the appropriate treatment pattern and, on the other hand, by covering numerous compositions it decreases the need for re-composition of services.

  Moreover, to support some level of automation in the configuration step, we proposed the concept of a user profile. To specify the values for some of the configuration parameters of SBBs (e.g., the modality of reminder service), the service tailoring platform uses the information in the user profiles of the care-receivers. Care-givers can further refine the values based on their comprehensive knowledge about the care-receiver's situation.

- **RQ5**: *What are the risks of using our approach?*
  In Chapter 4, we presented some risks that exist in the current situation of the homecare domain without using an IT-based system by interviewing the care-givers participating in the case study. Then, we introduced a Risk Driven Requirements Specification (RiDeRS) method to identify those risks which are added to the list of risks, as a result of employing an IT-based homecare system.

RiDeRS can also be used to specify additional requirements of the system to mitigate or prevent these risks. We introduced RiDeRS in the context of an IT-based homecare system, nevertheless it is proposed as a generic method for identifying the risks of any IT-based critical system.

Later, we observed the U-Care system behavior in a field test of the case study to see whether we identified the list of possible risks correctly and whether their identified countermeasures indeed prevented the anticipated risks. Our observation in this field test showed that there was another type of risk regarding usability of the system. Lack of usability (of the user interface) for the care-receivers would defeat the time-saving purpose of the system for the care-givers, because the care-receivers frequently asked for support from the care-givers to use the homecare system. This would create more work for the care-givers, rather than less. This is a risk and needs to be added to the homecare risk analysis methods. Before running the field test, we hypothesized that:

(1) If all assumptions (introduced in Section 4.3.1) are satisfied and the artifact (the system) is implemented correctly, then the goals (homecare with improved availability) would be satisfied.

This claim was falsified in the current field test. The results showed that the goals will not be reached unless the system is usable by the care-receivers. So we give a more prudent revised claim:

(1') If all assumptions are satisfied, then if the artifact is implemented correctly and usable by the care-receivers, then the goals are satisfied.

Usability is a requirement of the artifact, not an assumption about the environment. But this requirement does make an assumption about the environment, so this assumption must be added too, given an even more prudent claim:

(1") If all assumptions are satisfied, the care-receivers are elderly people in a care home environment, the artifact is implemented correctly, and the artifact usable by the care-receivers, then the goals are satisfied.

- **RQ6**: *Is the tailoring platform usable and useful?*
  To evaluate the usability and usefulness of the service tailoring platform, we implemented a prototype of the proposed approach. In Chapter 8, we reported on two series of experiments which were performed in the case study and presented their results. The results showed that the care-givers can create the desired service plans using the user interface of the service tailoring platform in a reasonably short time and with little effort.

  The results also indicate that the IT-system, in general, can improve the quality of care and increase the quality of life of care-receivers. Care-givers and care-receivers both perceived the system as useful.

9.2 Future Research

In this thesis, we have answered some research questions. As noted by John Archibald Wheeler: "We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance". This is also true with our work, in which answering those research questions leads to more lines of research which should be pursued. In the following, we discuss five topics for future research that could provide the next steps along the path to a practical and widely applicable set of tailorable IT-based homecare services:

- **Other Case Studies**
  We experimented with only one case study in the Netherlands in which the elderly persons were living in a nursing house. Applying the approach in other situations or cultures may have different results. For example, the approach should be tested in a situation where elderly live in their own home and receive support from care-givers remotely or from family members. Elderly move to a nursing house when they cannot live independently anymore and take care of themselves. Elderly who live at home, might appreciate IT-based services and be more motivated in using them as opposed to elderly living in a nursing home. This might lead to different types of requirements and tasks than we have found.

  Moreover, the approach should be evaluated using other tasks such as, monitoring sugar level and doing physical training. The service tailoring platform and process should be tested with those tasks in order to determine whether the platform and process are still usable.

Furthermore, people from different countries possess various cultures. Each culture has its own perspectives of the world. Thus, the same system could have different effects in different countries and thus lead to different requirements.

- **Other Types of Context**
  In this work, we had a chance to test the system with two types of context information namely: *time* and *location*. We have also designed the service tailoring platform based on only these two types of context information. We could consider other types of context information such as *people nearby* and *care-receivers' current activity*. There are some systems on the market such as wrist sensors which detect the user activities such as walking, running or falling. Detecting nearby people is very difficult. Nearby people could be detected by tracking their mobile phone, however, it is possible that the intended user does not carry his phone or carries somebody else's phone. Adding these types of context information could raise new options and possibilities in tailoring the services and could affect the composition and configuration of SBBs. This can also affect the user interface and the way we express the configuration parameters. Currently, we represent the context information and their tailoring possibilities in terms of IF-THEN-ELSE constructs, however, this might not be efficient when we have several context types.

- **One Service Plan for all Tasks**
  We made a design choice and created a separate service plan for each homecare task. Thus, a created service plan is per task and per care-receiver, for example, a service plan for *Client 1* and the *blood pressure monitoring* task. Having a service plan per task simplifies definition of the treatment patterns (the composition of services to support each task) and makes the patterns reusable for other cases and care-receivers. However, a disadvantage of this approach is that the care-givers may have to create multiple service plans for the same care-receiver, and while doing so they should be aware of what has been defined in other service plans as the tasks may have dependencies. This disadvantage was shown in the case study, where a care-giver needed extra time to create a new service plan after a previous service plan for the same care-receiver had already been defined. We solved this problem by providing some information to a care-giver about previously created plans for a specific care-receiver while she creates a new service plan for the same care-receiver, such that interdependencies can be tackled during the plan creation process (in Section 8.3.2).

Another possibility, which was not implemented, is that a service plan covers all homecare tasks of a single client (especially, since elderly clients tend to have many comorbidities). This way, a care-giver can select multiple tasks from a list of tasks to create one service plan for a care-receiver, allowing better handling of interdependencies and having a complete picture of a single care-receiver's care services.

We speculate that the time required to create the service plans one by one for each task or one service plan to cover all tasks will not differ that much. However, supporting the creation of a single service plan for a care-receiver, covering needs regarding multiple care tasks, will *undoubtable* increase the complexity of the tailoring platform. But most of this complexity is 'under the hood', and the care-giver may profit from a better trade-off between increased complexity (of the user interface) and increased functionality. Whether this is indeed the case is the subject of further research.

– **Interdependencies between Care Tasks**
All service plans that is created for one care-receiver to support his care tasks should be in harmony with each other. Tasks might have *pre* and *post* conditions that cause conflicts. For example, according to care task #1 a care-receiver has to consume *erythromycin* and according to task #2 this care-receiver is recommended to eat *grapefruit* because of his high blood pressure. According to medical protocols someone who has taken erythromycin should not eat grapefruit to avoid undesirable consequences. Therefore, for this care-receiver, the tailoring system should be aware of this medical conflict and warn the care-giver when creating a service plan for the second task.

This conflict, and pre and post condition might be of various nature. We enumerate here three types of these conflicts namely *time*, *location* and *medical*. Regardless of the order and the architecture of the service plan (either creating the service plans one by one for each task or one service plan to cover all tasks), the tailoring platform should be aware of these conditions and conflicts and warn the care-giver when creating the service plans. Nevertheless, the final decision is made by the care-giver to either the composition and/or parameter settings of the service plans to support a task are changed (for example, the time that a medicine is given is changed, or the type of medicine is changed, to avoid conflicts) to design a conflict free service plan, or simply ignore the warnings.

- **Tailorability of Tailoring Interface**
  We provide a static user interface for tailoring the services. A dynamic user interface (e.g., adding/removing some steps and their contents) could be provided depending on who is tailoring a service (i.e., which care-giver) and/or for whom the service is created (i.e., which care-receiver). The dynamicity can be provided based on the history of the care-givers' behaviour in using the tailoring interface or the care-receiver's profile. This would probably increase the usability of the user interface.

- **Modification of the Treatment Patterns**
  To facilitate service tailoring, we proposed the use of treatment patterns, each presenting an alternative composition of services for a task. There might be situations where the defined treatment patterns are not sufficient and a care institute wants to modify them. Thus, an additional tool and user interface could be developed to enable care-givers (this could be authorized care-givers such as physicians and not normal nurses) to modify treatment patterns themselves. The tool can check correctness properties of the modified patterns, such as absence of deadlock, before deploying them. This extension should be followed by an evaluation of its benefits and consequences.

- **Automatically Creating SBBs from Existing Services**
  Third parties may be interested in testing how their applications can be used together with other applications. Such a composite application (or service) could be defined in the service tailoring platform, and provided in the provisioning platform. The usefulness of the composite application can be evaluated by the care-givers during service tailoring and provisioning. This is currently not possible, since no support is available for automatically creating an abstraction of an application service that is registered with the provisioning platform, and making this abstraction into an SBB visible in the service tailoring platform.

  A new service provided by a third party can be used to create a service building block. The appropriate service building block of such provided service can be generated by a model driven technique/tool. Though due to the nature of homecare domain which is a critical domain, in case of using such tools and techniques, the final result should be double checked to be sure that the created service building block is error free.

9.3 Summary

Without a doubt, in the next ten to twenty years, several IT-based homecare services will be utilized in the practice of homecare. Uptake will be easier, especially since the next generation of elderly persons will be more familiar with IT-based devices such as smart phones or Tablet-PCs. The work in this thesis could be a small step towards *service tailoring of homecare services* with a general purpose of improving the quality of homecare services. In summary, we have designed and developed a service tailoring platform which allows domain experts (e.g., care-givers) to create personalized services to fulfil individual needs of care-receivers. The platform has been used and evaluated in a care institute in the Netherlands to create service plans for four care tasks similar to real world situations. The results show that use of such a platform in combination with different homecare application services could both improve the quality of the care and life of elderly.

References

[1] ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability. Technical report, International Organization for Standardization, Geneva, 1998.

[2] M. Aiello, T. Catarci, and M. Mecella. Smart homes infrastructures and interactions (shii 2009). In *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on*, pages 234 –235, 2009.

[3] Ashiful Alam, Mohammad Zarifi Eslami, and Klaas Sikkel. Elderly and homecare tasks: A literature review on problems. In *4th IADIS International Conference e-Health*, pages 211–216, 2012.

[4] C. Alexander. *A Pattern Language: Towns, Buildings, Construction*. Center for Environmental Structure Series. Oxford University Press, USA, 1977.

[5] Majd Alwan and Jeremy Nobel. State of Technology in Aging Services: Summary. Technical report, Center for Aging Services Technologies, www.agingtech.org, March 2008.

[6] Amigo. Ambient intelligence for the networked home environment. Available at: http://www.hitech-projects.com/euprojects/amigo/, last visited: January 2013.

[7] Ajit Appari and M. Eric Johnson. Information security and privacy in healthcare: current state of research. *International Journal of Internet and Enterprise Management*, pages 279–314, 2010.

[8] Yudistira Asnar, Paolo Giorgini, and John Mylopoulos. Goal-driven risk assessment in requirements engineering. *Requirements Engineering*, 16:101–116, June 2011.

[9] C. Atkinson and T. Kuhne. Model-driven development: a metamodeling foundation. *Software, IEEE*, 20(5):36–41, 2003.

[10] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, 2004.

[11] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A Survey on Context-aware Systems. *IJAHUC*, 2(4):263–277, 2007.

[12] Wolf-Tilo Balke and Matthias Wagner. Towards Personalized Selection of Web Services. In *the International World Wide Web Conference*, 2003.

[13] Montserrat Batet, David Isern, Lucas Marin, Sergio Martinez, Antonio Moreno, David Sanchez, Aida Valls, and Karina Gibert. Knowledge-driven delivery of home care services. *Journal of Intelligent Information Systems*, pages 1–36, 2010.

[14] Jorg Becker, Patrick Delfmann, Alexander Dreiling, Ralf Knackstedt, and Dominik Kuropka. Configurative Process Modeling - Outlining an Approach to Increased Business Process Model Usability. In *Information Resources Management Association Conference*, pages 615–619, 2004. New Orleans.

[15] George W Beeler. HL7 Version 3–An object-oriented methodology for collaborative standards development. *Intl. J. of Medical Informatics*, 48:151–161, 1998.

[16] Henry Been and Mohammad Zarifi Eslami. A Survey on Tasks Performed in Eldercare. In *4th IADIS International Conference e-Health*, pages 176–181, 2012.

[17] N. Bevan, J. Kirakowski, and J Maissel. What is Usability? In *Human Aspects in Computing: Design and Use of Interactive Systems with Terminals*, pages 651–655. Elsevier, 1991.

[18] Jerome Boyer and Hafedh Mili. IBM WebSphere ILOG JRules. In *Agile Business Rule Development*, pages 215–242. Springer Berlin Heidelberg, 2011.

[19] Bundesamt für Sicherheit in der Informations technik. BSI-Standard 100-1: Managementsysteme für Informationssicherheit (ISMS). Version 1.5, Available at: https://www.bsi.bund.de/cae/servlet/contentblob/471450/publicationFile/30749/standard_1001_pdf.pdf , last visited: February 22 2012, 2008.

[20] Bundesamt für Sicherheit in der Informations technik. BSI-Standard 100-2: IT-Grundschutz-Vorgehensweise. Version 2.0, Available at: https://www.bsi.bund.de/cae/servlet/contentblob/471452/ publicationFile/ 30758/standard_1002_pdf.pdf , last visited: February 22 2012, 2008.

[21] Bundesamt für Sicherheit in der Informations technik. BSI-Standard 100-3: Risikoanalyse auf der Basis von IT-Grundschutz. Version 2.5, Available at: https://www.bsi.bund.de/cae/servlet/contentblob/471454/ publicationFile/ 30747/standard_1003.pdf , last visited: February 22 2012, 2008.

[22] Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards requirements-driven information systems engineering: the Tropos project. *Inf. Syst.*, 27:365–389, September 2002.

[23] Anis Charfi and Mira Mezini. Hybrid web service composition: business processes meet business rules. In *2nd Intl. Conf. on Service oriented computing*, pages 30–38. ACM, 2004.

[24] Luca Chittaro, Roberto Ranon, Luca De Marco, and Augusto Senerchia. User Modeling of Disabled Persons for Generating Instructions to Medical First Responders. In *User Modeling, Adaptation, and Personalization*, volume 5535 of *LNCS*, pages 435–440. Springer Berlin / Heidelberg, 2009.

[25] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Acadamic Publishers, 2000.

[26] Lawrence Chung and Julio Cesar Sampaio do Prado Leite. Conceptual modeling: Foundations and applications. chapter On Non-Functional Requirements in Software Engineering, pages 363–379. Springer-Verlag, 2009.

[27] Lawrence Chung and Sam Supakkul. Capturing and reusing functional and non-functional requirements knowledge: A goal-object pattern approach. In *IRI*, pages 539–544, 2006.

[28] CLEAR. The Clinical Leading Environment for the Assessment of Rehabilitation protocols in home care, 2007. Available at: http://www.habiliseurope.eu/?q=node/5, last visited: January 2013.

[29] Jane Cleland-Huang, Raffaella Settimi, Xuchang Zou, and Peter Solc. Automated classification of non-functional requirements. *Requirements Engineering*, 12:103–120, 2007.

[30] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. (2009a), Version 3.1, Revision 3, Final, Available at: http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf, last visited: February 20 2012, July 2009.

[31] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components. (2009b), Version 3.1, Revision 3, Final, Available at: http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf, last visited: February 20 2012, July 2009.

[32] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components. (2009b), Version 3.1, Revision 3, Final, Available at: http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf, last visited: February 20 2012, July 2009.

[33] Maria F. Costabile, Daniela Fogli, Piero Mussio, and Antonio Piccinno. End-User Development: The Software Shaping Workshop Approach. In *End User Development*, volume 9, chapter 9, pages 183–205. Kluwer Academic Publishers, 2006.

[34] Francois de Brantes, Alice G. Gosfield, Elizabeth McGlynn, Meredith Rosenthal, and Jeffrey Levin-Scherz. Clinical Practice Guidelines for Older Patients With Comorbid Diseases. *The Journal of the American Medical Association*, 295(1):34, 2006.

[35] Paul De Clercq, Katharina Kaiser, and Arie Hasman. Computer-Interpretable Guideline formalisms. *Studies in health technology and informatics*, 139:22–43, 2008.

[36] Aubrey D. De Grey. Life Span Extension Research and Public Debate: Societal Considerations. *Studies in Ethics, Law, and Technology*, 1(1), 2007.

[37] Anind K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[38] Elisabetta Di Nitto, Carlo Ghezzi, Andreas Metzger, Mike Papazoglou, and Klaus Pohl. A journey to highly dynamic, self-adaptive service-based applications. *Automated Software Engineering*, 15:313–341, 2008.

[39] P. Dockhorn Costa, L. Ferreira Pires, and M.J. van Sinderen. Architectural Patterns for Context-Aware Services Platforms. In *Second International Workshop on Ubiquitous Computing*, pages 3–18. INSTICC Press, 2005.

[40] Dreaming. elDeRly-friEndly Alarm handling and MonitorING. Available at: http://www.dreaming-project.org/, last visited: January 2013.

[41] J.H. Duffus, S.S. Brown, and N.A.G.G. Fernicola. Glossary for chemists of terms used in toxicology. *International Union of Pure and Applied Chemistry*, 65:2003–2122, 1993.

[42] Oladimeji Ebenezer A., Supakkul Sam, and Chung Lawrence. Security Threat Modeling and Analysis: A Goal-Oriented Approach. In *Proc. of the 10th IASTED International Conference on Software Engineering and Applications*, 2006.

[43] Golnaz Elahi and Eric Yu. Modeling and analysis of security trade-offs - A goal oriented approach. *Data Knowledge Engineering*, 68:579–598, July 2009.

[44] Golnaz Elahi, Eric Yu, and Nicola Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requirement Engineering*, 15(1):41–62, March 2010.

[45] epSOS. Smart Open Services for European Patients, 2008. Available at: http://www.epsos.eu/, last visited: January 2013.

[46] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

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

[48] European Telecommunications Standards Institute: Human Factors (HF). Personalization and User Profile Management, User Profile Preferences and Information. ETSI, ES 202 746, V1.1.1, 2010.

[49] M. J. Field and K. H. Lohr. Clinical practice guidelines: Directions for a new program. Institute of Medicine report, Washington, DC: National Academy Press, 1990.

[50] Donald Firesmith. Analyzing and Specifying Reusable Security Requirements. In *Proc. Solid Freeform Fabrication Sym*, pages 507–514, 2003.

[51] Donald Firesmith. Specifying Reusable Security Requirements. *Journal of Object Technology*, 3(1):61–75, 2004.

[52] Donald Firesmith. Engineering safety-related requirements for software-intensive systems. In *Proc. of 27th International Conference on Software Engineering*, pages 720 – 721, May 2005.

[53] Juan Fraile, Javier Bajo, Ajith Abraham, and Juan Corchado. HoCaMA: Home Care Hybrid Multiagent Architecture. In *Pervasive Computing*, Computer Communications and Networks, pages 259–285. Springer London, 2010.

[54] Douglas Fridsma and Jan Thomsen. Representing Medical Protocols for Organizational Simulation: An Information-Processing Approach. *Computational and Mathematical Organization Theory*, 4:71–95, 1998.

[55] Katrin Gaßner and Michael Conrad. ICT enabled independent living for elderly, A status-quo analysis on products and the research landscape in the field of Ambient Assisted Living in EU-27. prepared by VDI/VDE Innovation und Technik GmbH, March 2010. Available at: http://www.ifap.ru/library/book467.pdf, last visited: January 2013.

[56] Pallapa Gautham and Das Sajal. Challenges of Designing Privacy Enhanced Context-Aware Middleware for Assisted Healthcare. In *Smart Homes and Health Telematics*, volume 5120 of *LNCS*, pages 200–207. Springer Berlin / Heidelberg, 2008.

[57] Andreas Girgensohn, Beatrix Zimmermann, Alison Lee, Bart Burns, and Michael E. Atwood. Dynamic Forms: An Enhanced Interaction Abstraction Based on Forms. In *IFIP TC13 Interantional Conference on Human-Computer InteractionHuman-Computer Interaction, INTERACT*, pages 362–367. Chapman & Hall, 1995.

[58] V. Gomoi and V. Stoicu-Tivadar. A new method in automatic generation of medical protocols using artificial intelligence tools and a data manager. In *Intl. Computational Cybernetics and Technical Informatics*, pages 243–246, 2010.

[59] Eduardo Manuel Goncalves da Silva. *User-centric service composition - towards personalised service composition and delivery*. PhD thesis, Enschede, the Netherlands, 2011.

[60] Colin Gordon and Jens Pihlkjaer Christensen. Health Telematics for Clinical Guidelines and Protocols. IOS Press, 1995.

[61] Florian Gottschalk, Wil M.P. van der Aalst, Monique H. Jansen-Vullers, and Marcello La Rosa. Configurable workflow models. *International Journal of Cooperative Information Systems (IJCIS)*, 17(2):177–221, 2008.

[62] Maja Hadzic and Elizabeth Chang. Ontology-Based Support for Human Disease Study. *Hawaii International Conference on System Sciences*, 6:143a, 2005.

[63] Raf Haesen, Monique Snoeck, Wilfried Lemahieu, and Stephan Poelmans. On the Definition of Service Granularity and Its Architectural Impact. In *Advanced Information Systems Engineering*, volume 5074 of *LNCS*, pages 375–389. Springer Berlin / Heidelberg, 2008.

[64] Austin Henderson and Morten Kyng. Design at work. chapter There's no place like home: continuing design in use, pages 219–240. L. Erlbaum Associates Inc., 1992.

[65] Andrea Herrmann and Ayse Morali. Riskrep: Risk-based security requirements elicitation and prioritization (extended version), 2010.

[66] Andrea Herrmann and Barbara Paech. MOQARE: Misuse-oriented Quality Requirements Engineering. *Requirements Engineering*, 13:73–86, 2008.

[67] Jongyi Hong, Euiho Suh, and Sung-Jin Kim. Context-aware Systems: A Literature Review and Classification. *Expert Syst. Appl.*, 36(4):8509–8522, 2009.

[68] Siv Hilde Houmb, Shareeful Islam, Eric Knauss, Jan Jürjens, and Kurt Schneider. Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements Engineering - Special Issue on RE'09: Security Requirements Engineering*, 15:63–93, 2010.

[69] Volker Hoyer, Katarina Stanoevska-Slabeva, Till Janner, and Christoph Schroth. Enterprise Mashups: Design Principles towards the Long Tail of User Needs. In *International Conference on Services Computing (SCC)*, pages 601–602. IEEE Computer Society, 2008.

[70] IBM. DB2 database software. Available at: http://www-01.ibm.com/software/data/db2/, last visited: March 2013.

[71] IBM. WebSphere ILOG JRules. Available at: http://publib.boulder.ibm.com/infocenter/brjrules/v7r1/index.jsp, last visited: March 2013.

[72] IBM. WebSphere Lombardi Edition. Available at: http://www-01.ibm.com/software/integration/lombardi-edition/, last visited: March 2013.

[73] David Isern, Antonio Moreno, David Sánchez, Ákos Hajnal, Gianfranco Pedone, and László Zsolt Varga. Agent-based execution of personalised home care treatments. *Appl. Intell.*, 34(2):155–180, 2011.

[74] Shareeful Islam. Software development risk management model: a goal driven approach. In *Proc. of the doctoral symposium for ESEC/FSE on Doctoral symposium*, ESEC/FSE Doctoral Symposium '09, pages 5–8, 2009.

[75] Shareeful Islam and Siv Hilde Houmb. Integrating risk management activities into requirements engineering. In *4th International Conference on Research Challenges in Information Science*, pages 299 –310, may 2010.

[76] ISO. Risk management – principles and guidelines. International Standard 31000, 2009.

[77] ISO/IEC. *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.

[78] ISO/IEC. Information technology – security techniques – guidelines for the management of IT security – part 1: Concepts and models for it security. International Standard 13335-1, 2004.

[79] Dörr J, Punter T, Bayer J, Kerkow D, Kolb R, Koenig T, Olsson T, and Trendowicz A. Quality Models for Non-functional Requirements. IESE-Report Nr. 010.04/E, 2004.

[80] M.A. Jackson. *Problem Frames: Analysing and Structuring Software Development Problems*. Addison-Wesley, 2000.

[81] Jan Jürjens. Towards Development of Secure Systems Using UMLsec. In *Proc. of the 4th International Conference on Fundamental Approaches to Software Engineering*, FASE '01, pages 187–200, 2001.

[82] Jan Jürjens. Developing Safety-Critical Systems with UML. In *UML 2003 - The Unified Modeling Language. Modeling Languages and Applications*, Lecture Notes in Computer Science, pages 360–372. Springer Berlin / Heidelberg, 2003.

[83] Jan Jürjens. *Secure systems development with UML*. Springer, 2005.

[84] Peter J. Kammer, Gregory Alan Bolcer, Richard N. Taylor, Arthur S. Hitomi, and Mark Bergman. Techniques for Supporting Dynamic and Adaptive Workflow. *Computer Supported Cooperative Work*, 9:269–292, 2000.

[85] Athanasios Karapantelakis and Gerald Q. Maguire. Utilizing Social Context for Providing Personalized Services to Mobile Users. In *Proceedings of the 5th European Conference on Smart Sensing and Context (EuroSSC)*, volume 6446 of *Lecture Notes in Computer Science*, pages 28–41. Springer, 2010.

[86] Alan Kay. Microelectronics and the personal computer. *Scientific American*, pages 231–244, 1977.

[87] Rick Kazman, Mark Klein, and Paul Clements. ATAM: Method for Architecture Evaluation. CMU/SEI-2000-TR-004, Software Eng. Inst., Carnegie Mellon University, 2000.

[88] Rick Kazman, Mark H. Klein, Mario Barbacci, Thomas A. Longstaff, Howard F. Lipson, and S. Jeromy Carrière. The Architecture Tradeoff Analysis Method. In *ICECCS*, pages 68–78, 1998.

[89] Werner KieBling. Foundations of preferences in database systems. In *the 28th international conference on Very Large Data Bases*, VLDB '02, pages 311–322. VLDB Endowment, 2002.

[90] Eun-Jung Ko, Hyung-Jik Lee, and Jeun-Woo Lee. Ontology and CDSS based Intelligent Health Data Management in HealthCare Server. In *World Academy of Science, Engineering and Technology*, 2007.

[91] Dimitri Konstantas, Richard Bults, Aart van Halteren, Katarzyna Wac, Val Jones, Ing Widya, Rainer Herzog, and Barbara Streimelweger. Mobile health care: Towards a commercialization of research results. In *European Conference on eHealth (ECEH)*, volume 91 of *LNI*, pages 155–166. GI, 2006.

[92] I. Korhonen, J. Parkka, and M. Van Gils. Health monitoring in the home of the future. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3):66 – 73, May-June 2003.

[93] Akhil Kumar and Wen Yao. Process Materialization Using Templates and Rules to Design Flexible Process Models. In *Rule*

Interchange and Applications, volume 5858 of *LNCS*, pages 122–136. Springer Berlin / Heidelberg, 2009.

[94] Ariella Lang, Nancy Edwards, and Andrea Fleiszer. Safety in home care: a broadened perspective of patient safety. *Intl. J. for Quality in Health Care*, 20:130–135, 2008.

[95] Seon-Woo Lee, Yong-Joong Kim, Gi-Sup Lee, Byung-Ok Cho, and Nam-Ha Lee. A remote behavioral monitoring system for elders living alone. In *International Conference on Control, Automation and Systems, ICCAS '07*, pages 2725 –2730, 2007.

[96] N. Leveson. *Engineering a safer world: Systems thinking applied to safety*. MIT Press, 2012.

[97] Nancy G. Leveson. *Engineering a Safer World: Systems Thinking Applied to Safety*, page 4. To be published by MIT Press in Fall, 2011.

[98] James R. Lewis. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[99] Henry Lieberman, Fabio Paterno, Markus Klann, and Volker Wulf. End-User Development: An Emerging Paradigm. In *End User Development*, volume 9 of *Human-Computer Interaction Series*, pages 1–8. Springer Netherlands, 2006.

[100] L. Lin, BA. Nuseibeh, DC Ince, M. Jackson, and JD. Moffett. Analysing Security Threats and Vulnerabilities Using Abuse Frames. Technical Report No: 2003/ 10, Department of Computing, The Open University, United Kingdom, 2003.

[101] Lin Liu, Eric Yu, and John Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *Proc. of the 11th IEEE International Conference on Requirements Engineering*, pages 151–161, 2003.

[102] Xuanzhe Liu, Gang Huang, and Hong Mei. A User-Oriented Approach to Automated Service Composition. In *IEEE International Conference on Web Services (ICWS)*, pages 773 – 776, 2008.

[103] Xuanzhe Liu, Yi Hui, Wei Sun, and Haiqi Liang. Towards Service Composition Based on Mashup. In *IEEE Congress on Services*, pages 332 –339, 2007.

[104] Allan MacLean, Kathleen Carter, Lennart Lövstrand, and Thomas Moran. User-tailorable systems: pressing the issues with buttons. In *SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 175–182. ACM, 1990.

[105] Norbert Malanowski, Rukiye Ozcivelek, and Marcelino Cabrera. Active Ageing and Independent Living Services, The Role of Information and Communication Technology. European Communitiy, 2008. Available at: http://www.umic.pt/images/stories/publicacoes2/ JRC41496.pdf, last visited: January 2013.

[106] Jawahar Malhotra. Dynamic Extensibility in a Statically-Compiled Object-Oriented Language. In *First JSSST International Symposium on Object Technologies for Advanced Software*, pages 297–314. Springer-Verlag, 1993.

[107] MATCH. Mobilising Advanced Technologies for Care at Home, 2005-2012. Available at: http://www.match-project.org.uk/main/main.html, last visited: January 2013.

[108] Tony McBryan, Marilyn R. McGee-Lennon, and Phil Gray. An integrated approach to supporting interaction evolution in home care systems. In *1st Intl. Conf. on PErvasive Technologies Related to Assistive Environments*, pages 1–8. ACM, 2008.

[109] John McDermott and Chris Fox. Using Abuse Case Models for Security Requirements Analysis. In *Proc. of the 15th Annual Computer Security Applications Conference*, ACSAC '99, pages 55–65, 1999.

[110] Marilyn Rose McGee-Lennon. Requirements engineering for home care technology. In *twenty-sixth annual SIGCHI Conf. on Human factors in computing systems*, CHI '08, pages 1439–1442. ACM, 2008.

[111] Lucas Meertens, Maria Iacob, and Lambert Nieuwenhuis. A method for business model development. In *Business Modeling and Software Design*, volume 109 of *Lecture Notes in Business Information Processing*, pages 113–129. Springer Berlin Heidelberg, 2012.

[112] Ministers, local government, NHS, social care, professional and regulatory organisations. Putting people first: a shared vision and commitment to the transformation of adult social care. Technical report, NHS Direct, December 2007.

[113] Andrew P. Moore, Robert J. Ellison, and Richard C. Linger. Attack modeling for information security and survivability, 2001.

[114] Anders Mørch. Designing for radical tailorability: coupling artifact and rationale. *Knowledge-Based Systems*, 7(4):253 – 264, 1994.

[115] Anders Mørch. Application Units: Basic Building Blocks of Tailorable Applications. In *EWHCI'95*, pages 45–62, 1995.

[116] Anders Mørch. Computers and design in context. chapter Three levels of end-user tailoring: customization, integration, and extension, pages 51–76. MIT Press, Cambridge, MA, USA, 1997.

[117] Anders Mørch, Oliver Stiemerling, and Volker Wulf. Tailorable groupware: issues, methods, and architectures report of a workshop held at GROUP'97, Phoenix, AZ, Nov. 16th, 1997. *SIGGROUP Bull.*, 19(1):4–7, April 1998.

[118] Anders I. Mørch, Gunnar Stevens, Markus Won, Markus Klann, Yvonne Dittrich, and Volker Wulf. Component-based technologies for end-user development. *Commun. ACM*, 47(9):59–62, 2004.

[119] Haralambos Mouratidis, Paolo Giorgini, and Gordon Manson. Integrating security and systems engineering: Towards the modelling of secure information systems. In Johann Eder and Michele Missikoff, editors, *Proc. of the 15th Conference On Advanced Information Systems Engineering*, volume 2681 of *Lecture Notes in Computer Science*, pages 63–78, 2003.

[120] MPOWER. Middleware Platform for eMPOWERing cognitive disabled and elderly, 2006-2009. Available at: http://www.sintef.no/Projectweb/MPOWER, last visited: January 2013.

[121] Ingrid Mulder, Yvonne Schikhof, Martijn Vastenburg, Alan Card, Tory Dunn, Andreas Komninos, Marilyn McGee-Lennon, Mark Santcroos, Gabriele Tiotto, Mieke van Gils, Jan-Willem van 't Klooster, Annelies Veys, and Mohammed Zarifi Eslami. Designing with Care: The Future of Pervasive Healthcare. *IEEE Pervasive Computing*, 8(4):85–88, 2009.

[122] Claudia Müller, Cornelius Neufeldt, David Randall, and Volker Wulf. ICT-development in residential care settings: sensitizing design to the life circumstances of the residents of a care home. In *Proceedings of the 2012 ACM annual conference on Human*

Factors in Computing Systems, CHI '12, pages 2639–2648. ACM, 2012.

[123] National Institute of Standards and Technology. Common Industry Specification for Usability - Requirements, June 2007. Available at: http://zing.ncsl.nist.gov/iusr/documents/CISU-R-IR7432.pdf, last visited: January 2013.

[124] Object Management Group. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms. Available at: http://www.inf.ufrgs.br/ juliog/profiles/06-05-02.pdf, last visited: February 27 2012, 2004.

[125] Object Management Group. Business Process Modeling Notation (BPMN) Version 1.0. OMG Final Adopted Specification, 2006. Available at: http://www.bpmn.org, last visited: May 2011.

[126] B. Paech and D. Kerkow. Non-functional requirements engineering - quality is essential. *10th international workshop on requirements engineering: foundation of software quality, REFSQ 04, Essener Informatik Beitrage Bd 9, Essen*, pages 237–250, 2004.

[127] M. P. Papazoglou and D. Georgakopoulos. Introduction. *Commun. ACM*, 46:24–28, October 2003.

[128] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-Oriented Computing: State of the Art and Research Challenges. *Computer*, 40(11):38–45, 2007.

[129] Praxiom Research Group limited. ISO 27001 AND ISO 27002, PLAIN ENGLISH DEFINITIONS. Available at: http://www.praxiom.com/iso-27001-definitions.htm, last visited: February 13 2012, Updated on December 22, 2011.

[130] Silvana Quaglini, Mario Stefanelli, Giordano Lanzola, Vincenzo Caporusso, and Silvia Panzarasa. Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine*, 22(1):65–80, 2001.

[131] Steven R. Rakitin. Coping with defective software in medical devices. *Computer*, 39:40–45, 2006.

[132] Marcello La Rosa, Florian Gottschalk, Marlon Dumas, and Wil M. P. van der Aalst. Linking Domain Models and Process Models for Reference Model Configuration. In *Business Process Management Workshops*, pages 417–430, 2007.

[133] B. Schilit, N. Adams, and R. Want. Context-aware Computing Applications. In *Workshop on Mobile Computing Systems and Applications*, pages 85 –90, 1994.

[134] B. Selic. The pragmatics of model-driven development. *Software, IEEE*, 20(5):19–25, 2003.

[135] B. Shackel. The concept of usability. In *Visual display terminals: Usability issues and health concerns*, pages 45–87. Englewood Cliffs, NJ:Prentice-Hall, 1984.

[136] J.H. Shin, G.S. Chung, K.K. Kim, J.S. Kim, B.S. Hwang, and K.S. Park. Ubiquitous House and Unconstrained Monitoring Devices for Home Healthcare System. In *6th International Special Topic Conference on Information Technology Applications in Biomedicine*, pages 201 –204, 2007.

[137] G. Sindre and A.L. Opdahl. Eliciting security requirements by misuse cases. In *Proc. of 37th International Conference on Technology of Object-Oriented Languages and Systems*, pages 120 –131, 2000.

[138] Guttorm Sindre and Andreas L. Opdahl. Templates for Misuse Case Description. In *Proc. of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality*, pages 4–5, 2001.

[139] Guttorm Sindre and Andreas L. Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10:34–44, 2005.

[140] E Smith and J H Eloff. Security in health-care information systems–current trends. *International Journal of Medical Informatics*, 54(1):39–54, 1999.

[141] Kamila Smolij and Kim Dun. Patient health information management: searching for the right model. Published online, the American Health Information Management Association, 2006. Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2047307, last visited: October 2011.

[142] Barry Smyth and Paul Cotter. A personalized television listings service. *Commun. ACM*, 43:107–111, 2000.

[143] Ian Sommerville and Pete Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, Inc., 1st edition, 1997.

[144] Bianying Song, K.-H. Wolf, M. Gietzelt, O. Al Scharaa, U. Tegtbur, R. Haux, and M. Marschollek. Decision support for teletraining of COPD patients. In *3rd Int. Conf. Pervasive Computing Technologies for Healthcare*, pages 1–6, 2009.

[145] Michael Spahn, Christian Dörner, and Volker Wulf. End user development: Approaches towards a flexible software design. In *16th European Conference on Information Systems, ECIS*, pages 303–314, 2008.

[146] Micro Speretta and Susan Gauch. Personalized Search Based on User Search Histories. In *the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 622–628. IEEE Computer Society, 2005.

[147] D. H. Stamatis. *Failure Mode and Effect Analysis - FMEA from Theory to Execution*. American Society for Quality Press, 2003.

[148] Bea Steenbekkers and Marlies van der Linden. Ageing in Europe: Trends and Challenges. In *ELDERATHOME: The prerequisites of the elderly for living at home: Criteria for dwellings, surroundings and facilities: Final report*. TTS Institute's Publication 393, 2004.

[149] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004.

[150] Alistair G. Sutcliffe and Shailey Minocha. Scenario-based Analysis of Non-Functional Requirements. In *Proc. of the Fourth International Workshop on Requirements Engineering: Foundation of Software Quality - REFSQ*, pages 219–234, 1998.

[151] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., 2nd edition, 2002.

[152] T-SENIORITY. Expanding the benefits of Information Society to Older People through digital TV channels, 2006. Available at: http://tseniority.idieikon.com/index.php/lang-en/the-project, last visited: January 2013.

[153] The Council of The European Union. Council conclusions on a safe and efficient healthcare through ehealth. In *Official Journal of the European Union*, December 2009.

[154] Adam Trendowicz and Teade Punter. Quality modeling for software product lines. In *Proc. of 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, QAOOSE*, 2003.

[155] United Nations International Strategy for Disaster Reduction UN-ISDR. Terminology on disaster risk reduction. Geneva, May 2009.

[156] US FDA Center for Devices and Radiological Health. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. Available at: http://www.fda.gov/MedicalDevices/DeviceRegulationand Guidance/GuidanceDocuments/ucm089543.htm, last visited: February 20 2012, May 2005.

[157] Axel van Lamsweerde, Simon Brohez, Renaud De Landtsheer, and David Janssens. From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering. In *Proc. of Workshop on Requirements for High Assurance Systems*, pages 49–56, 2003.

[158] Jan-Willem van ’T Klooster, Bert-Jan van Beijnum, and Hermie Hermens. Experiences in Testing Personalized Care Systems in Nursing Home. In *Proceedings of the 4th Dutch Bio-Medical Engineering Conference (BME2013)*, 2013.

[159] Stale Walderhug, Erlend Stav, and Marius Mikalsen. Experiences from Model-Driven Development of Homecare Services: UML Profiles and Domain Models. In *Models in Software Engineering*, volume 5421 of *LNCS*, pages 199–212. Springer Berlin / Heidelberg, 2009.

[160] Feng Wang and Kenneth J. Turner. Towards personalised home care systems. In *1st intl. Conf. on PErvasive Technologies Related to Assistive Environments*, PETRA ’08, pages 1–7. ACM, 2008.

[161] C.C. White, D. Fang, E.-H. Kim, W.B. Lober, and Y. Kim. Improving Healthcare Quality through Distributed Diagnosis and Home Healthcare. In *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, pages 168 –172, April 2006.

[162] Earl L Wiener. Cockpit automation. In *Human factors in aviation, Academic Press series in cognition and perception*, pages 433–461, 1988.

[163] Roel Wieringa. Design Science as Nested Problem Solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, pages 1–12, New York, 2009. ACM.

[164] Roel Wieringa. Design Science Methodology: Principles and Practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE 2010)*, pages 493–494. ACM, 2010.

[165] Roel Wieringa. A Unified Checklist for Observational and Experimental Research in Software Engineering (Version 1). Technical Report TR-CTIT-12-07, Enschede, March 2012.

[166] Roel Wieringa. Designing Technical Action Research and Generalizing from Real-World Cases. In *Advanced Information Systems Engineering*, volume 7328 of *Lecture Notes in Computer Science*, pages 697–698. Springer Berlin / Heidelberg, 2012.

[167] Roel Wieringa and Hans Heerkens. Designing Requirements Engineering Research. In *Workshop on Comparative Evaluation in Requirements Engineering (CERE'07)*, pages 36–48. IEEE Computer Society, October 2007.

[168] Roel Wieringa and Ayse Morali. Technical Action Research as a Validation Method in Information Systems Design Science. In *7th International Conference, DESRIST*, volume 7286 of *Lecture Notes in Computer Science*, pages 220–238. Springer, 2012.

[169] Markus Won, Oliver Stiemerling, and Volker Wulf. Component-Based Approaches to Tailorable Systems. In *End User Development*, volume 9 of *Human-Computer Interaction Series*, pages 115–141. Springer Netherlands, 2006.

[170] World Health Organization. *International classification of functioning, disability and health : ICF short version*. 2001.

[171] Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota, and Marten van Sinderen. Dynamic Homecare Service Provisioning Architecture. In *9th International Conference on Service-Oriented Computing and Applications (SOCA)*, pages 1–8. IEEE, 2011.

[172] Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota, and Marten van Sinderen. Toward Dynamic Service Provisioning in the Homecare Domain. In *5th International Conference on Pervasive Computing Technologies for Healthcare*

(PervasiveHealth), Workshop on Designing and Integrating Independent Living Technology, pages 292–299. IEEE, May 2011.

[173] Mohammad Zarifi Eslami, Brahmananda Sapkota, Alireza Zarghami, Eelco Vriezekolkand, Marten van Sinderen, and Roel Wieringa. Risk and accountability analysis of context-aware systems: a case study and some lessons learned. In *24th International Conference on Advanced Information Systems Engineering, CAiSE'12 FORUM*, June 25-29 2012.

[174] Mohammad Zarifi Eslami and Marten van Sinderen. Flexible home care automation adapting to the personal and evolving needs and situations of the patient. In *3rd International Conference on Pervasive Computing Technologies for Healthcare*, pages 1 –2, April 2009.

[175] Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service tailoring: Towards personalized homecare services. In *International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing*, pages 109–121. SciTePress, July 2010.

[176] Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Flexible Homecare Application Personalization and Integration Using Pattern-Based Service Tailoring: Supporting Independent Living of Elderly with IT. In *11th IEEE International Conference on Computer and Information Technology (CIT)*, pages 467–474. IEEE Computer Society, 2011.

[177] Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Personalized Service Creation by Non-technical Users in the Homecare Domain. *Procedia CS*, 5:409–417, 2011.

[178] Shuai Zhang, Sally I. McClean, Bryan W. Scotney, Xin Hong, Chris D. Nugent, and Maurice D. Mulvenna. Decision Support for Alzheimer's Patients in Smart Homes. In *CBMS*, pages 236–241, 2008.

[179] Qi Zhao, Gang Huang, Jiyu Huang, Xuanzhe Liu, and Hong Mei. A Web-Based Mashup Environment for On-the-Fly Service Composition. In *IEEE International Symposium on Service-Oriented System Engineering*, SOSE '08, pages 32–37. IEEE Computer Society, 2008.

Appendix A

Questionnaires

Usability Survey

Please rate the usability of the system by answering the all questions. If a question is not applicable, use "NA".

Strongly disagree <- 1 2 3 4 5 6 7 -> Strongly agree, NA (Not Applicable)

	1	2	3	4	5	6	7	NA
1. Overall, I am satisfied with how easy it is to use this system.								
2. It was simple to use this system.								
3. I can effectively complete my work using this system.								
4. I can effectively complete my work using this system.								
5. I am able to efficiently complete my work using this system.								
6. I feel comfortable using this system.								
7. It was easy to learn to use this system.								
8. I believe I became productive quickly using this system.								
9. The system gives error messages that clearly tell me how to fix problems.								
10. Whenever I make a mistake using the system, I recover easily and quickly.								
11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear.								
12. It is easy to find the information I needed.								
13. The information provided for the system is easy to understand.								
14. The information is effective in helping me complete the tasks and scenarios.								
15. The organization of information on the system screens is clear.								
16. The interface of this system is pleasant.								
17. I like using the interface of this system.								
18. This system has all the functions and capabilities I expect it to have.								
19. Overall, I am satisfied with this system.								

Figure A-1 Usability survey for the first series of experiments, close-end questions

Please provide your opinions on the system by answering the questions below.

Q1. List the most **negative** aspect(s):

1.

2.

3.

4.

Q2. List the most **positive** aspect(s):

1.

2.

3.

4.

Q3. Which parts are still missing in the system?

1.

2.

3.

4.

Figure A-2 Usability survey for the first series of experiments, open-end questions

Usability Survey of the UCare system (final questionnaires from the care-givers)

Please provide your opinions on the system by answering the questions below. You can also comment on each question if you think the question is not applicable or complete enough.

1. Which things you can **do** with the system that you could not do before?
   - Answer:
   - Comment on the answer (optional):
2. Which things you can**not do** with the system that you could do before?
   - Answer:
   - Comment on the answer (optional):
3. Using the system you can schedule, personalize and measure four care tasks namely: blood pressure, weight, oxygen saturation and medication intake. Can you see any other tasks which could be performed using the system which is not supported now?
   - Answer:
   - Comment on the answer (optional):
4. Have you ever get upset (frustrated, angry) using the system? If yes, please describe the situation.
   - Answer:
   - Comment on the answer (optional):
5. Have you ever been pleasantly surprised when you were using the system? A situation that you didn't expect. If yes, please describe the situation.
   - Answer:
   - Comment on the answer (optional):
6. Does using the system save your time? In which task(s) and in which way?
   - Answer:
   - Comment on the answer (optional):
7. Does using the system cause some tasks to take more time than before? In which task and in which way?
   - Answer:
   - Comment on the answer (optional):

Figure A-3 Usability survey for the second series of experiments, page 1

8. Does using the system make things easier? In which task(s) and in which way?
    - Answer:
    - Comment on the answer (optional):
9. Does using the system make things difficult? In which task(s) and in which way?
    - Answer:
    - Comment on the answer (optional):
10. Does using the system increase the **quality of care**? In which way?
    - Answer:
    - Comment on the answer (optional):
11. Does using the system decrease the **quality of care**? In which way?
    - Answer:
    - Comment on the answer (optional):
12. Does using the system increase the **quality of life** of elderly? In which way?
    - Answer:
    - Comment on the answer (optional):
13. Does using the system decrease the **quality of life** of elderly? In which way?
    - Answer:
    - Comment on the answer (optional):
14. In which situations do you think the IT-based homecare system could be used successfully in practice? Please give examples.
    - Answer:
    - Comment on the answer (optional):
15. In which situations do you think a system like this **should not** be used? Please give examples
    - Answer:
    - Comment on the answer (optional):

Figure A-4 Usability survey for the second series of experiments, page 2

Author Publications

Alireza Zarghami, Eelco Vriezekolk, Mohammad Zarifi Eslami, Marten van Sinderen, and Roel Wieringa. Assumption-based Risk Identification Method (ARM) in Dynamic Service Provisioning. In *The 21st IEEE International Requirements Engineering Conference (RE)*. IEEE, 2013.

Alireza Zarghami, Mohammad Zarifi Eslami, Marten van Sinderen, and Roel Wieringa. Dynamic homecare service provisioning: A field test and its results. In *Process Support and Knowledge Representation in Health Care*. Springer, 2013.

Mohammad Zarifi Eslami, Brahmananda Sapkota, Andrea Herrmann, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Risk Driven Requirements Specification (RiDeRS) of IT-based Homecare Systems. In *Proceedings of the CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering (CAiSE)*, CEUR Workshop Proceedings. CEUR-WS.org, 2013.

Mohammad Zarifi Eslami, Alireza Zarghami, Marten van Sinderen, and Roel Wieringa. Care-giver Tailoring of IT-based Healthcare Services for Elderly at Home: A Field Test and its Results. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE, 2013.

Ashiful Alam, Mohammad Zarifi Eslami, and Klaas Sikkel. Elderly and homecare tasks: A literature review on problems. In *4th IADIS International Conference e-Health*, pages 211–216, 2012.

Henry Been and Mohammad Zarifi Eslami. A Survey on Tasks Performed in Eldercare. In *4th IADIS International Conference e-Health*, pages 176–181, 2012.

Alireza Zarghami, Brahmananda Sapkota, Mohammad Zarifi Eslami, and Marten van Sinderen. Decision as a Service: Separating

Decision-making from Application Process Logic. In *The 16th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, pages 103–112. IEEE, 2012.

Mohammad Zarifi Eslami, Brahmananda Sapkota, Alireza Zarghami, Eelco Vriezekolk, Marten van Sinderen, and Roel Wieringa. Risk Identification of Tailorable Context-aware Systems: a Case Study and Lessons Learned. In *Proceedings of the CAiSE'12 Forum at the 24th International Conference on Advanced Information Systems Engineering (CAiSE)*, volume 855 of *CEUR Workshop Proceedings*, pages 40–49. CEUR-WS.org, 2012.

Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota, and Marten van Sinderen. Service Realization and Compositions Issues in the Homecare Domain. In *6th International Conference on Software and Data Technologies (ICSOFT), Volume 1*, pages 347–356. SciTePress, July 2011.

Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota, and Marten van Sinderen. Toward Dynamic Service Provisioning in the Homecare Domain. In *5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Workshop on Designing and Integrating Independent Living Technology*, pages 292–299. IEEE, May 2011.

Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Personalized Service Creation by Non-technical Users in the Homecare Domain. *Procedia CS*, 5:409–417, 2011.

Alireza Zarghami, Mohammad Zarifi Eslami, Brahmananda Sapkota, and Marten van Sinderen. Dynamic Homecare Service Provisioning Architecture. In *9th International Conference on Service-Oriented Computing and Applications (SOCA)*, pages 1–8. IEEE, 2011.

Alireza Zarghami, Mohammad Zarifi Eslami, and Marten van Sinderen. What Do Homecare Provider Stories Tell Us about Dynamicity? In *17th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, volume 44, pages 154–155. ICB Institut fur Informatik und Wirtschaftsinformatik, 2011.

Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Flexible Homecare Application Personalization and Integration Using Pattern-Based Service Tailoring: Supporting Independent Living of Elderly with IT. In *11th IEEE International Conference on Computer and Information Technology (CIT)*, pages 467–474. IEEE Computer Society, 2011.

Mohammad Zarifi Eslami, Alireza Zarghami, Brahmananda Sapkota, and Marten van Sinderen. Service Tailoring: Towards Personalized Homecare Services. In *4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing (ACT4SOC)*, pages 109–121. SciTePress, July 2010.

Athanasios Karapantelakis, Alisa Devlic, Mohammad Zarifi Eslami, and Saltanat Khamit. Printing in Ubiquitous Computing Environments. In *6th International Conference on Ubiquitous Intelligence and Computing*, UIC '09, pages 83–98. Springer-Verlag, 2009.

Ingrid Mulder, Yvonne Schikhof, Martijn Vastenburg, Alan Card, Tory Dunn, Andreas Komninos, Marilyn McGee-Lennon, Mark Santcroos, Gabriele Tiotto, Mieke van Gils, Jan-Willem van 't Klooster, Annelies Veys, and Mohammed Zarifi Eslami. Designing with Care: The Future of Pervasive Healthcare. *IEEE Pervasive Computing*, 8(4):85–88, 2009.

Mohammad Zarifi Eslami and Marten van Sinderen. Flexible home care automation adapting to the personal and evolving needs and situations of the patient. In *3rd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1 –2. IEEE, 2009.

SIKS Dissertation Series

====
1998
====

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games
1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

====
1999
====

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling;
Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven
Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent
Mechanism for Discrete Reallocation.

====
2000
====

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie;
een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations,
Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with
Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language
for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

====
2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)

Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments
inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative
E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

====
2003
====

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

====
2004
====

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

====
2005
====

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based
Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in
Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics
meets pragmatics

2005-15 Tibor Bosse (VU)

Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by
Exploiting Application Semantics

====
2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools
for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment,
and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of
Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web

2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured
Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated
media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

====
2007
====

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems:
a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy:
a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)

Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI)
Structural Features in XML Retrieval

2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

====
2008
====

2008-01 Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach

2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective

2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning

2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference

2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT)
Discourse oriented summarization

2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach

2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks

2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms
for the Markov Decision Process Framework in First-Order Domains.

2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective

2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM)
Adaptive Active Vision

2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of
Focused Text Search

2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van
elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven

2008-21 Krisztian Balog (UVA)
People Search in the Enterprise

2008-22 Henk Koning (UU)
Communication of IT-Architecture

2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching

2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using
Spender-signed Currency

2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled

2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks

2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users,
and Other Humans

2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and
Querying Media Content

2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure

====
2009
====

2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU)
Understanding Classification

2009-07 Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law & the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
perating Guidelines for Services

2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT)
Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System

2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU)
Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM)
Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VUA)
Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI)
"RAM: Array Database Management through Relational Mapping"

2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage

2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors

2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management:
An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings

2009-42 Toine Bogers
Recommender Systems for Social Bookmarking

2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and

Mobile Ambients

2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations

2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful

2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion

====
2010
====

2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter

2010-02 Ingo Wassink (UT)
Work flows in Life Science

2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems

2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI

2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance

2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children 2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning

2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice

2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA)
People’s Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of

Emergent Narrative

2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation

2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions

2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL)
Automatisch contracteren

2010-28 Arne Koopman (UU)
Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI)
Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT)
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31 Victor de Boer (UVA)
Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT)
An Adaptive Service Oriented Architecture: Automatically solving
Interoperability Problems

2010-33 Robin Aly (UT)
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning; Facilitating competence development
through a learning path specification

2010-37 Niels Lohmann (TUE)
Correctness of services and their composition

2010-38 Dirk Fahland (TUE)
From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web

2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting -
the computational e3-service approach

2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous
Data Sources, Illustrated in the Television Domain

2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services

2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples

2010-48 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2010-49 Jahn-Takeshi Saito (UM)
Solving difficult game positions

2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives

2010-51 Alia Khairia Amin (CWI)
Understanding and supporting information seeking tasks in multiple sources

2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive
Models of Trust and Attention

2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access

====
2011
====

2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of
an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of
temporal-difference learning algorithms

2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of
an Emerging Discipline.

2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues 2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT)
Cloud Content Contention

2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)
The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans
On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein(UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification

2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR)
Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR)
Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU)
Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU)
Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU)
Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU)
Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN)
Machine Learning for Pairwise Data, Applications for Preference Learning
and Supervised Network Inference

2011-38 Nyree Lemmens (UM)
Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU)

Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU)
Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT)
Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU)
Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU)
Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT)
Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)
Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD)
Towards Contextualized Information Delivery: A Rule-based Architecture
for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz(VU)
Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT)
Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT)
Conversational interfaces for task-oriented spoken dialogues:
design aspects influencing interaction quality

====
2012
====

2012-01 Terry Kakeeto (UvT)
Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair(VU)
Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU)
Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU)
Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU)
Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU)
Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU)
When the Going Gets Tough: Exploring Agent-based Models of Human Performance
under Demanding Conditions

2012-08 Gerben de Vries (UVA)
Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT)
Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE)
Towards a Generic Distributed Adaptive Hypermedia Environment

2012-11 J.C.B. Rantham Prabhakara (TUE)
Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE)

Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT)
Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14 Evgeny Knutov(TUE)
Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU)
Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes

2012-16 Fiemke Both (VU)
Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT)
Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU)
Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE)
What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN)
Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD)
Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT)
Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT)
Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

2012-24 Laurens van der Werff (UT)
Evaluation of Noisy Transcripts for Spoken Document Retrieval

2012-25 Silja Eckartz (UT)
Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA)
Making Sense of Legal Text

2012-27 Hayrettin Grkk (UT)
Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT)
Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT)
Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD)
Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN)
A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD)
Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN)
Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN)
Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU)
Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN)
Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN)
A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU)
Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT)
Risk-aware Design of Value and Coordination Networks

2012-40 Agus Gunawan (UvT)
Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU)
Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU)
Reflection Amplifiers in self-regulated Learning

2012-43 Withdrawn
-

2012-44 Anna Tordai (VU)
On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT)
A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA)
Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA)
Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE)
Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM)
Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD)
Ontology driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD)
Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

====
2013
====

2013-01 Viorel Milea (EUR)
News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI)
MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

2013-03 Szymon Klarman (VU)
Reasoning with Contexts in Description Logics

2013-04 Chetan Yadati(TUD)
Coordinating autonomous planning and scheduling

2013-05 Dulce Pumareja (UT)
Groupware Requirements Evolutions Patterns

2013-06 Romulo Gonzalves(CWI)
The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

2013-07 Giel van Lankveld (UT)
Quantifying Individual Player Differences

2013-08 Robbert-Jan Merk(VU)
Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

2013-09 Fabio Gori (RUN)
Metagenomic Data Analysis: Computational Methods and Applications

2013-10 Jeewanie Jayasinghe Arachchige(UvT)
A Unified Modeling Framework for Service Design

2013-11 Evangelos Pournaras (TUD)
Multi-level Reconfigurable Self-organization in Overlay Services

2013-12 Maryam Razavian (VU)
Knowledge-driven Migration to Services

2013-13 Mohammad Zarifi Eslami (UT)
Service Tailoring: User-centric Creation of Integrated IT-based Homecare Services to Support Independent Living of Elderly

Samenvatting

Dit proefschrift richt zich op de problematiek van het zelfstandig wonen van ouderen door middel van IT-gebaseerde thuiszorg. Zelfstandig wonen van ouderen wordt gezien als een manier om om te gaan met de gevolgen van een vergrijzing van de bevolking (vooral in geindustrialiseerde landen), waaronder stijgende uitgaven voor de gezondheidszorg en een bijbehorend tekort aan gezondheidszorg-professionals.

Ons belangrijkste doel is het verbeteren van de dienstontwikkelingsproces voor het thuiszorg domein, door middel van het verminderen van de benodigde IT-vaardigheden, tijd en moeite om nieuwe diensten te creeren, en ondertussen de individuele behoeften van de gebruikers en de functionaliteit van de gecreÃ«erde diensten op elkaar af te stemmen. We noemen onze benadering van het creeren van diensten “service tailoring”. Service tailoring, zoals voorgesteld in dit proefschrift, is een manier van het creeren van nieuwe diensten en het aanpassen van reeds bestaande diensten, door de professionals in de gezondheidszorg (zorgverleners) in het creatieproces te betrekken en de ouderen (zorgontvangers) de primaire gebruikers van de gemaakte diensten zijn.

Voor het opstellen van een programma van eisen, zijn we begonnen met een literatuurstudie van bestaande thuiszorgsystemen en dienstpersonalisatie-technieken waarbij de gebruiker centraal staat, en een marktonderzoek van de huidige thuiszorg-technologieen en producten. Vervolgens voerden we vijf series van interviews met zorgverleners in een zorginstelling in Nederland[1]. Deze instelling bestaat uit woonblokken, waar ouderen kunnen wonen en 24 uur per dag zorgdiensten krijgen van professionele zorgverleners. De eerste, tweede en derde serie interviews werden gedaan voor de prototyping van onze aanpak, en de vierde en vijfde serie interviews werden uitgevoerd om de bruikbaarheid van de aanpak te controleren nadat het prototype werd gebruikt door de zorgverleners en de zorgontvangers. We voerden

[1] http://www.orbisconcern.nl/

de eerste serie van interviews door middel van vragenlijsten om de bestaande situatie te analyseren en om te leren welke soorten diensten werden verwacht van een IT-thuiszorg systeem. Deze interviews waren nuttig om inzicht te krijgen in veelvoorkomende taken en hoe deze taken worden uitgevoerd. Na het ontwerpen van onze aanpak hebben wij de tweede serie van het interview met dezelfde zorg-verleners uitgevoerd om onze ontwerpen te valideren en te verfijnen voor de implementatie van een prototype van onze aanpak.

Het gebruik van nieuwe IT-diensten kunnen nieuwe soorten risico's introduceren in de omgeving waar deze diensten worden uitgevoerd. Dit is met name het geval in het thuiszorg domein, omdat het welzijn van mensen en gezondheid kan afhangen van de diensten. Risico's kunnen mogelijk leiden tot ongewenste of gevaarlijke situaties, en een gebrek aan vertrouwen in de diensten, bijv. door meervoudige risico situaties, wat kan leiden tot een daling in het gebruik van de IT-diensten in de praktijk. We hebben een Risk Driven Requirements Specification (RiDeRS) methode voorgesteld om potentiele risico's van het gebruik van IT-gebaseerde diensten te identificeren, en om aanvullende eisen voor deze diensten (en de onderliggende IT-systeem) op te stellen om deze risico's te verminderen of te voorkomen. Om RiDeRS te definieren, hebben we een literatuurstudie uitgevoerd van bestaande requirements elicitatiemethodes die risico's gebruiken om risico-verkleinende requirements te verkrijgen. Vergeleken met bestaande methoden, is het vernieuwende aan RiDeRS dat het meer systematisch rekening houdt met de eigenschappen van van de omgeving van het systeem. In RiDeRS hebben we de eigenschappen van gebruikers beschouwd als aanvulling op hun doelen om een lijst van mogelijke risico's te identificeren en de eisen gespecificeerd die risico's kunnen voorkomen of inperken. Daarom voerden we een risico analyse uit met behulp van RiDeRS door het uitvoeren van de derde serie interviews met dezelfde zorgverleners om potentiele risico's van het gebruik van onze diensten te identificeren en daaropvolgend verdere requirements van het onderliggende IT-systeem te identificeren.

Na de evaluatie van het resultaat van de eerste drie reeksen interviews, hebben we de architectuur van een dienst maatwerk-platform ontworpen. Om deze architectuur te evalueren, ontwikkelden we een prototype van het dienst maatwerk-platform als onderdeel van het U-Care[2] project, dat vervolgens werd gebruikt in twee reeksen van experimenten om de eigenschappen van de benadering te valideren. De experimenten werden uitgevoerd in een real-life toepassing bij de zorginstelling. Een aantal gebruikers (8 zorgontvangers en 4 zorgverleners) hebben zich opgegeven als vrijwilliger om het U-Care

[2]http://www.utwente.nl/ewi/ucare/

system[3] te gebruiken.

We veronderstelden dat dienst maatwerk moet worden uitgevoerd door een zorgverlener. We identificeerden verschillende soorten zorgverleners die in hun dagelijks leven interactie hebben met zorgontvangers en hen hulp verlenen in het thuiszorg domein. De geidentificeerde types verzorgers zijn professionele verpleegkundigen, familieleden, mantelzorgverleners (vrijwillige, niet-professionele verzorgers), ergotherapeuten, fysiotherapeuten, artsen, apothekers en psychologen. In onze optiek, dient een professionele verpleegkundige, als een verzorger, degene te zijn die de diensten zou moeten afstemmen, omdat zorgontvangers meer tijd doorbrengen met professionele verpleegkundigen dan met andere vormen van zorgverleners.

We wilden IT-transparantie bereiken door gebruik te maken van het concept van service building blocks (SBBs). Het begrip SBB wordt gebruikt om de aanduiding van de kleinste beheersbare eenheid van dienstfunctionaliteit vanuit het oogpunt van verzorgers (bijvoorbeeld een herinnering-SBB geeft zorgontvangers een melding dat een actie uitgevoerd moet worden). Een SBB biedt een generieke service-interface die gebruikt kan worden door zorgverleners in het proces van dienstmaatwerk. Een SBB geeft ook een overzicht van configuratieparameters om een verzorger in staat te stellen om verschillende aspecten van de SBB te specificeren, zoals service activiteiten en gebruikersinterface modaliteiten. Elke SBB representeert een concrete dienst of alternatieve concrete diensten, en verbergt de technische details die niet relevant zijn om een service op maat te maken (d.w.z. niet relevant voor de zorgverlener). Het resultaat van een dienst maatwerk proces is een zogenaamd dienstplan, dat een samengestelde dienst is afgestemd op de behoeften van een specifieke zorg-ontvanger zoals begrepen door de verzorger. Een dienstplan bevat voldoende informatie voor het geautomatiseerd afleiden van een volledige implementatie op het doelplatform voor de uitvoering ervan. Het vanuit het niets ontwerpen van een dergelijk dienstplan is een moeilijke en tijdrovende bezigheid. We gebruiken het concept van ontwerppatronen om het proces van het creeren van een service plan te vereenvoudigen. We maken gebruik van behandelingspatronen als uitgangspunt voor het proces van het op maat maken, waarbij een behandelingspatroon een activiteitsstructuur voor het uitvoeren van een generieke thuiszorg taak is (bijv. de bloeddruktoezichtstaak). Zo hoeft de zorgverlener geen dienst te creeren vanuit het niets, maar selecteert de thuiszorgtaak, ondersteund door middel van een

[3] Het U-Care systeem omvat een maatwerk-platform, een provisioning platform, en sommige thuiszorg toepassingen

menu. Het maatwerk-platform presenteert dan het overeenkomstige behandelingspatroon als het oorspronkelijke dienstplan, dat verder moet worden verfijnd en aangevuld door de zorgverlener.

Het service maatwerk-platform is verantwoordelijk voor het verbeteren van het creeren en afstemmen van de dienstverlening van plan door middel van een grafische gebruikersinterface (GUI) voor de zorgverleners. Om de haalbaarheid van de voorgestelde architectuur te demonstreren, ontwikkelden we een prototype van het maatwerk-platform. Het prototype werd in twee series van experimentele praktijkproeven (met een totale duur van 4 maanden) geevalueerd. Na de eerste reeks experimenten hebben we een vierde serie van interviews met de zorgverleners gehouden, de resultaten geevalueerd en het systeem verbeterd. Na de tweede reeks experimenten waarin de gebruikers het verbeterde systeem gebruikten, voerden we de laatste serie van interviews met de gebruikers uit om de bruikbaarheid van het systeem en het nut van de dienst maatwerk-platform te evalueren. We vroegen de gebruikers om hun mening over het systeem om te zien of een dergelijk systeem gebruikt zou kunnen worden in de praktijk en of het inderdaad de zorgverleners tijd zou kunnen besparen en de kwaliteit van leven van de zorgontvangers zou kunnen verhogen. Tot slot reflecteren we op de verrichte werkzaamheden en de bereikte resultaten in het kader van het thuiszorg-domein, en vervolgens bespreken of de voorgestelde aanpak breder kan worden toegepast in de thuiszorg of zelfs in de bredere klasse van context-aware kritieke systemen.

Glossary

accountability The allocation of responsibility in case of materialization of a safety risk. 55

availability The readiness for correct service, or more elaborately, accessibility and usability when needed by an authorized entity. 55

availability risk The possibility of the system or its services not being available when needed. 55

building block In information systems, building block is the smallest distinct self-contained unit, and it is represented as a abstraction of implementation component which can be used in tailoring applications. 28

capability assumption Relates to capabilities (and limitations) of an end-user. We are only interested in capabilities that are needed for accessing a service through the provided interface. 59

care-giver Expert and volunteer who provides care and social services to elderly. 22

care-receiver Elderly who lives in their own home or in elderly-care institution and receives care services from their care-givers. 22

context information Any information about the physical context of the system which can be used to adapt the response of a homecare system to add value to the provided services to entitled users. 25

elder-care institution Professional institution, which hires experts and volunteer care-givers to provide care and social services to their care-receivers. 22

elderly person A person who needs the help of professional care-givers to deal with age-associated diseases. Examples of aging-associated diseases are cardiovascular disease, type 2 diabetes and hypertension. The incidence of these and other diseases increases rapidly with age. 22

homecare Providing care services to care-receivers in their home environment. 23

homecare system A system includes platforms, services, devices, data and networks that are required to support independent living of elderly. 23

identity assumption Relates to the identity of an end-user who use a device with sensors that measures necessary user context parameters. 61

location assumption Relates to the location of an end-user carrying a device with sensors that measures location parameter for the context-aware service. 61

medical protocol A sequence of connected steps to be followed by care-givers for each specific task. 47

need Different needs, requirements, preferences and situations of an elderly. 3

needs assumption Relates to an end-user having needs matching the user context in which the context-aware service is being offered. 61

procedure assumption Relates to the procedures followed by an end-user to access and consume a service. The designer of a system assumes a procedure for each service. 60

risk The possibility of loss or injury. 54

safety risk A possible threat to the safety of a user, where safety is the absence of loss or injury. 55

service building block Homecare support actions are represented as user-level service descriptions, and referred to as Service Building Blocks (SBBs) mainly when these actions have independently useful functionalities. Each SBB corresponds to functionality that has been implemented by a device and/or software application, and is available for use by the care-receiver. Each SBB has configuration parameters for specifying behavior constraints. By composing and configuring the SBBs in a treatment pattern, we will be able to provide required homecare services to care-receivers. 90

service plan The outcome of a service tailoring process, which represents a composite service tailored to the specific needs of a specific elderly as understood by the care-givers. 85

service tailoring Set of activities simply to modify a computer application within the context of its use by its end-users. 28

service tailoring process A flow of actions to help an end-user in creating the personalized service that is needed. 7

tailorability of a homecare system A care-giver can configure the behavior of the system without help from technical personnel. 33

tailoring platform A platform includes a architecture and tailoring software, where the combination provides an environment for tailoring of services by the care-givers, and requires minimal technical knowledge and less time/effort. 23

timing assumption Relates to the availability of an end-user (at a specific place/time). 61

treatment pattern An activity structure for handling a generic homecare task. We make use of treatment patterns as a starting point for the tailoring process. 47

user preferences Rules and conditions set by the user to characterize his/her desire in each context situation. 104

user profile A structured representation of the user's personal data, needs, and preferences that is used by the tailoring platform to configure SBBs of a service plan. 103