UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

A reference architecture for healthcare supportive home systems from a systems-of-systems perspective

Lina María Garcés Rodríguez

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: ______________________

Lina María Garcés Rodríguez

A reference architecture for healthcare supportive home systems from a systems-of-systems perspective

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP and to the Université de Bretagne-Sud – UBS, Institut de recherche en informatique et systèmes aléatoires – IRISA, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics (ICMC-USP) and Doctorat en Informatique (UBS), in accordance with the international academic agreement for Ph.D double degree signed between ICMC-USP and UBS. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics / Computer Science

Advisor: Prof. Dr. Elisa Yumi Nakagawa (ICMC-USP, Brazil)
Advisor: Prof. Dr. Flávio Oquendo (IRISA-UBS, France)

USP – São Carlos
July 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

G216r

GARCÉS RODRÍGUEZ, Lina María
A reference architecture for healthcare supportive home systems from a systems-of-systems perspective / Lina María GARCÉS RODRÍGUEZ; orientadora Elisa Yumi NAKAGAWA; coorientador Flavio OQUENDO. -- São Carlos, 2018.
213 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2018.

1. Reference Architecture. 2. System-of-Systems. 3. Healthcare Supportive Home system. 4. Software Architecture. 5. Health Service Bus. I. NAKAGAWA, Elisa Yumi , orient. II. OQUENDO, Flavio, coorient. III. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Lina María Garcés Rodríguez

Uma arquitetura de referência para sistemas de casas inteligentes de apoio ao cuidado da saúde desde uma perspectiva de sistemas-de-sistemas

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e à Université de Bretagne-Sud – UBS, Institut de recherche en informatique et systèmes aléatoires – IRISA, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional (ICMC-USP) e Doutora em Informática (UBS), de acordo com o convênio acadêmico internacional para dupla titulação de doutorado assinado entre o ICMC-USP e a UBS. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional / Ciência da Computação

Orientadora: Prof. Dr. Elisa Yumi Nakagawa (ICMC-USP, Brazil)
Orientador: Prof. Dr. Flávio Oquendo (IRISA-UBS, France)

USP – São Carlos
Julho de 2018

To my family Myriam, Ricardo, Silvia and Paula for their support in this journey.

ACKNOWLEDGEMENTS

This thesis is not only the result of the last five years of work, but it was a consequence of cumulative learning, experiences, and collaborations, in both personal and professional levels.

I am deeply grateful to Prof. Elisa Yumi Nakagawa for her support and guide during this stage of my life. Most of the amazing experiences during my Ph.D. studies would not have been possible without her aid. I expect this work is the start of future achievements together.

To Prof. Flavio Oquendo for his contributions and confidence vote at accepting me as a student in the ArchWare laboratory at IRISA-UBS.

To Eliana and Carla, who have contributed to important aspects of home health care.

To the examination board, who gave me suggestions to improve my research.

To my parents Ricardo and Myriam, who have supported me in all my decisions even when those decisions were difficult to accept.

To my sisters Silvia and Paula, and my life sisters Eli, Caro, and Vânia, who remember me that dreams come true.

My very special thanks to Brauner. "A true friend is the one who holds your hand and touches your heart" (Gabo).

To all my Brazilian friends, with whom I have created my second home.

To my Colombian friends, who despite distance are always present.

To my French friends, who will always be in my heart.

Results obtained with this thesis were possible to the financial support from the Brazilian funding agencies FAPESP (Research Support Foundation of the State of São Paulo) through the Grants No: 2013/20317-9 and 2015/19192-2, and CAPES (Coordination for higher Education Staff Development).

“What is the use of living, if it be not to strive for noble causes and to make this muddled world a better place for those who will live in it after we are gone? ”

(Winston Churchill, 1908.)

ABSTRACT

Population ageing has been taking place all over the world, being estimated that 2.1 billion people will be aged 60 or over in 2050. Healthcare Supportive Home (HSH) Systems have been proposed to overcome the high demand of remote home care for assisting an increasing number of elderly people living alone. Since a heterogeneous team of healthcare professionals need to collaborate to continually monitor health status of chronic patients, a cooperation of pre-existing e-Health systems, both outside and inside home, is required. However, current HSH solutions are proprietary, monolithic, high coupled, and expensive, and most of them do not consider their interoperation neither with distributed and external e-Health systems, nor with systems running inside the home (e.g., companion robots or activity monitors). These systems are sometimes designed based on local legislations, specific health system configurations (e.g., public, private or mixed), care plan protocols, and technological settings available; therefore, their reusability in other contexts is sometimes limited. As a consequence, these systems provide a limited view of patient health status, are difficult to evolve regarding the evolution of patient's health profile, do not allow continuous patients monitoring, and present limitations to support the self-management of multiple chronic conditions. To contribute to solve the aforementioned challenges, this thesis establishes *HomecARe*, a reference architecture for supporting the development of quality HSH systems. *HomecARe* considers HSH systems as Systems-of-Systems (SoS) (i.e., large, complex systems composed of heterogeneous, distributed, and operational and managerial independent systems), which achieve their missions (e.g., improvement of patients' quality of life) through the behavior that emerges as result of collaborations among their constituents. To establish *HomecARe*, a systematic process to engineer reference architectures was adopted. As a result, *HomecARe* presents domain knowledge and architectural solutions (i.e., architectural patterns and tactics) described using conceptual, mission, and quality architectural viewpoints. To assess *HomecARe*, a case study was performed by instantiating *HomecARe* to design the software architecture of *DiaManT@Home*, a HSH system to assist at home patients suffering of diabetes mellitus. Results evidenced *HomecARe* is a viable reference architecture to guide the development of reusable, interoperable, reliable, secure, and adaptive HSH systems, bringing important contributions for the areas of e-Health, software architecture, and reference architecture for SoS.

Keywords: Reference Architecture, Software Architecture, System-of-Systems, Healthcare Supportive Home system, Chronic Condition, Health Service Bus.

RESUMO

O envelhecimento da população é um fenômeno mundial e estima-se que no ano 2050, 2,1 bilhões de pessoas terão 60 anos ou mais. Sistemas de casas inteligentes para o cuidado da saúde (*em inglês Healthcare Supportive Home - HSH systems*) têm sido propostos para atender a alta demanda de serviços de monitoramento contínuo do número cada vez maior de pacientes que vivem sozinhos em suas residências. Considerando que o monitoramento do estado de saúde de pacientes crônicos requer a colaboração de equipes formadas por profissionais de várias especialidades, é fundamental que haja cooperação entre sistemas eletrônicos de saúde (por exemplo, sistemas de prontuário eletrônico ou sistemas de atenção de emergência), sendo eles externos ou internos à residência. Entretanto, as soluções de HSH existentes são comerciais, monolíticas, altamente acopladas e de alto custo. A maioria delas não considera a interoperabilidade entre sistemas distribuídos e exteriores ou internos à residência dos pacientes, como é o caso de robôs de companhia e monitores de atividade. Além disso, os sistemas de HSH muitas vezes são projetados com base em legislações locais, na estrutura do sistema de saúde (por exemplo, público, privado ou misto), nos planos de cuidados nacionais e nos recursos tecnológicos disponíveis; portanto, a reusabilidade desses sistemas em outros contextos é não é uma tarefa trivial. Em consequência, os sistemas de HSH existentes oferecem uma visão restrita do estado de saúde do paciente, são difíceis de evoluir acompanhando as mudanças no perfil de saúde do paciente, impossibilitando assim seu monitoramento contínuo e limitando o suporte para o paciente na autogestão de suas múltiplas condições crônicas. Visando contribuir na resolução dos desafios apresentados, esta tese estabelece a *HomecARe*, uma arquitetura de referência para apoiar o desenvolvimento de sistemas de HSH de qualidade. A *HomecARe* considera os sistemas de HSH como Sistemas-de-Sistemas (do inglês Systems-of-Systems - SoS) (ou seja, sistemas grandes e complexos formados por outros sistemas heterogêneos, distribuídos e que apresentam independência em seu gerenciamento e operação), que cumprem suas missões (por exemplo, melhoria da qualidade de vida do paciente) mediante o comportamento que emerge resultante da colaborações entre seus sistemas constituintes. Para estabelecer a *HomecARe*, foi adotado um processo sistemático que apoia a engenharia de arquiteturas de referência. Como resultado, a *HomecARe* contém o conhecimento do domínio, bem como soluções arquiteturais (por exemplo, padrões arquiteturais e táticas) que são descritas usando os pontos de vista conceitual, de missão e de qualidade. A *HomecARe* foi avaliada por meio da condução de um estudo de caso em que a arquitetura de referência foi instanciada para projetar o *DiaManT@Home*, um sistema de HSH que visa apoiar pacientes diagnosticados com diabetes mellitus na autogestão de sua doença. Os resultados obtidos evidenciaram que a *HomecARe* é uma arquitetura de referência viável para guiar o desenvolvimento de sistemas de HSH reusáveis, interoperáveis, confiáveis, seguros e adaptativos, trazendo importantes contribuições nas áreas de saúde eletrônica, arquitetura de software e arquiteturas de referência para SoS.

Palavras-chave: Arquitetura de Referência, Arquitetura de Software, Sistema-de-Sistemas, Casa Inteligente, Doença Crônica, Barramento de Serviços de Saúde.

RÉSUMÉ

Le vieillissement de la population est une tendance mondiale. Selon les estimations, en 2050, 2.1 milliard de personnes seront âgés de 60 ans ou plus. Les logiciels d'aide aux soins de santé à domicile (ou en anglais Healthcare Supportive Home -HSH systems) ont été proposés pour répondre à la forte demande de soins de santé à distance pour les personnes âgées vivant seules. Étant donné que les équipes de professionnels de la santé ont besoin de collaborer pour continuellement surveiller l'état de santé des patients souffrant de maladies chroniques, il est nécessaire de faire coopérer les systèmes logiciels d'e-santé préexistants. Cependant, les systèmes de HSH actuels sont propriétaires, monolithiques, fortement couplés, et coûteux ainsi que la plupart d'entre eux ne considèrent pas des interactions dynamiques avec systèmes de e-Health fonctionnant à l'intérieur (e.g., des robots compagnons interactifs ou moniteurs d'activité) ni à l'extérieur de la maison. Ces systèmes sont parfois conçus et fondés sur des législations locales, des configurations des systèmes de santé spécifiques (e.g., publique, privé ou hybride), des plans nationaux de soins de santé, et des ressources technologiques disponibles; leur réutilisation dans d'autres contextes est donc souvent limitée. De ce fait, les systèmes de HSH fournissent une vue limitée de l'état de santé des patients, sont difficiles à évoluer en fonction de l'évolution de la santé des patients et ne permettent pas la surveillance constante des patients. Ils ont des limites sérieuses pour aider l'autogestion des multiples maladies chroniques. En réponse à ces problématiques, cette thèse propose *HomecARe*, une architecture de référence pour permettre le développement des systèmes logiciel de HSH de qualité. *HomecARe* considère les systèmes HSH comment Systèmes-de-Systèmes (en anglais Systems-of-Systems - SoS) (i.e., systèmes complexes à grande échelle qui sont composées de systèmes hétérogènes, distribués, et avec indépendance opérationnelle et managériale), qui réalisaient leurs missions (e.g., l'amélioration de la qualité de vie des patients) grâce aux comportements qui émergent des collaborations entre les différents systèmes constitutifs. Pour établir *HomecARe*, un processus systématique pour la conception des architectures de référence a été adopté. *HomecARe* représente la connaissance du domaine et des solutions architecturales (e.g., patrons et stratégies d'architecture) en utilisant les points de vue architecturales des concepts, missions, et qualité. Pour évaluer *HomecARe*, un cas d'étude a été mené. Dans cette étude, *HomecARe* a été utilisé dans la conception de l'architecture logicielle du *DiaManT@Home*, un système de HSH pour l'assistance aux patients diabétiques. Les résultats montrent qui *HomecARe* est une solution viable pour guider le développement des systèmes de HSH réutilisables, interopérables, fiables, sécurisés, et adaptatifs. Cette thèse apporte d'importantes contributions dans les domaines d'e-santé, architectures logicielles, et architectures de référence de SoS.

Mots-clés: Architecture de Référence, Architecture Logicielle, Systèmes-de-Systèmes, Maisons Intelligents, Maladies Chroniques, Bus de Service de Santé.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AAL	Ambient Assisted Living
ADL	Architecture Description Language
AoDL	Activities of Daily Life
BPM	Business Process Management
BPMN	Business Process Management Notation
BPSv	Business Process Service
CSSv	Constituent System Service
CSv	Control Service
DiaManTHome	Diabetes Management at Home
e-Health	Electronic Health
ESB	Enterprise Service Bus
GORE	Goal Oriented Requirements Engineering
HHA	Home Health Aide
HHT	Home Healthcare Team
HIS	Health Information Systems
HL7	Health Level Seven
HomecARe	Home Healthcare Systems, an Architecture of Reference
HSB	Health Service Bus
HSH	Healthcare Supportive Home
ICD	International Classification of Diseases
KAOS	Keep All Objectives Satisfied
LOINC	Logical Observation Identifier Names and Codes
LPN	Licensed Practical Nurse
m-KAOS	KAOS for SoS missions
MD-OD	Physician
MSW	Medical Social Worker
MV	Mission View
OT	Occupational Therapist
PCA	Personal Care Aide
ProSA-RA	Process that Systematizes the Architectural design of Reference Architectures
PT	Physical Therapist

QA	Quality Attribute
QM	Quality Model
QM4AAL	Quality Model for Ambient Assisted Living
QoL	Quality of Life
QoS	Quality of Service
RA	Reference Architecture
RAModel	Reference Architecture Model
RD	Registered Dietician
RM	Reference Model
RN	Registered Nurse
S3	Service-Oriented Solution Stack
SaS	Self-adaptive Systems
SLP	Speech-Language Pathologist
SNOMED CT	Systematised Nomenclature of Medicine - Clinical Terms
SOA	Service Oriented Architecture
SoaML	Service Oriented Architecture Modelling Language
SoS	Systems-of-Systems
SPL	Software Product Line
SPLA	Software Product Line Architecture
SysML	Systems Modelling Language
UML	Unified Modelling Language
XML	Extensible Markup Language

CONTENTS

CHAPTER

1

INTRODUCTION

As stated by the United Nations, world population aged 60 or over is increasing faster than younger age groups (United Nations, 2017). The population ageing phenomenon is caused by decreasing of birth rates and increasing of life expectancy. Despite populations in many countries are still considered young, in 2017, in all world, 962 million people aged 60 or over embracing 13 % of the global population. Moreover, it is estimated that in 2050, all countries (excluding those in Africa) will have at least a quarter of their population at ages 60 and above, that is, in 33 years from now 2.1 billion people will be aged 60 or over (United Nations, 2017). This ageing phenomenon will lead to a growing number of older people living alone and in need of intensive care, and in a rapid growth in the number of people with physical disabilities. Moreover, the challenge of bringing care and assistance to those people will become more and more important from both a social and an economic point of view. Additionally, ageing phenomenon will lead to dramatic challenges for health care and care systems, state pensions schemes, and employers alike (AALIANCE, 2010).

Furthermore, elderly people have a high risk of suffering of chronic conditions[1] in countries throughout the world, regardless of income level. Specifically, about 91% of older adults have a least one chronic condition (National Council on Aging, 2014), i.e., human diseases that persists for a long time, requiring ongoing medical attention and/or limiting activities of daily living. Chronic conditions can include both physical conditions (e.g., arthritis, cancer, and HIV infection), and mental and cognitive disorders (e.g., ongoing depression, substance addiction, and dementia) (U.S. Department of Health and Human Services., 2014). Additionally, 73% of older adults in USA and 50 million people in European Union live with multiple chronic conditions (i.e., two or more chronic illnesses at the same time), deeply impacting on their quality of life (STRUCKMANN *et al.*, 2013; National Council on Aging, 2014). Figure 1 shows the presence of co-morbidity among chronic conditions in USA (Centers for medicare and medicaid services, 2012). For example, from Figure 1, it is possible to infer that *stroke* and *heart failure*

[1] Also referred as chronic diseases, chronic illnesses, or non-communicable diseases (NCDs)

are high co-morbid conditions, due to about 55% of persons who present such chronic conditions, also have five or more additional chronic health conditions.

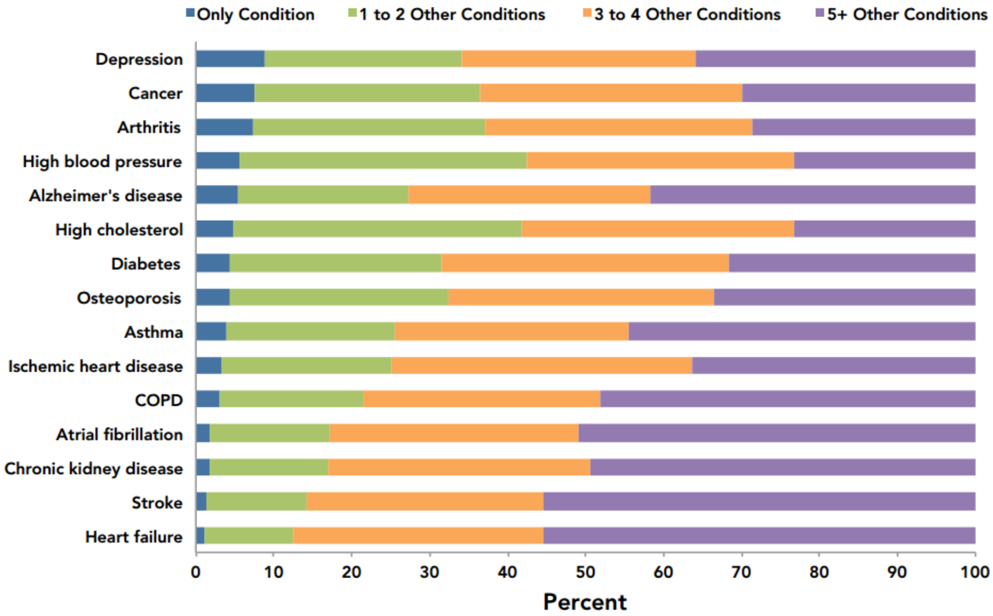


Figure 1 – Co-morbidity among chronic conditions. Source: (Centers for medicare and medicaid services, 2012)

Multiple chronic conditions also place a significant burden on financial and human resources. In addition, increasing healthcare expenditures and shortages, as well as disparities in the supply of health professionals, raise concerns about health system sustainability in many countries. This is supported by the fact that about 70% to 80% of healthcare costs are spent on chronic diseases, which correspond to €700 billion in the European Union (STRUCKMANN *et al.*, 2013). In this context, an alternative to reduce costs, associated with chronic disease management, is introducing the concept of “patient–centered”, which aims to bring the patients care from the hospital to their homes, and consequently, improving their quality of life (WARTENA *et al.*, 2010). Figure 2 depicts how the costs of care by day decrease, according as quality of life increase. In this perspective, in the last 20 years, it has augmented the interest in developing devices, platforms, technologies and services that bring healthcare assistance at home to elders (and in general to all patients) improving thus, their quality of life in an economical way.

Aiming at enhancing the quality of life for everyone, the Ambient Assisted Living (AAL) concept emerged in the 1990s, and since the middle of the 2000s, it has received more attention. AAL is a relatively new field and has become an increasingly important as a multidisciplinary research topic for both medical and technological research communities. AAL refers to concepts, products, and services, improving autonomy/independence, comfort, safety, security, and health, for everyone (with a focus on elderly persons) in all stages of their life (AALIANCE, 2010). AAL is primarily concerned with the individual in his or her immediate environment (e.g., home or work) by offering user-friendly interfaces for all sorts of equipments

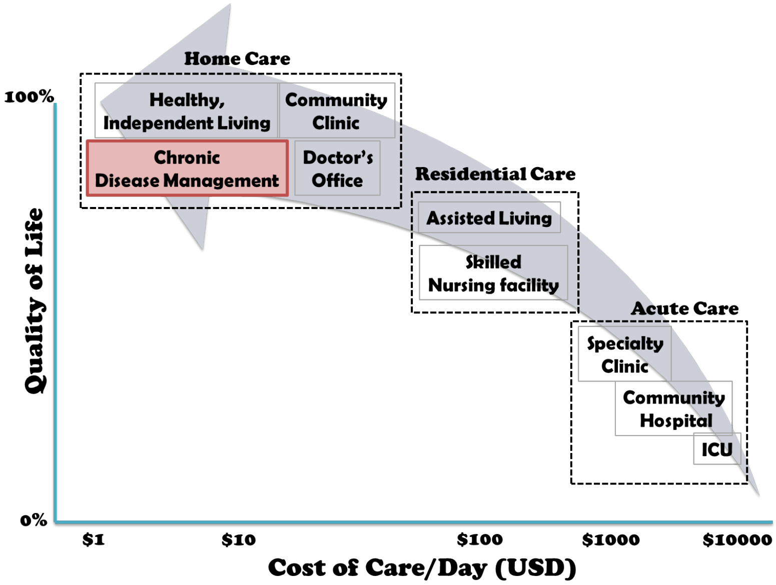


Figure 2 – Reducing costs and improving quality of life. Adapted from: (WARTENA *et al.*, 2010)

in the home and outside, taking into account that many older people have impairments in vision, hearing, mobility, or dexterity (PIEPER; ANTONA; CORTÉS, 2011). To achieve these goals, AAL interlinks, improves, and proposes new technologies and combines ICT (Information and Communication Technologies) and social environments. In this perspective, AAL can also refer to intelligent systems of assistance (or age-based assistance systems) (AALIANCE, 2010; PIEPER; ANTONA; CORTÉS, 2011). In this perspective, various AAL systems have been developed in recent years. Representative examples of AAL systems are systems for activity monitoring, systems for detection of situations of helplessness, and systems for remotely tracking vital signs (e.g., heart rate, pulse, and blood pressure). Furthermore, a range of AAL platforms have been developed in the last years, aimed at supporting and facilitating the development of these systems (ANTONINO *et al.*, 2011; MEMON *et al.*, 2014).

Healthcare Supportive Homes - HSH (or Health Smart Homes), are a special type of AAL system, that aim to support patients diseases management, monitor and improve patients health status, and to bring assistance when emergency situations occur (MAEDER; WILLIAMS, 2017). Specifically, a HSH intends to offer an autonomous life, in their residence, to people suffering from various pathologies and handicaps that should normally force them into a hospitalization or placement in specialized structures. (RIALLE *et al.*, 2002; NOURY *et al.*, 2003; MAEDER; WILLIAMS, 2017; MAJUMDER *et al.*, 2017). As stated in (AALIANCE, 2010), it is expected that a HSH can be: a) personalizable, i.e., tailored to the users' needs; b) adaptive, i.e., capability to react to the dynamic changes in device/service availability, resource availability, system environment, or user requirements; and c) anticipatory, i.e., anticipating users' desires or situations as far as possible without conscious mediation. Furthermore, these systems must also be non-invasive or invisible, distributed throughout the environment or directly integrated into appliances or furniture. Additionally, according to EVAAL [2], such systems must present the following core functionalities: (a) Sensing: capability of collecting information from any relevant place (e.g., in-/on-body and in-/on-appliance), or environment (e.g., home, outdoor, vehicles, and public spaces); (b) Reasoning: aggregation, processing, and analysis of data to either infer new data or deduce actions to be performed; (c) Acting: automatic control of the environment through actuators; (d) Communicating: communications among sensors, external systems, reasoning systems, and actuators, where all these components can be connected dynamically; and (d) Interacting: interaction between human users and systems by means of personalized interfaces. Moreover, HSH systems must collaborate with external e-Health systems to establish a fully understanding about patients health history, and to offer continuous monitoring of patients conditions (MAEDER; WILLIAMS, 2017).

Considering the aforementioned panorama, it is possible to define HSH systems as Systems-of-Systems (SoS), that is, systems whose constituent elements are also considered systems distributed over patient's home, and presenting operational and managerial independence from a central entity (e.g., health services providers). Examples of constituents systems of a HSH system are wearable biomedical sensors, activity detection systems, sleep monitoring systems, environment monitoring systems, home security systems, home energy management systems, home automation systems, and companion robots (MAJUMDER *et al.*, 2017). These constituents must collaborate among them to accomplish one or more global missions, i.e., complex activities that can not be addressed by any constituent individually. For instance, a mission of a HSH system can be *to continuously offer an entire panorama about patient's health status alerting possible emergency situations*. To accomplish this mission, behaviours of HSH systems must allow the identification symptoms and emergency situations, and execution of preventive or corrective plans in a reliable way. However, such behaviours can not be executed by particular systems (or parts of them), but they are possible through interactions between constituent systems, and between such systems and the HSH system.

1.1. Problem Statement

Nowadays, HSH systems have become important for supporting daily life at home of patients suffering of chronic conditions, since most patients prefer to be treated at home and to remain active and independent for as long as possible. In the study conducted in (BAL *et al.*, 2011), it was concluded that nearly 90% of interviewed seniors said that remaining in their home was very important to them for managing their diseases. Moreover, the importance of HSH systems is noticed in the amount of projects finalized that were funded by the European Commission (i.e., Framework Programme 7[3]) and the AAL Joint Programme[4] in the last years,

[2] Evaluating AAL systems through competitive benchmarking - <http://evaal.aaloa.org/>
[3] <http://ec.europa.eu/research/health/index.cfm?pg=projects>
[4] <http://www.aal-europe.eu/>

i.e., more than 18 projects involving multiple international consortia, in the area of smart home systems, offering tele-monitoring services to assist chronic conditions management, such as, chronic obstructive pulmonary disease, renal insufficiency, stroke, heart failure, mental decline diseases, Parkinson's disease, and Alzheimer's disease.

Despite the variety of offered solutions for chronic disease management, there are important issues that such systems present (MAEDER; WILLIAMS, 2017; MAJUMDER *et al.*, 2017). Existing HSH systems are sometimes proprietary, monolithic, high coupling, and expensive solutions for patients and their relatives. Most of such systems do not consider their interoperation with existing, distributed, and external systems, such as Electronic Health Records (EHR), Patient Health Records (PHR), emergency systems (e.g., ambulances, fire departments), and other Health Information Systems (HIS). The majority of existing HSH systems do not contemplate interoperation with other systems running inside the home, such as domotic systems, company robots or activity monitors provided by different companies. In the same perspective, most of the individual HSH solutions have a limited view of patient problems, since they are oriented to manage one disease or specific diseases but do not consider the overall patient health profile, e.g., a patient suffering of dementia and hearth failure must acquire two different systems to manage both conditions at home, increasing the cost of her/his treatment. Moreover, current HSH systems are difficult to evolve regarding patients health profile changes, e.g., new conditions can appear or disappear and system services must be added or deactivated. Finally, existing HSH systems are region focused, limiting their use to those regions, e.g., systems designed considering local legislations, health system configurations (e.g., public, private or mixed), care plan protocols, and technological settings. In this perspective, reuse of those systems in other contexts is limited.

Aiming for guidelines to develop AAL systems and e-Health systems (e.g., EHR or HIS), several reference architectures have been proposed (GARCÉS *et al.*, 2017b; GARCÉS *et al.*, 2015). Examples of those architectures are UniversAAL[5], OASIS (KEHAGIAS *et al.*, 2010), Continua (WARTENA *et al.*, 2010), PERSONA (TAZARI *et al.*, 2010a). In short, a reference architecture refers to a software architecture that encompasses the knowledge about how to design concrete architectures of systems of a given application domain (NAKAGAWA; OQUENDO; MALDONADO, 2014). Some benefits at adopting reference architectures in software development are (NAKAGAWA; OQUENDO; MALDONADO, 2014; ANGELOV; TRIENEKENS; KUSTERS, 2013; MARTÍNEZ-FERNÁNDEZ *et al.*, 2013): (i) positive impact in project team productivity, since architectural knowledge is reused; (ii) standardized solution for software systems in a domain bringing a solution for systems interoperability; (iii) support for evolution of software systems; and (iv) improvement in the quality of software systems.

In spite of the existence of reference architectures to orient the development of interoperable, platform independent, and standardized AAL systems, none of those reference architectures

[5] <http://universaal.sintef9013.com/index.php/en/>

is oriented to support the self-management of multiple chronic-conditions neither consider interventions at home to assist patients when a health critical situation occurred (GARCÉS *et al.*, 2017b). Moreover, few of such architectures contemplate their interoperation with other solutions of smart homes, wireless or body sensor networks, assistance robots and HIS. Reference architectures proposed for the AAL domain are too abstract that is required a high learning curve to be used in the development of specific HSH solutions (GARCÉS *et al.*, 2017b).

Finally, the problem addressed in this PhD project is the lack of guidelines to orient the development of interoperable, standardized, adaptive, evolutionary, reusable HSH systems for supporting the continuous monitoring, at home, of patients suffering of chronic conditions.

1.2. Research Questions and Objectives

Considering the challenges associated at developing HSH systems, mentioned in Section 1.1, the main objective of this thesis is to establish a reference architecture, named *HomecARe*, that offers guidelines for the development of software systems for supporting chronic diseases management at home, promoting their quality, interoperability with existing e-Health systems, low coupling, and reusability.

Research Question: It is intended to investigate with this thesis whether or not: *Is HomecARe a suitable approach to address the challenges found at developing HSH systems?*.

Specific Objectives: In order to accomplish the general objective and to answer the research question, the following specific objectives were defined:

- *Establishment of domain models:* It was investigated information sources that describe knowledge needed to understand the domain of HSH systems. Based on such knowledge, models representing missions and qualities of HSH systems were proposed. Such models conform *HomecARe* allowing it to offer well-established domain knowledge that can be used as knowledge repositories, fomenting its reuse in the development of HSH systems;

- *Establishment of the reference architecture for HSH systems:* It was defined the architectural significant requirements (functional and non-functional) for *HomecARe*, and the design decisions that allow to achieve such requirements. Semi-formal modeling languages (e.g, UML and SoaML) were used to represent the architectural decisions contained in the reference architecture to facilitate its understanding and reusability when concrete HSH systems are constructed. Moreover, to improve the usability of knowledge provided by *HomecARe*, architectural viewpoints (i.e., conceptual, missions, and quality viewpoints) and views were selected to document the rational behind taken decisions;

- *Definition of guidelines to instantiate HomecARe:* It was established orientations on how to use *HomecARe* to design software architectures for HSH systems. Those guidelines facilitate the correct use of this reference architecture, avoiding misunderstandings when such architecture is instantiated; and

- *Evaluation of HomecARe:* The proposed reference architecture was evaluated through, the conduction of a case study, to obtain evidences that allow to answer the research question and discover improvements to be done.

Aiming accomplish the general objective and answer the research question proposed for this thesis, ProSA-RA (NAKAGAWA *et al.*, 2014) was followed. ProSA-RA, is a four-step process for the building of reference architectures, focusing on how to design, represent, and evaluate such architectures. ProSA-RA is detailed in Chapter 2.

1.3. Contributions

The present thesis contributes to the areas of AAL, health care, software architecture and systems-of-systems. Regarding the AAL domain, an identification of reference models and architectures was made. Moreover, quality attributes and a further establishment of a quality model was made. Hence, AAL software architects can reuse such knowledge to select the most adequate reference and to identify important qualities for their projects. For the healthcare domain, are offered guidelines to construct HSH systems that give support to the chronic disease management of patients at home. HSH systems can be designed achieving qualities as interoperability, reusability, security, safety, performance and reliability. Moreover, such systems can be defined under a patient centered perspective, since patients profiles can be used to configure them to achieve patients needs. Furthermore, a generic health service bus is defined. Such bus intends solve interoperability problems among systems dealing with health information. Thus, this bus can be reused in healthcare systems different from those specified in this thesis. In a perspective of software architecture, this thesis supplies evidence of the applicability of ProSA-RA for their consolidation. Similarly, architectural decisions (e.g., patterns or styles) selected to conform the reference architecture, can be reused in other domains presenting similarities in quality attributes requirements. Respecting, systems-of-systems domain, this thesis evidences the importance of reference architectures to guide the systematic development of those systems in specific domains, since they offer a broader perspective of the solution space containing well defined domain knowledge and architectural decisions. Specifically, this thesis subsidizes a process for the establishment, modelling and validation of missions of SoS in reference architectures.

1.4. Thesis Outline

This thesis monograph is structured as follows. Chapter 2 presents the state of the art regarding reference architectures and system-of-systems. The HSH domain is detailed in Chapter

3, which describes stakeholders, characteristics of HSH systems, interoperability standards in e-Health systems, and the relevance of quality systems for the domain. The architectural analysis of *HomecARe* is detailed in Chapter 4. Different types of domain models, namely, missions, constituents operational and communicational capabilities, data entities, and emergent behaviours are presented in this chapter, as well as, the architectural significant requirements of this reference architecture. Results of the architectural synthesis of *HomecARe* are described in Chapter 5. Three viewpoints compose this chapter: (i) Conceptual viewpoint, which details all architectural elements of *HomecARe*, such as, services offered by constituents systems, services for controlling and executing business processes, consumer services, the Health Service Bus (HSB), and a variety of repositories; (ii) Missions viewpoint, which describes the different processes and services architectures that HSH systems can configure to accomplish their missions, as well as, services interfaces, contracts and protocols to address interoperability issues; and (iii) Quality viewpoint, which presents architectural decisions made to satisfy quality requirements related to interoperability, reliability, security, and adaptivity. The evaluation of *HomecARe* is presented in Chapter 6. For this, a case study investigating the viability of using *HomecARe* to design the software architecture of a HSH system was planned and conducted. This chapter also discusses evaluation results and threats to validity. Finally, contributions, limitations, future works, and extensions of this thesis are discussed in Chapter 7.

CHAPTER

2

STATE OF THE ART

In this chapter, the theoretical background containing the main topics embraced in this thesis, namely, reference architecture, and Systems-of-Systems (SoSs) is given. The HSH systems topic was omitted from this chapter, since it is described in depth in Chapter 3.Section 2.1 details reference architectures and related concepts, such as software architecture, architectural styles and patterns, reference models, and product line architectures. Moreover Section 2.1 covers approaches for the engineering of reference architectures, and details ProSA-RA, the process followed in this thesis to design *HomecARe*. SoSs are described in Section 2.2, highlighting their characteristics, possible architectural solutions, and the concept of mediators. Section 2.3 presents the state of the art of existing reference architectures in areas related to HSH systems, namely, AAL and e-Health. Finally, some considerations about how the background presented in this chapter supported the engineering of *HomecARe*, are given in Section 2.4.

2.1. Reference Architecture

Software architecture researchers started to define reference architectures at the beginning of this century. One of the first definition was given by Kruchten who establishes that a reference architecture is *"a predefined architectural pattern, or set of patterns, possibly partially or completely instantiated, designed and proven for use in particular business and technical contexts, together with supporting artefacts to enable their use. Often, these artefacts are harvested from previous projects"* (KRUCHTEN, 2000). This concept was supported and enhanced by Reed, who in turn suggests that *"a reference architecture consists of information accessible to all project team members that provides a consistent set of architectural best practices"* (REED, 2002). However, other researchers do not consider architectural patterns in their definitions but high-level software elements. In this perspective, Garland and Anthony set that a reference architecture *"describes the high-level set of elements involved in applications from a particular domain along with their interactions"* (GARLAND; ANTHONY, 2003), and Bass et al. establish that *"a reference architecture mapped functionalities onto software elements (that cooperatively implement the functionality) and the data flows between them"* (BASS; CLEMENTS; KAZMAN, 2003). It is possible to observe that the last two definitions state that a reference architecture must to include the interchange of data between elements in the reference architecture. Moreover, it is possible to find definitions from an enterprise point of view. In this context, Rosen et al. define a reference architecture as a *''working example of a critical aspect of your enterprise architecture, such as (...) how to work with your organization's message bus or (...) how to work with your business rules engine"* (ROSEN *et al.*, 2007).

Furthermore, Angelov et al. set that a reference architecture is *"a generic architecture for a class of information systems that is used as a foundation for the design of concrete architectures from this class"* (ANGELOV; GREFEN; GREEFHORST, 2009), establishing thus that a reference architecture is a special type of software architecture instead of an architectural pattern or sets of software elements, as presented in earlier definitions. Moreover, Muller establishes that a *"reference architecture is created by capturing the essentials of existing architectures and by taking into account future needs and opportunities, ranging from specific technologies, to patterns to business models and market segments"* (MULLER, 2012). Additionally, Muller describes that the *"purpose of a reference architecture is to provide guidance for the development of architectures for new versions of the system or extended systems and product families"* (MULLER, 2012). An important remark of Muller's definition is that the author proposes that a reference architecture should address technical and business architectures as well as the customer context. Besides, Estefan et al. establish that a *"reference architecture models the abstract architectural elements in the domain of interest independent of the technologies, protocols, and products that are used to implement a specific solution for the domain"* (ESTEFAN *et al.*, 2012).

Even though the aforementioned definitions are different, they present the same essence: the reuse of knowledge about software development in a given domain, in particular, with regard to architectural design (NAKAGAWA; OQUENDO; MALDONADO, 2014). In this perspective, Nakagawa et al. state the following definition for reference architecture, that in turn is the adopted in this thesis: *"A reference architecture refers to an architecture that encompasses the knowledge about how to design concrete architectures of systems of a given application domain; therefore, it must address the business rules, architectural styles (sometimes also defined as architectural patterns that can also address quality attributes in the reference architecture), best practices of software development (for instance, architectural decisions, domain constraints, legislation, and standards), and the software elements that support development of systems for that domain. All of this must be supported by a unified, unambiguous, and widely understood domain terminology"* (NAKAGAWA; OQUENDO; MALDONADO, 2014).

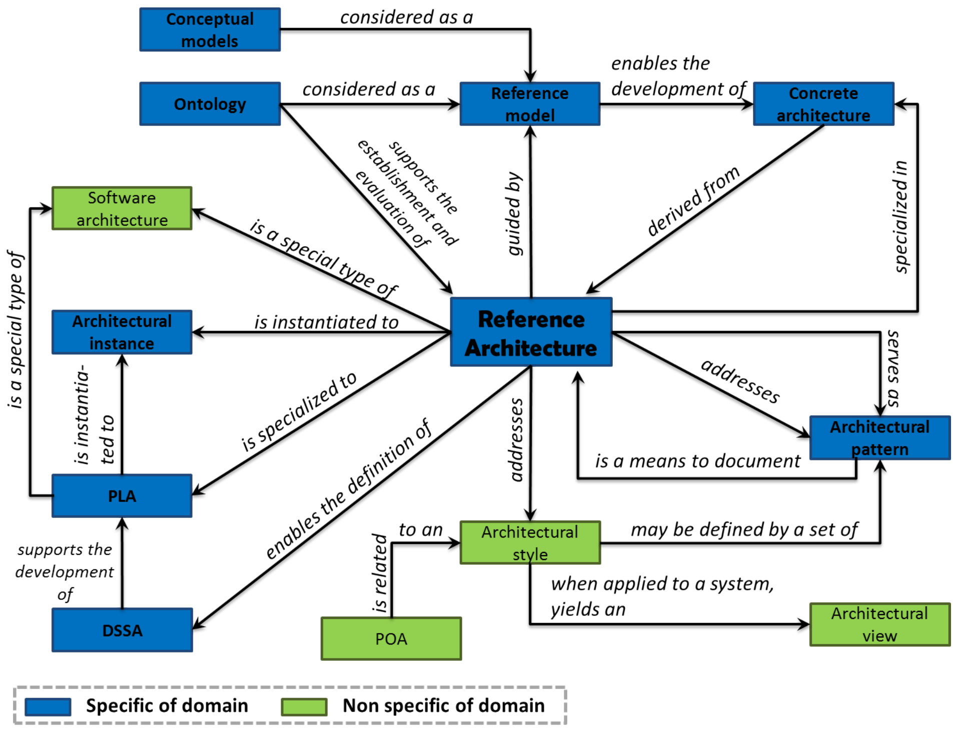


Figure 3 – Conceptual diagram of reference architecture's related terminology

Related Concepts to Reference Architecture

The term reference architecture has been sometimes used interchanged with other terms, such as, reference model, platform-oriented architecture (POA), product line architecture (PLA), or domain-specific software architecture (DSSA). Moreover, other concepts, e.g., architectural patterns, architectural styles, concrete architecture, and ontologies, are somehow related to reference architecture. Figure 3 shows a conceptual diagram that summarizes relationships among those concepts. Related concepts are detailed in the remainder of this section.

- **Reference Models:** are considered abstract frameworks whose purpose is domain modelling, representing relationships between domain entities. Reference models can vary from simple conceptual model to well-established ontologies (NAKAGAWA; BARBOSA; MALDONADO, 2009; NAKAGAWA; OQUENDO; MALDONADO, 2014). These models are independent of specific standards, technologies, implementations, or other concrete details (MACKENZIE *et al.*, 2006). Reference models can support the establishment of reference architectures, since concepts and functionalities defined in these models can be mapped into software architecture entities (BASS; CLEMENTS; KAZMAN, 2003).
- **Architectural styles:** are defined as a "specialization of element and relation types, together with a set of constraints on how they can be used" (BASS; CLEMENTS; KAZMAN, 2003). A style reduces the sets of possible forms and imposes a certain degree of uniformity on the architecture. An style may be defined by the selection of an architectural framework,

by a middleware, by a recommended set of patterns, or by an architecture description technique or tool (KRUCHTEN, 2003). Some examples of styles are module, component and connector (C&C) and allocation styles as presented in (GARLAN *et al.*, 2010).

- **Architectural patterns:** Architectural patterns and styles are, sometimes used as similar concept. An architectural pattern "describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven architecture designs for its solution" (BUSCHMANN *et al.*, 1996). Architecture designs are specified by describing its constituent components, their responsibilities and relations, and the ways in which they collaborate" (BUSCHMANN *et al.*, 1996). The difference between architectural pattern and style is that, the former suggests an architecture design based on the problem and the context, whilst, the later focuses on the architectural design, with more lightweight guidance on when a particular style may or may not be useful (GARLAN *et al.*, 2010). Moreover, Cloutier et al. establish the relation between architectural patterns and reference architectures in twofold. Firstly, architectural patterns are one of the inputs into a reference architecture, and one of the means to document such architectures; and, lastly, the reference architecture serves as an architectural pattern for future architectures in the specific domain (CLOUTIER *et al.*, 2010).

- **Domain-Specific Software Architecture (DSSA):** It is considered as a process (ARMITAGE, 1993; TRACZ, 1995; HAYES-ROTH *et al.*, 1995) to support the development of domain models, reference requirements, and reference architectures for a family of applications within a particular problem domain (also known as a product-line) (TRACZ, 1995). An important element for the development and use of DSSAs is the *DSSA library*. This library contains domain-specific software assets for reuse in the DSSA process, and its purpose is to control component version (ARMITAGE, 1993). DSSA and reference architecture definitions can be related in the sense that, a reference architecture for a specific domain is an input for developing DSSAs in such domain. In this context, it is possible to infer that a DSSA is more specific than a reference architecture, since DSSA includes selection of operating system, middleware, and data persistence services (BOSCH; RAN, 2000), which are not considered in reference architectures.

- **Product-Line Architecture (PLA):** It is a special type of software architecture used to build a product line, explicitly describing commonality and variability. Moreover, a PLA is the basis for the architecture of all product line members (NAKAGAWA; OLIVEIRA; BECKER, 2011). PLAs differs form reference architectures, since the last deal with the range of knowledge of an application domain, providing standardized solutions for a broader domain, whilst, PLAs are more specialized, focusing sometimes on a specific subset of the software systems of a domain and providing standardized solutions for a smaller family of systems (NAKAGAWA; OLIVEIRA; BECKER, 2011). Another essential difference is that PLAs are concerned with the variabilities among products. Furthermore, reference architectures are generally on a higher level of abstraction compared to PLAs (NAKAGAWA; OQUENDO; MALDONADO, 2014).

- **Architectural Instance and System Architecture:** As defined by Garlan and Perry, an architectural instance refers to the architecture of a specific system (GARLAN; PERRY, 1994). Shortly, a system architecture is a means for describing the elements and interactions of a complete system including its hardware and software elements (Software Engineering Institute, 2014). In this perspective, a reference architecture is much broader in scope than an architectural instance, contributes to communication effectiveness, and provides guidance for future architecture instantiations (CLOUTIER *et al.*, 2010).

- **Software Architecture:** A software architecture is defined by Bass et al. as " the structure or structures of the system, which comprise software elements (e.g., services, components, modules), the externally visible properties of those elements (i.e., behaviour of each element), and the relationships among them" (BASS; CLEMENTS; KAZMAN, 2003). An architecture can be documented in an architectural description (ISO/IEC/IEEE, 2011), which in turn, is also denominated as concrete architecture (ANGELOV; TRIENEKENS; GREFEN, 2008). Moreover, software architectures play a fundamental role in determining the system quality (e.g., performance, portability, and maintainability). Decisions made at the architectural level directly enable, facilitate, or interfere with the achievement of business goals as well as functional and quality requirements (NAKAGAWA; OQUENDO; BECKER, 2012). In this context, reference architectures refer to a special type of software architecture that captures the essence of the architectures of a set of software systems of a given domain (NAKAGAWA; OQUENDO; BECKER, 2012).

- **Platform-oriented architecture (POA):** POAs have also been proposed and widely used as reference architectures. However, POAs are not related to the specific application domain, but to a specific architectural style or technology (NAKAGAWA; OQUENDO; BECKER, 2012). An example is OASIS (ESTEFAN *et al.*, 2012), a POA related to Service Oriented Architecture (SOA) style.

Reference Architecture Engineering

The use of systematic process to build reference architectures can improve their effectiveness to communicate their solutions, and to achieve their purposes. Nowadays, it is possible to found several initiatives (BAYER *et al.*, 2004; DOBRICA; NIEMELA, 2008; MULLER; LAAR, 2008; GALSTER; AVGERIOU, 2011; CLOUTIER *et al.*, 2010; TRIENEKENS *et al.*, 2011; ANGELOV; GREFEN; GREEFHORST, 2012) oriented to provide processes, guidelines, principles, and recommendations to enhance the engineering of reference architectures.

Muller and Laar (2008) propose a set of recommendations in order to create, update and maintain understandable reference architectures. Trienekens et al. (2011) establish a set of quality attributes for reference architectures (i.e., completeness, acceptability, buildability, applicability, and understandability) that could be considered during their development. Angelov et al. (2012) provide a framework for analysing, classifying, and designing successful reference architectures, considering congruence among their context, goals, and specification. Dobrica and Niemela (2008) define an approach for designing reference architectures as basis to establish product-lines in the embedded systems domain. Similarly, Bayer et al. (2004) establish PuLSE-DSSA, where reference architectures are created by capturing knowledge from existing PLAs. Galster and Avgeriou (2011) provide a six-steps procedure to design reference architectures based on empirical methods to create their foundations and to validate such architectures.

ProSA-RA

Nakagawa et al.(2014) define ProSA-RA, a process for the building of reference architectures, proposing methods to design, represent, and evaluate these architectures. ProSA-RA is the result of the experience in the establishment of architectures in domains of software engineering (NAKAGAWA *et al.*, 2011; NAKAGAWA *et al.*, 2007; OLIVEIRA; NAKAGAWA, 2011) and embedded systems (BORG, 2011; DUARTE, 2012; FEITOSA, 2013). In this thesis, ProSA-RA was selected as the method to develop *HomecARe* , therefore, it is detailed as follows.

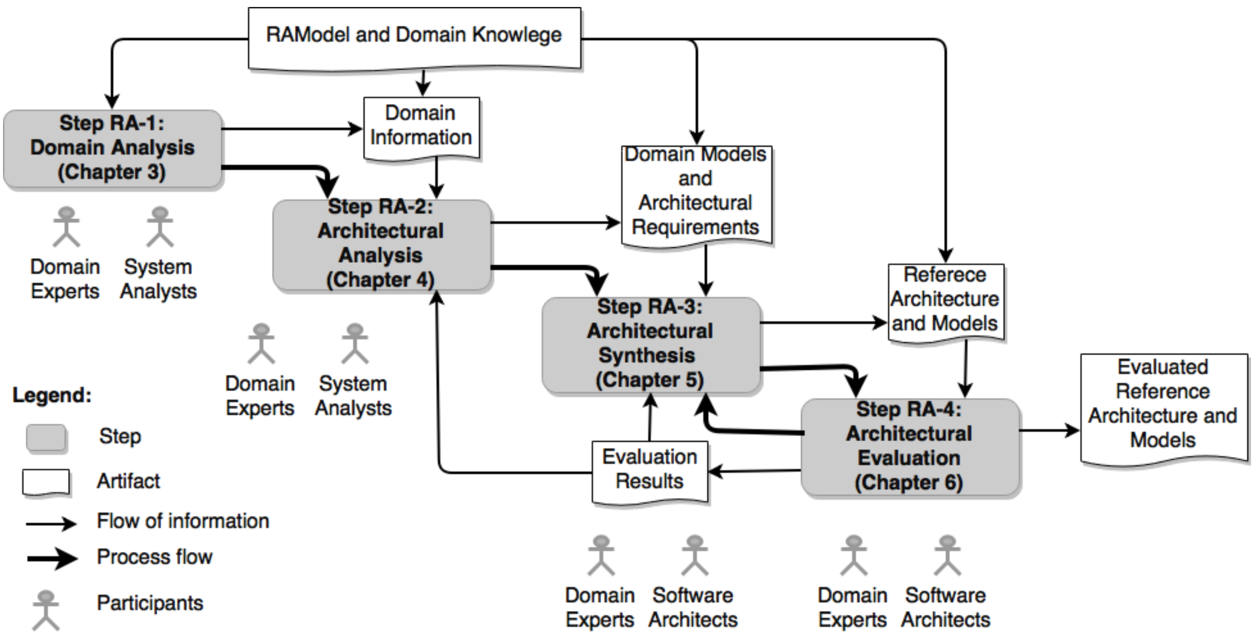


Figure 4 – Process for the construction of reference architectures - ProSA-RA. Adapted from (NAKAGAWA *et al.*, 2014)

As depicted in Figure 4, ProSA-RA comprises four steps. The purpose of the first step - **RA-1** - is to obtain the most understanding as possible of the domain for which a reference architecture is being defined, for that, it is important to establish and bound the scope of the domain.

Hence, information sources providing knowledge on business processes, stakeholders needs, development of software systems, concepts, legislations and standards for the domain, among others, are identified and analysed. As an output artefact, a base of domain information is defined. In the step 2 - **RA-2** - is aimed to identify requirements of software systems and define the requirements (functional and non-functional) for the reference architecture, i.e., the architecturally significant requirements. Models representing domain knowledge can be defined with the objective of organize and share such knowledge among stakeholders. Domain models and architectural requirements are the inputs of step 3 - **RA-3** - that intends to formalize architectural solutions, selecting architectural decisions documenting them in viewpoints and views, and representing such decisions using formal or semi-formal modelling languages, such as UML or SysML. Several viewpoints, such as logical, runtime, data, and physical viewpoints, proposed for the description of software architectures (KRUCHTEN, 2000) can be adopted to describe reference architectures (GUESSI; BUENO; NAKAGAWA, 2011). Both reference architecture and domain models are assessed in step 4 - **RA-4** - through the conduction of surveys with experts, case studies, or scenario based techniques. Finally, issues found in the evaluation step must be resolved revising models, architectural decisions, their representations and documentation.

RAModel

The four steps defined in ProSA-RA are supported by RAModel (Reference Architecture Model), a reference model for reference architectures, providing information on possibly all elements (and their relationships) that could be contained in reference architectures, independently from application domains or purpose of such architectures (NAKAGAWA; OQUENDO; BECKER, 2012). In short, RAModel is composed of four groups of elements:

- **Domain group:** It contains elements related to self-contained, specific information of the space of human action in the real world, such as domain legislations, standards, and certification processes, which impact systems and related reference architectures;
- **Application group:** It contains elements that provide a good understanding of the reference architecture, its capabilities and limitations. It also contains elements related to the business rules (or functionalities) that can be present in software systems built from the reference architecture;
- **Infrastructure group:** It refers to elements that can be used to build the software systems based on the reference architecture. These elements are responsible for enabling these systems to auto-mate, for instance, processes, activities, and tasks of a given domain; and
- **Crosscutting group:** It aggregates a set of elements that are usually spread across and/or tangled with elements of other three groups (domain, application, and infrastructure). We have observed that communication (that we have identified as internal and external) in the

software systems built from the reference architecture, as well as the domain terminology and decisions are present in a spread and tangled way when describing other groups and are, therefore, crosscutting elements.

2.2. System-of-Systems

Current software systems have become large and complex, principally due to the diversity of stakeholders, multidisciplinary practitioners, heterogeneous technologies, ubiquity, and undetermined requirements and behaviours. In this perspective, a class of systems has risen, known as Systems-of-Systems (SoSs). An SoS is a system whose constituent elements are systems themselves. Constituent systems collaborate among them to achieve high-level missions that can not be addressed by any system independently. In short, missions are systems activities to pursue stakeholders goals (BEALE; BONOMETTI, 2006). For instance, one mission for the Apollo 12 system is (BEALE; BONOMETTI, 2006): “to perform inspection, survey, and sampling in lunar mare area”.

A system can be considered an SoS if presents the following characteristics (NIELSEN *et al.*, 2015; MAIER, 1999):

- *Operational independence:* Constituent systems are independent and able to operate even when the SoS is disassembled; hence, constituents must be low coupled without prejudicing operations of their peers;

- *Managerial independence:* Constituent systems are governed by their own rules rather than by external ones when they are participating of an SoS. This characteristic rises challenges to create quality SoSs, since there is an uncertainty about how reliable are operations offered by those systems, and to what extend they can affect negatively or positively SoS behaviours, and hence the achievement of its missions;

- *Distribution:* Constituent systems are dispersed; hence, it is required a type of mediator to allow the communication of their operations results to other constituents and, depending of the type of SoS, to the central authority. The mediator must allow the communication, coordination, and (if required) translation of messages to allow inter-operations between constituents and the SoS;

- *Evolutionary development:* SoSs can be under constant change due to missions modifications, e.g., market tendencies can change business strategies, resulting in changes of missions, and hence, requiring the inclusion of new constituents or removal of those that are not more longer required.

- *Emergent behaviour:* An SoS behaviour emerges as a result of the synergistic collaboration of its constituent systems. An emergent behaviour allows to address a global mission than those individual missions accomplished by the constituent systems separately. Depending on the complexity to characterize, measure, and predict emergent behaviours, an SoS can be classified as deterministic or stochastic (MAIER, 1999). Deterministic SoSs present simple or weak emergent behaviours that can be predicted using discrete event system formalisms or control theory approaches (MITTAL; RAINEY, 2015). Stochastic SoSs present strong emergent behaviours that through the use of heuristics or estimation theory can be measured (MITTAL; RAINEY, 2015). When an SoS is deterministic, it is possible to define its emergent behaviours at design time, relating them to the accomplishment of SoS missions before constituent systems self-organize as an SoS (MITTAL; RAINEY, 2015). In runtime, an SoS can present novel emergent behaviours previously not identified, in this case, the SoS lies in an stochastic domain, hence, it is needed to define feedback mechanisms to transform it in a deterministic SoS (MITTAL; RAINEY, 2015).

- *Dynamic reconfiguration:* An SoS can change its structure and composition for ensuring its reliability when faults or unexpected behaviours of their constituent systems occur, or when unanticipated behaviours emerge during SoS operation (sometimes caused by the stochastic nature of the SoS).

SoSs are classified as (OUSD(AT&L), DoD, 2008; DAHMANN; REBOVICH; LANE, 2008): (i) *Directed*, if the constituent systems are controlled by a central authority to satisfy the SoS missions; (ii) *Acknowledged*, if the constituent systems maintain independent management and missions, but collaborate with the SoS to achieve its missions; (iii) *Collaborative*, when the constituent systems are not forced to follow a central management, but voluntarily collaborate to achieve the SoS missions; and (iv) *Virtual*, if the SoS has not clear missions, hence, the SoS behaviours are highly emergent and their constituent systems are difficult to distinguish.

Architectural Solutions for System-of-Systems

Architectural decisions can be reused to construct systems software architectures, involving important choices on software structures (e.g., components and connectors) and the overall system, to satisfy and balance functional and quality requirements (ZIMMERMAN, 2011). These decisions can be captured or documented by architectural patterns (sometimes referred as styles) (HARRISON; AVGERIOU; ZDUN, 2007), which are well proven solutions to recurring problems. Architectural patterns help architects to understand the impact of architectural decisions on quality attributes requirements, because patterns contain information about consequences and context of their usage (HARRISON; AVGERIOU; ZDUN, 2007).

In the context of SoSs, some studies have investigated taxonomies of architectural patterns and styles to address challenges associated to these systems (ROMAY; CUESTA; FERNÁNDEZ-SANZ, 2013; ROTHENHAUS; MICHAEL; T., 2009; INGRAM; PAYNE; FITZGERALD, 2015; GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017), namely, mitigation of the impact caused by changes occurred at constituent systems level, supporting to the understanding of emergent behaviours in runtime, centralization or decentralization of the decision-making authority, separation of concerns of SoS and of their constituents, facilitation of constituent systems collaboration, and adaptability and resilience through SoS dynamic reconfigurations (INGRAM; PAYNE; FITZGERALD, 2015). The remainder of this section details architectural patterns and styles reported in literature that are suitable for designing SoS architectures, and which are of interest for this thesis development.

Centralised Architecture: In a centralized architecture, a central controller, defined as a hub, is responsible for guarantee the correct behaviour and allow the accomplishment of missions of the SoS (INGRAM; PAYNE; FITZGERALD, 2015). Control can be characterized as: (i) fully centralised, when just one central controller is responsible for meeting SoS goals; (ii) hierarchical centralised, when constituent systems act as controllers itself; (iii) hybrid centralised-distributed, if control activities are distributed among different constituent systems and hubs (INGRAM; PAYNE; FITZGERALD, 2015).

Service Oriented Architecture: Service Oriented Architecture (SOA) gives support to deal with distributed business processes. Systems participating in a SOA-based system, execute business activities and are heterogeneous and under control of different owners (JOSUTTIS, 2007). SOA pattern promotes interoperability between distributed systems to accomplish business functionality (services) through the easy integration of their capabilities (JOSUTTIS, 2007). SOA-based system behaviour and performance can be studied by analysing service descriptions (INGRAM; PAYNE; FITZGERALD, 2015). SOA also improves reliable solutions, since capabilities can be offered by multiple providers (e.g., constituent systems), hence, if a provider is unavailable another can replace its participation. SOA minimizes the impact of modifications and failures of the SOA-based systems on participant systems, and vice-versa, due this pattern promotes loose-coupling. Moreover, SOA improves flexibility and horizontal scalability, since new systems or systems capabilities can be added to a SOA solution due to standardized interfaces and pre defined contracts. Vertical scalability is achieved with the coordination or adaptation of services for creating composed and process services, which are high-level services defined to execute complex business activities work-flows.

Composed and process services are designed using orchestration or choreography approaches. In orchestration, a central controller coordinates all process activities. Choreography approach involves collaboration between participants (services), which are responsible for executing one or more activities. In choreography, no central controller exists, hence, collaboration rules must be defined, since services are unaware of activities performed by other participants (JOSUTTIS, 2007). Choreography allows better scalability than orchestration, since control can be distributed among participants, however, it can impact on the system performance, since the full-decentralized control requires to exchange large amounts of information between participants (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017).

Enterprise Service Bus: Enterprise Service Bus (ESB) is the technical backbone of SOA (JOSUTTIS, 2007). The ESB provides connectivity, data transformation (to allow semantic and syntactic interoperability), protocol transformation (allowing technical interoperability), intelligent routing (e.g., using mediators, point-to-point connectors, or interceptors), means to deal with security and reliability of services, service management (adding, removing, deactivating, or updating services), and monitoring and logging services operations (to measure Quality of Service - QoS). In the context of SoS, heterogeneous ESB can be integrated to offer large scale mediation among distributed heterogeneous constituent systems.

Publish-Subscribe Architecture: Publish-Subscribe pattern allows event consumers (subscribers) to be registered for specific events, and event producers to publish (raise) specific events that reach a specified number of consumers. The Publish-Subscribe mechanism is triggered by the event producers and automatically executes a callback-operation to the event consumers. The mechanism thus takes care of decoupling producers and consumers by transmitting events between them (AVGERIOU; ZDUN, 2005).

In the context of SoS, a constituent system can act as a publisher and/or subscriber. This pattern can be divided in data-centric or content-based publish-subscriber pattern with centralized or decentralized control (INGRAM; PAYNE; FITZGERALD, 2015). Some benefits of this pattern are: (i) performance improvement at communicating data among SoS entities, (ii) promotion of modifiability due to low coupling between entities, (iii) encourage dynamic scalability and evolution, since entities can enter or exit without affecting others entities (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017), and (iv) provides resilience, since subscribers can register with multiple publishers for a topic (INGRAM; PAYNE; FITZGERALD, 2015).

Pipe and Filter Architecture: Pipe and filter patterns allows to divide complex task in more easy understood sub-task that are sequentially executed. Each sub-task is implemented as filter that has as responsibility to handle only such task. A filter has a limited amount of inputs and outputs and is connected with other filters though pipes. Filters are unaware of the purpose of their peers. Each filter consumes and delivers data incrementally, which maximizes the throughput of each individual filter, since filters can potentially work in parallel (AVGERIOU; ZDUN, 2005). In the context of SoS, pipe and filters pattern can grant concurrent execution of adaptations by layers, where each layer is seen as a filter (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017).

Trickle-Up Software Pattern: The Trickle-Up pattern is a multi layered pattern that allows separation of concerns for data management. Data object management and fusion logic are independent services that can be binding in runtime to promote control and monitoring of data management (ROTHENHAUS; MICHAEL; T., 2009). This pattern is principally used when environment data are collected and must be further analysed and aggregated to create consolidated reports of a situation. In SoS context, the trickle-up pattern can be used for knowledge aggregation and discovering required to support SoS emergent behaviours.

Reconfiguration Control Architecture: The aim of this pattern is to promote dynamic reconfigurations of software architectures. For this, a reconfiguration control entity (e.g., a constituent system) is responsible for monitoring constituent systems performance and functionality in the form of metadata obtained from each system. Based on such metadata, the controller, making use of reconfiguration policies, determines when a reconfiguration is necessary and what actions must be executed (INGRAM; PAYNE; FITZGERALD, 2015). In SoS, this pattern can be used to avoid degradation of SoS missions due to modifications or unavailability of its constituent systems.

Contract Monitor: This pattern aims to monitor constituent systems interfaces in order to identify possible deviations from its expected functionalities that can prejudice expected SoS emergent behaviours. For using this pattern, it is assumed that constituent systems interfaces are associated to contracts of behaviour, and that composition of such contracts can be correlated to SoS emergent behaviours (INGRAM; PAYNE; FITZGERALD, 2015). Contract monitors can be implemented internally in a constituent system, as an entity, under SoS control, monitoring constituent systems interfaces, or as an external entity monitoring interactions between constituent systems to study emergent behaviours (INGRAM; PAYNE; FITZGERALD, 2015).

Pace Layering: Also named layers of change, is proposed as an initial strategy to design SoS (ROMAY; CUESTA; FERNÁNDEZ-SANZ, 2013). Layers allow the construction of complex behaviours through layers hierarchies, where lower layers implement fast adaptations (i.e., reconfigurations on constituent systems) and higher layers are responsible for time demanding adaptations (i.e., selection of the best policy or plan to achieve SoS missions based on current system status)(GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017; ROMAY; CUESTA; FERNÁNDEZ-SANZ, 2013). Lower layers adaptations aim to achieve performance requirements, and adaptations in higher layers seem to address reliability requirements (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017). Moreover, as layers allow separation of concerns, this property supports maintainability and reusability requirements. Interoperability can also be supported by SoS lower layers at establishing well-defined interfaces of constituent systems.

Evolution Styles: Evolution styles have been proposed to modify software architectures of running systems regarding new requirements, technologies, or to achieve self-* properties, e.g., self-healing, self-organization (CUESTA *et al.*, 2013; GARLAN *et al.*, 2009).

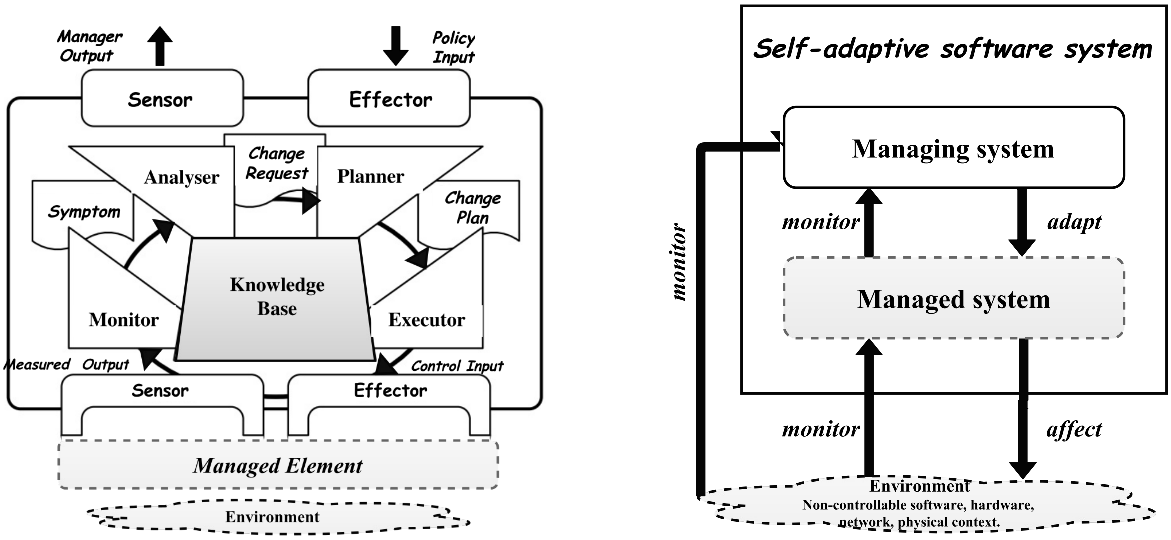
Evolutions styles are composed of (CUESTA; ROMAY, 2010): (i) evolution conditions that allow to identify situations that should be handled through system evolution ; (ii) evolution decisions, that are alternatives made as reaction to an evolution condition; (iii) evolutionary steps that realize the evolution decision; (iv) evolution patterns, that are sequence of evolution decisions, being represented as a decision tree; and (v) evolution styles, that are a set of evolution patterns conceptually related. In SoS context, evolution styles could orient reconfigurations of SoS architectures due to emergent behaviours or modifications in missions to be performed by the SoS.

Reflective Architecture: Reflection is defined as "the capability of a system to rationalize and act upon itself" (CUESTA; FUENTE; BARRIO-SOLáRZANO, 2001; MAES, 1987). A system with reflection capability is composed of two layers. A bottom layer describing system operations and configuration (e.g., components, interfaces, data, interconnections, etc.), and an upper layer embracing an internal meta-model representing how the system perceives and/or modifies itself (CUESTA; FUENTE; BARRIO-SOLáRZANO, 2001). Such model contain all structural an behavioural aspects of the system and is separated from the application logic components (AVGERIOU; ZDUN, 2005). Hence, the system will reflect changes performed in the model, and vice versa through the execution of two basic operations named *reification* and *reflection*. Reification is the action to transfer current system status to make alterations in the meta-model. Reflection is the operation of change system configuration or behaviour regarding modifications in the meta-model. In this perspective, reflection architecture makes the system more flexible, since it can adapt itself to changing conditions (CUESTA; FUENTE; BARRIO-SOLáRZANO, 2001; CUESTA; ROMAY, 2010). Moreover, reflection allows for coping with unforeseen situations automatically (AVGERIOU; ZDUN, 2005). In SoS, the reflection architecture can support evolution and changes in runtime, promoting the anticipation of SoS predicted emergent behaviours and the detection of unforeseen ones.

MAPE-K: Autonomic computing has feedback control loops as first-class entities. The most used one is the Autonomic Manager (IBM, 2006), or MAPE-K loop, which consists of five components, i.e., Monitor, Analyser, Planner, Executor and Knowledge, and two pairs of sensors and actuators interfaces.

The MAPE-K loop is illustrated in Figure 5a. *Sensors* collect information about the managed element (e.g., robot's navigation application). The *monitor* component filters the accumulated sensor data, and stores relevant events in the *knowledge base* for future reference. The *analyser* compares event data against patterns in the knowledge base to diagnose and store symptoms. The *planner* interprets symptoms and devises a plan to execute changes in the managed process through the *effectors* (IBM, 2006; MULLER; KIENLE; STEGE, 2009). An autonomic manager maintains its own knowledge (e.g., information about its current state as well as past states) and has access to knowledge that is shared among collaborating autonomic managers. MAPE-K loops have been used as an architectural solution to design software architectures of Self-adaptive systems (SaS) (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017).

Control in autonomic managers can be centralized, decentralized or fully-decentralized (WEYNS *et al.*, 2013). Decentralization level is defined based on how control decisions in a SaS are coordinated among the MAPE components. In *centralized control*, a single component dedicated to one of the MAPE activity exists. *Decentralized control* is characterized by the existence of multiple components responsible for one of the MAPE activities, i.e., multiple components realizing monitoring activities. In centralized and decentralized control, MAPE components are deployed in a single node. *Fully-decentralized control* is similar to decentralized control, but with multiple MAPE components deployed in multiples nodes (GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017). To realize decentralized and fully-decentralized SaS, an arrangement of collaborating autonomic managers works towards a common goal (MULLER; KIENLE; STEGE, 2009). Arrangements can be done through sensors and effectors, which communicate autonomic manager status to peers and execute input policies sent from peers.



(a) Autonomic manager. Adapted from (IBM, 2006).

(b) SaS constituents. Adapted from (WEYNS; AHMAD, 2013).

Figure 5 – Approaches to develop SaS. Source:(GARCÉS; GRACIANO-NETO; NAKAGAWA, 2017)

Mediators

In an SoS perspective, mediators are architectural elements that connect constituent systems to allow their communication, coordination, cooperation and collaboration (OQUENDO, 2016). Similarly as connectors in software architectures, mediators can be seen as first class entities, specifying their internal behaviours, protocols, duties, commitments and roles, making possible the definition of mediators patterns to be reused by software architects in different SoS projects.

Mediators connectors have been studied in other type of complex systems, such as services based systems in distributed environments. In this context, several types of mediators (some of them considered as patterns) to improve interoperation among heterogeneous components and services have been proposed (LI *et al.*, 2008; SPALAZZESE; INVERARDI; ISSARNY, 2009; ISSARNY; BENNACEUR, 2012; TOMSON; PREDEN, 2013; BENNACEUR; ISSARNY, 2015). However, the application of these mediators connectors in SoSs is limited, since these systems present issues beyond interoperability, e.g., decentralized control, runtime configuration, dynamic environment and architecture, and undetermined behaviours, that must be considered when an SoS architecture is being defined or evolved (GARCÉS; OQUENDO; NAKAGAWA, 2018).

A taxonomy of mediators is proposed, and is presented in detail in Appendix C. This taxonomy aims to aid on the selection of mediators types that will compose the software architecture of an SoS, in both, design and execution time. The taxonomy was established based on a deeper study on software connectors (LOPES; WERMELINGER; FIADEIRO, 2003; LAU; VELASCO; ZHENG, 2005; AMIRAT; OUSSALAH, 2009; KIWELEKAR; JOSHI, 2010; MEHTA; MEDVIDOVIC; PHADKE, 2000) and mediators (LI *et al.*, 2008; SPALAZZESE; INVERARDI; ISSARNY, 2009; ISSARNY; BENNACEUR, 2012; TOMSON; PREDEN, 2013; BENNACEUR; ISSARNY, 2015), and based on knowledge on software architecture and SoSs.

In short, three categories of mediators are proposed:

- *communication mediators* allowing data transmission between constituent systems and the SoS. Mediators for communication purposes are: (i) pipes offering unidirectional transference; (ii) collaborators, allowing operations of request/supply between mediators, constituents and the SoS; (iii) distributors, to perform broadcasting of information; and (iv) routers, to control and coordinate data flow;

- *conversion mediators* supporting interoperability of constituent systems, and security in information exchange. Types of conversion mediators are: (i) filters, to simplify data structures or select relevant information; (ii) wrappers, to add encryption data or allow message creation; (iii) adaptors, to allow semantic interoperability through the translation or adaptation of data contents; and (iv) aggregators, to collect and merge different data, allowing the creation of SoS knowledge; and

- *control mediators*, providing mechanisms to create centralized, decentralized, and fully-decentralized control in an SoS. Control types depend of the level of managerial independence of constituent systems. This type of mediators also are proposed to allow reflection mechanisms in an SoS. Control mediators are related with MAPE operations offered by autonomic managers, hence, in this category can be found the following mediators: (i) monitors, used to collect information from constituent systems and establish their current status; (ii) analysers, to establish situations of both SoSs and their constituents based on historical knowledge on its operations, supporting the prediction of expected emergent behaviours; (iii) planners, to select reconfigurations plans to be performed, allowing thus, the dynamic reconfiguration of an SoS; and (iv) executors, to realize reconfigurations plans.

2.3. Reference Architectures for AAL and e-Health Systems

Currently, to the best of our knowledge, there is no reference architecture for HSH systems. In this section, it is provided a review about existing reference architectures in two related domains, namely, AAL and e-Health systems. Table 1 shows each reference architecture found during the review, its identification, authors, the title of the study that reported it, its name, and its purpose. Reference architectures RA1 to RA11 were previously identified at conducting a systematic literature review (SLR) on reference models and reference architectures for AAL systems (GARCÉS *et al.*, 2015; GARCÉS *et al.*, 2017b). The other reference architectures (RA12 to RA18) were obtained after updating the SLR.

Table 1 – Reference Architectures for AAL and e-Health Systems.

ID	**Author**	**Study title**	**Name**	**Purpose**
RA1	(LIU *et al.*, 2005)	Reference Architecture of Intelligent Appliances for the Elderly	SISARL	Personal care.
RA2	(BERGER; FUCHS; PIRKER, 2007)	Ambient Intelligence - From Personal Assistance to Intelligent Megacities	AmIRA	Person-centered health management, personal and home safety & security, personal care, and situation awareness.
RA3	(KURSCHL; MITSCH; SCHOENBOECK, 2008)	An engineering toolbox to build situation aware ambient assisted living systems	NR	Situation awareness
RA4	(FERNANDEZ-MONTES *et al.*, 2009)	Smart Environment Software Reference Architecture	NR	Smart environment and home automation.
RA5	(HIETALA *et al.*, 2009)	FeelGood - Ecosystem of PHR based products and services	Feelgood	Health, rehabilitation, and care.
RA6	(KAMEAS; CALEMIS, 2010)	Pervasive Systems in Health care	NR	Person-centered health management, tele-monitoring and self-management of chronic diseases, personal activity management, and personal care.

Continued on next page

Table 1 – *Continued from previous page*

ID	Author	Study title	Name	Purpose
RA7	(KEHAGIAS *et al.*, 2010)	Implementing an Open Reference Architecture Based on Web Service Mining for the Integration of Distributed Applications and Multi-Agent Systems	OASIS	Person-centered health management, tele-monitoring and self-management of chronic diseases, personal and home safety & security, personal activity management, participation in community activities, mobility, and support for working.
RA8	(WARTENA *et al.*, 2010)	Continua: The Reference Architecture of a Personal Telehealth Ecosystem	Continua	Person-centered health management, tele-monitoring and self-management of chronic diseases, personal activity management, and personal care.
RA9	(TAZARI *et al.*, 2010b)	PERSONA (PERceptive Spaces prOmoting iNdependent Aging)	PERSONA	Person-centered health management, personal and home safety & security, participation in community activities, mobility, and home automation.
RA10	(HANKE; MAYER; HOEFTBERGER, 2011)	Universal open architecture and platform for AAL	UniversAAL	Person-centered health management, personal and home safety & security, and entertainment and leisure.
RA11	(TUOMAINEN; MIKKANEN, 2011)	Reference architecture of application services for personal wellbeing information management.	Coper	Person-centered health management, and personal care.
RA12	(DENTI, 2014)	Novel pervasive scenarios for home management: the Butlers architecture	Butlers	Smart environment.
RA13	(NITZSCHE *et al.*, 2014)	Communication Architecture for AAL	AALICE	Tele-medicine, and health, rehabilitation and care.
RA14	(CAMARINHA-MATOS *et al.*, 2014)	Care services provision in ambient assisted living	AAL4ALL	AAL ecosystem.
RA15	(LOSAVIO; ORDAZ; ESTELLER, 2015)	Quality-Based Bottom-up Design of Reference Architecture applied to Healthcare Integrated Information Systems	HIS-RA	Health Information Systems.
RA16	(OPENEHR, 2015)	openEHR Architecture.	openEHR	Electronic Health Records.
RA17	(SAMARIN, 2016)	Smart Homes as Systems-of-Systems, a Reference Architecture	SHaaSoS	Home automation.
RA18	(BANDARA, 2017)	Connected Health Reference Architecture	HC-WSO2	Connected health.

As evidenced in Table 1, the first reference architectures was defined in 2005, thirteen years ago, and since then, these domains count with at least eighteen of these architectures. The time period between 2005 and 2011 has the most concentration of reference architectures, most of them supported by the European Commission framework programmes (i.e., FP6 and FP7), when the AAL concept has started its consolidation.

Reference architectures for AAL and e-Health are focused on supporting the development of the following software systems:

- **Person-centred health management:** Software systems that, through the use of mobile devices, allow the access to medical records (e.g., containing current treatment, chronic diseases, allergies, or medications). Such systems empowers the person with relevant knowledge and on-line support allowing him/her to take more responsibility for their own health. Considered in RA2, RA6, RA7, RA8, RA9, RA10, and RA11.

- **Personal and home safety & security:** Software systems oriented to recognize emergencies in the home, using sensors (e.g., positioned at electrical devices, doors, or windows, among other appliances) that are integrated into a house-control system. Considered in RA2, RA7, RA9, and RA10.
- **Personal care:** Software systems, sometimes involving assistance robots, to support people in carrying out activities of daily life such as, taking drugs, dressing and undressing, and personal hygiene. Considered in RA1, RA2, RA7, RA8, and RA11.
- **Health, rehabilitation, and care:** Software providing memory services, e.g., smart medication dispenser, tele-monitoring of rehabilitation activities, and communication with carers. Considered in RA5, RA6, and RA13.
- **Tele-monitoring and self-management of chronic diseases:** Software systems that monitor physiological parameters of people and control their health (in most cases not in an invasive manner). For this, such systems can use wearable sensors, advanced signal processing techniques and network systems. This information can be provided remotely to users, their families, and clinicians, so they are constantly aware of the health conditions of subjects. Moreover, this information support clinicians to make accurate diagnosis, identify the correct therapies, and intervene at the right time. Considered in RA6, RA7, and RA8.
- **Personal activity management:** Software systems that monitor persons activities of daily life (AoDL) aiming to provide information about their physical or mental condition. These systems can signal cognitive decline or prevent incidents. Considered in RA6, RA7, and RA8.
- **Participation in community activities:** Systems that facilitate access to news and allow active interaction of elders in a community's events and decisions. Considered in RA7 and RA9.
- **Mobility:** Systems that support the physical mobility of people (e.g., localization/positioning and navigation support systems), to assist driving (e.g., autonomous vehicles), and to improve the use of public transport (e.g., systems with travel time, or pre-trip planning functionalities). Considered in RA7 and RA9.
- **Support for working:** Systems that allow individuals with disabilities to work and elderly people to extend their working capabilities, e.g., smart computer interfaces or assistant robots. Considered in RA7.

- **Entertainment and leisure:** Systems focused on supporting brain training, physical exercising, and gaming. Considered in RA10.
- **Situation awareness:** Software that provides anticipatory assistance to humans needs based on reliable analysis of the user situation. Considered in RA2 and RA3.
- **Smart environment:** Systems composed of embedded sensor and actuator systems that are networked by wireless Internet. These systems have capabilities to sense, elaborate, and communicate in different environments, e.g., cars, factories, buildings, offices, shops, hospitals, open spaces (AALIANCE, 2010). Considered in RA4 and RA12.
- **Home automation:** Systems commonly called "domotic", composed of devices to manage and automate house furnitures (AALIANCE, 2010). Intended in RA4, RA9, and RA17.
- **Tele-medicine:** Systems that make use of audio, video, and other telecommunications and electronic information processing technologies for the transmission of health information. These systems remotely support the diagnosis and treatment of medical conditions, provide health services, or aid healthcare professionals (MAHEU; WHITTEN; ALLEN, 2001). Considered in RA13.
- **AAL ecosystem:** Systems that allow the interaction of a set of actors on top of a common technological platform for AAL to support the provision of integrated care and assistance services for senior citizens (CAMARINHA-MATOS *et al.*, 2014). Considered in RA14.
- **Health Information Systems:** Systems that collect data from the health sector and other relevant sectors, analyse the data, and ensures their overall quality, relevance and timeliness. These systems offer health information for decision-making purposes (WHO, 2008). Considered in RA15.
- **Electronic Health Records:** Systems created to accurately manage patients data and capture their health situations over time. Considered in RA16.
- **Connected Health:** Integrated systems, devices, and stakeholders to provide healthcare services in a collaborative manner (BANDARA, 2017). Considered in RA17.

Most of the reference architectures (RA1, RA2, RA6, RA7, RA8, RA9, RA10, and RA11) for AAL and e-Health systems found in literature, are oriented to the development of software systems for person-centred health management, personal and home safety & security, and personal care. These reference architectures were proposed between 2005 and 2011 (i.e., period in which the AAL concept has started its consolidation), and as evidenced, their focus are the creation of solutions to support patients in their daily activities.

An interesting observation is, that two of the most recent reference architectures (RA13 - RA18), which were proposed during the last four years, changed the focus from patient-oriented as presented by their predecessors, to offer solutions for large-scale and interconnected AAL and e-Health systems. For instance, the architecture proposed by Nitzsche et al. (2014) (RA13) is oriented to the integration of existing tele-medicine solutions with AAL systems running in different environments, to support the management of all health services in Germany. Similarly, Camarinha-Matos et al. (2014), proposed AAL4ALL (RA14), a reference architecture focused on the standardization of AAL ecosystems, involving heterogeneous solutions of AAL and e-Health, technological infrastructure, health organizations, and different types of stakeholders interests. Losavio et al. (2015), with HIS-RA (RA15), established guidelines to integrate heterogeneous, independent, and distributed Healthcare Information Systems (HIS), allowing the creation of software products for those systems. A particular type of HIS is the Electronic Health Record (EHR) system, that is the focus of the organization *openEHR*, that intends to standardize the development of EHR (and Personal Health Records - PHR) in all over the world with its open architecture (RA16). Samarin (2016) proposed the first reference architecture (RA17) that considers smart homes as SoS, i.e., homes whose constituents (e.g., systems for activity monitoring, situation awareness, home security, energy consumption management), are pervasive, and operational and managerial independent. Finally, Bandara (2017) established a reference architecture from a perspective of connected health to support the offering of efficient and reliable health services through the connection of all e-Health systems (e.g., systems to support financial management, resources optimization, epidemic management, or emergency management).

Despite the great variety of existing reference architectures for AAL and e-Heatlh, few of them have offered solutions for supporting patients in the self-management and continuous monitoring of chronic diseases, such as, diabetes mellitus, dementia, heart failure, cancer, among other important disease, that are currently affecting global public health systems. As evidenced in Table 2, just three reference architectures (RA6, RA7, and RA8) provide reusable knowledge to design systems for tele-monitoring and self-management of chronic diseases. Considering that HSH systems aim to offer the required assistance to chronic patients at home, and that, there is no approach to support the quality development of these systems, RA6, RA7, and RA8 were analysed regarding their viability for guiding the construction of HSH systems. For this purpose, available documentation of the three reference architectures was reviewed. Specifically, it was investigated whether using these architectures, the resulting HSH systems can present the characteristics detailed in Table 2, which are required to solve the challenges presented in those systems.

Table 2 – Viability of reference architectures to develop HSH systems.

Characteristic	RA6	RA7	RA8
Self-management of chronic conditions at home			
Continuous monitoring of patients health		✓	
Provision of complete profiles of patients health situation			
Prevention, identification, and recovery of emergency situations at home			
Requirements specification of quality attributes		✓	✓

Continued on next page

Table 2 – *Continued from previous page*

Characteristic	**RA6**	**RA7**	**RA8**
Adequate domain knowledge	✓		✓
Instantiation guidelines		✓	✓
Architectural solutions description (e.g., architectural patterns or tactics)	✓	✓	
Interoperability with systems outside home			✓
Interoperability with systems inside home			
Adequate abstraction level			
The construction of software easy to maintain and evolve		✓	

Brief discussions about this viability investigation are given as follows.

- Despite the three reference architectures claimed to support patients in the self-management of their conditions, none of them gives orientations to define how this support can be provided at home. Moreover, it is not possible neither to establish a complete knowledge about patients health situations nor to detect emergency occurrences in patient's health and environment. Additionally, all three architectures present high abstractions levels, requiring further steps for their use in the development of more concrete HSH systems.

- Continuous monitoring of patients health is justly addressed by RA7, that has defined an infrastructure to connect health monitoring services running at patients home with services for the same purpose running in health providers installations. However, RA7 does not specify neither the type of monitoring services required to attend patients conditions (i.e., monitoring services must be setted for each patient profile, since her/his health status is unique), nor the nature of services offered for the care team (i.e., depending of patient conditions, different health specialists are involved in the monitoring of her/his status).

- Two reference architectures (RA7 and RA8) specified requirements of quality attributes related to security, reliability, interoperability, maintainability, and adaptation. However, just RA7 offered architectural solutions to address these requirements. This reference architecture is based on SOA, software agents, and ontologies, structured in seven tiers. Despite RA7 and RA8 reported the importance of addressing interoperability, they do not specify in detail how this attribute is addressed by systems running inside and outside home. For instance, RA7 claims to address interoperability by using web services, and RA8 through by using some kind of network interfaces. However, no approach is suggested to achieve semantic, syntactic, and process interoperability. Regarding maintainability issues, only RA7 declares its achievement through the use of services, and the modularity of functionalities in tiers.

- An adequate description of domain knowledge is offered by RA6 and RA8. RAModel was used to specify the adequacy of domain descriptions offered by reference architectures (GARCÉS *et al.*, 2017b). RA7 and RA8 provided guidelines for their instantiation in concrete solutions, which due their high abstraction level, can lead to possible misunderstandings.

In this perspective, none of the existing reference architectures for AAL and e-Health is suitable to architect HSH systems, and hence, to offer solutions for supporting chronic patients, at home, in the self-management and continuous monitoring of their conditions. Moreover, existing reference architectures do not consider the provision of a complete patients health situation to physicians and other members of the healthcare team, limiting the communication of patients information with other e-Health systems (e.g., epidemiologic surveillance systems). Additionally, these architectures do not define mechanisms to prevent, identify, and recover from emergency situations (related with patient's environment or health) at home. Finally, interoperability with internal and external systems is limited at using these reference architectures, hindering the integration of HSH systems with the entire e-Health ecosystem.

Therefore, the reference architecture proposed in this thesis advances the state of the art, since it intends to address the characteristics proposed in Table 2, to overcome challenges related with HSH systems, and contribute to the areas of AAL, e-Health, and reference architectures for SoS.

2.4. Final Considerations

To the best of our knowledge, there is a lack of reference architectures offering guidelines to orient and standardize the development of quality-based HSH systems. In this thesis, is proposed *HomecARe*, a reference architecture for HSH systems that assist chronic patients in the self-management and continuous monitoring of their conditions, at home.

To create *HomecARe*, the four steps proposed by ProSA-RA, namely, domain analysis, architectural analysis, architectural synthesis, and architectural evaluation, were conducted. ProSA-RA was detailed in Section 2.1. Results of performing each step are presented in Chapters 3 (domain analysis), 4 (architectural analysis), 5 (architectural synthesis), and 6 (architectural evaluation), as depicted in Figure 4.

In this chapter, the theoretical background considered during the conduction of this thesis was given. In particular, in Chapter 5, some of the architectural patterns for SoS detailed in Section 2.2 were used as start points to design *HomecARe*. Moreover, *HomecARe* considers several mediators described in the taxonomy proposed in Section 2.2.

CHAPTER

3

HEALTHCARE SUPPORTIVE HOMES DOMAIN ANALYSIS

This chapter presents the analysis of the HSH domain, as stated in the first step of ProSA-RA described in Section 2.1. Section 3.1 provides an overview of the concept of home health care, detailing its importance to improve patients quality of life from their residences. Additionally, four categories of stakeholders are presented, as well as their involvement in the provision of healthcare services to patients at home. Section 3.2 gives the state of the art on HSH systems, detailing some existing systems in this domain and their purpose, as well as other studies investigating these systems. Considering the current interoperability challenge that HSH systems must face, Section 3.3 describes how interoperability has been treated in other e-Health systems, such as HIS, highlighting the importance of addressing technical, semantic, syntactic, and process interoperability within the domain. Moreover, aiming the investigation of other quality attributes requirements in the domain, Section 3.4 presents the QM4AAL (Quality Model for AAL systems), that is further used to identify which qualities HSH systems must be focused on. This chapter ends with some final considerations in Section 3.5.

3.1. Healthcare Supportive Home

Home health care is a range of supportive care services that can be provided to patients at their home (MEDICARE, 2017). Home care allows patients to remain in the comfort of their home, while they are receiving services to recover from illness, injury, or disability, or to support patients with the management of their chronic conditions, e.g., diabetes, hypertension, heart failure, and cancer (MCLAIN; O'HARA-LESLIE; WADE, 2016). Home health care may be provided for people who have cognitive or physical disabilities to help them complete activities of daily living (AoDLs). Home health care may also be provided for patients who are on hospice. Hospice home health care is for patients who have been diagnosed with a terminal illness (an illness that cannot be cured) and have a prognosis of six months or less. Hospice home health care allows people to remain in the comfort of their homes, surrounded by familiar people and things (MCLAIN; O'HARA-LESLIE; WADE, 2016).

People who receive home care have a variety of needs depending on their physical condition and specific disease or injury. Patients may need reminders or help to take their medications, and often need assistance with ambulation (walking) and transferring (moving) from a bed to a chair or wheelchair, or getting in and out of the shower. Many patients have adaptive equipment, such as walkers, wheelchairs, canes, and prosthetic devices, which assist them in moving around their home. Often, patients require help with AoDLs, such as toileting, bathing, dressing, and eating, and some of them need help to manage their budget, and purchase and cook food. Patients may need assistance with changing simple dressings on wounds, making and changing their bed linens, doing laundry, and maintaining a safe and clean home (MCLAIN; O'HARA-LESLIE; WADE, 2016).

Stakeholders of HSH Systems

AAL systems, including HSH systems, involve multiple stakeholders, i.e., people, organisations, and system (or parts of it), which are affected by the system functionality, and have a direct or indirect influence on the system requirements. In the context of AAL systems, Huch (2010) established four categories of stakeholders, namely primary, secondary, tertiary, and quaternary stakeholders, which are detailed as follows:

A. Primary and Secondary Stakeholders

Home healthcare services are offered by the *home healthcare team*, which is responsible for supporting patients to improve their health conditions. Roles of the home healthcare team members can change depending on the country legislations, but in a general way, basic roles are the following (MCLAIN; O'HARA-LESLIE; WADE, 2016; TAO; MCROY, 2015):

a. Professional Caregivers: Professional caregivers deliver services at the elder patient's home to help them to develop independence with daily activities in a convenient and comfortable setting, allowing relatives to be closely involved in the recovering/rehabilitation process (TAO; MCROY, 2015). Professional caregivers conforming the healthcare team are the following:

- **Physician (MD or DO)** oversees patient care, diagnoses and monitors conditions, and prescribes medications and treatments. Many doctors also have speciality certifications, such as endocrinologist, pathologist, etc (MCLAIN; O'HARA-LESLIE; WADE, 2016).
- **Registered Nurse (RN)** coordinates and manages the patient's care. A RN performs assessments, monitors test and laboratory results, administers treatments and medications, monitors the patient's condition, and provides education to the patient and family. RNs

supervise Licensed Practical Nurses (LPNs), Home Health Aides (HHA), and Personal Care Aides (PCA) (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Licensed Practical Nurse (LPN)** may administer medications, check vital signs, provide wound care, collect samples for testing, such as urine and blood, and assist with patient self-care activities (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Physical Therapist (PT)** helps patients to strengthen and restore their ability to be mobile and prevent further injury. PTs may teach patients to use special equipment, such as walkers and canes, assist patients with specific exercises to help regain mobility and strength, and administer treatments, for instance, heat, or cold to help improve patient circulation, reduce pain, prevent disability, and improve muscle and joint function (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Occupational Therapist (OT)** work with patients to help them to learn how to live with a disability so they may function as independently as possible. OTs help patients in the performing of their activities of daily living such as dressing, eating, and bathing. An OT teaches patients how to use assistive and adaptive devices such as special forks, plates, long-handled shoe horns and sponges, and raised toilet seats (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Speech-Language Pathologist (SLP)** or speech therapist works with patients who have communication or swallowing disorders, have experienced strokes or accidents, or have a neurological health problem. SLPs teach patients exercises to improve speech, to effectively communicate, and to safely swallow. They may suggest special diets to aid in swallowing, such as mechanical diets and thickened liquids (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Registered Dietician (RD)** evaluates a patient's nutritional intake and orders special diets for patients. They provide education to patients and relatives about special diets to manage their illness and improve their nutrition (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Medical Social Worker (MSW)** works with the patient and family to help them get support services, such as counselling, financial assistance, and community services. An MSW provides emotional support to the family and works as an advocate to meet the patient's needs (MCLAIN; O'HARA-LESLIE; WADE, 2016).

b. Non-professional Caregivers: Include unpaid family caregivers (e.g., sometimes a family member) and paid informal caregivers, namely, homecare aides, either managed by the family or a commercial or government agency in the community (TAO; MCROY, 2015). Non-professional caregivers that participate in the home healthcare team are detailed as follows:

- **Home Health Aide (HHA)**, under the supervision of a RN, provides supportive care to patients within their homes. A HHA works to increase or maintain independence, health, and well-being of patients. HHAs provide or assist with self-care activities, such as bathing, dressing, grooming, toileting, feeding, skin care, use of medical supplies and equipment (e.g., walkers and wheelchairs), and assisting with light housework, laundry, and home safety (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Personal Care Aide (PCA)**, under the supervision of a RN, provides self-care and companionship to patients. PCAs assist with self-care activities such as bathing, dressing, grooming, toileting, feeding, skin care, and use of assistive devices, e.g., walkers and wheelchairs. They also assist with housekeeping tasks, namely laundry, changing bed linens, washing dishes, and preparing meals. PCAs may not perform any type of medical service or task, as a HHA may. A PCA may not take vital signs or glucose meter readings (MCLAIN; O'HARA-LESLIE; WADE, 2016).

- **Patient/Family** are the most important parts of the healthcare team. Patients have the right to make decisions about their health care. They have a right to be informed about treatments and the care they receive. They have a right to refuse treatments, medications, and services. All patients and their relatives are unique and have various needs, desires, cultures, and traditions. It is important that the healthcare team respects these individual differences and work to meet each patient's needs. Without the patient, the healthcare team is not necessary (MCLAIN; O'HARA-LESLIE; WADE, 2016).

B. Tertiary and Quaternary Stakeholders

Tertiary stakeholders are suppliers of AAL systems, e.g., research organisations, health organizations, enterprises with a business in tele-medicine or tele-care or providers of the IT infrastructure. Quaternary stakeholders are supporters of AAL systems, e.g., policy-makers or social (and private) insurance companies (HUCH, 2010).

3.2. Healthcare Supportive Home Systems

HSH[1] systems are considered as a special type of AAL system, and were initially defined in 2002 by Rialle et al.(2002) as "Smart Homes that aim to monitor and improve health related parameters". Similarly, Noury et. al(2003) established the HSH as "a specialization of the Smart Home that aims at giving an autonomous life, in their residence, to people suffering from various pathologies and handicaps that should normally force them into a hospitalization or placement in specialized structures (NOURY *et al.*, 2003)". A recent definition is given by Bennett et al.(2017), who establish that a HSH is "a home or dwelling with a set of networked sensors and

[1] Also found in literature as Health Smart Homes (HSH), Smart Home Healthcare (SHH) or Home-based Consumer Health (HCH).

devices that extend the functionality of the home by adding intelligence, automation, control, contextual awareness, adaptability, and functionality both remotely and locally, in the pursuit of improving the health and well-being of its occupants and assisting in the delivery of healthcare services".

In the context of this thesis, the adopted definition of HSH is[2]:

> A HSH system is the combination of prevention tools and services that address not only response to alarm situations but also trigger preventive actions based on perceived trends and behavioural patterns. Such services are based on intelligent situation analysis, help in the assessment of patient health conditions, and assist on fitted interventions. Moreover, HSH could consider local interventions (e.g., resorting to robotics), tele-presence (in connection with carers), or interventions provided by active support teams from the care network.

HSH Systems - State of the Art

The development of HSH systems has significantly increased in the last years, principally because patients prefer to treat their conditions at home for comfort reasons. In (GAIKWAD; WARREN, 2009), it was evidenced that treating patients suffering of chronic diseases, with the support of health-based ICT, improves functional and cognitive patient outcomes and reduces healthcare spending. Besides HSH systems offer means to patients for their self-management of chronic diseases, such systems are a common strategy to reduce health services costs and, at the same time, they improve quality of life, maximize independence, minimize disabilities of older adults, and reduce the change of social-isolation (WARTENA *et al.*, 2010; PARÉ *et al.*, 2010; REEDER *et al.*, 2013; BENNETT; ROKAS; CHEN, 2017). In this perspective, HSH systems may be considered part of public health strategy in next years (REEDER *et al.*, 2013).

Moreover, the positive impact of such technologies is evidenced by the amount of projects related to HSH systems that have been funded by European Commission (EC) [3], the AAL Joint Programme (AAL JP)[4], and the Horizon 2020 (H2020) programme. By 2017, 22 projects were funded and it is expected new projects are supported by the H2020 programme[5]. The 22 funded projects are listed in Table 3, providing details about the founding programme call, project names, links to respective websites, and the chronic condition for which each project was conceived.

2 This definition was adapted from the "supportive environment" description given by (CAMARINHA-MATOS *et al.*, 2011).
3 <http://ec.europa.eu/research/health/index.cfm?pg=projects>
4 <http://www.aal-europe.eu/>
5 Through the call: "Health, demographic change and well-being"

Table 3 – HSH funded projects.

Programme	Project Name	Url	Focus
EC-FP7	Chronious	<http://www.cordis.europa.eu/project/rcn/85452_en.html>	COPD[6] and renal insufficiency.
EC-FP7	HomeCare	<http://wedo.tttp.eu/>	Stroke, heart failure, COPD.
EC-FP7	Giraff+	<http://www.giraffplus.eu/>	Chronic diseases.
EC-FP7	Hearthcycle	<http://www.heartcycle.eu/>	Heart failure.
AAL JP	Agnes	<http://www.aal-europe.eu/projects/agnes/>	Mental decline diseases.
AAL JP	Alladin	<http://www.aal-europe.eu/projects/alladin/>	Dementia.
AAL JP	Amica	<http://www.aal-europe.eu/projects/amica/>	COPD.
AAL JP	Bedmond	<http://www.aal-europe.eu/projects/304/>	Neurodegenerative disease.
AAL JP	CCE	<http://www.cceproject.eu/>	Dementia.
AAL JP	ecAALIX	<http://www.aal-europe.eu/projects/ecaalyx/>	Chronic conditions.
AAL JP	H@H	<http://www.aal-europe.eu/projects/healthhome/>	Chronic hearth failure.
AAL JP	HELP	<http://www.aal-europe.eu/projects/help/>	Parkinson's disease.
AAL JP	HERA	<http://www.aal-europe.eu/projects/hera/>	Mild cognitive impairment, Alzheimer's disease.
AAL JP	IS-ACTIVE	<http://www.aal-europe.eu/projects/is-active/>	COPD.
AAL JP	REMOTE	<http://www.aal-europe.eu/projects/remote/>	Chronic conditions.
AAL JP	Alfa	<http://www.aal-alfa.eu/>	Alzheimer's disease
AAL JP	ECH	<http://www.aal-europe.eu/projects/ecarehome/>	Mental disorders.
AAL JP	Dem@care	<http://www.demcare.eu/>	Dementia.
H2020	SMART4MD	<http://www.smart4md.eu/>	Mild dementia.
H2020	SMART2D	<http://ki.se/en/phs/smart2d>	Diabetes.
H2020	CONNECARE	<http://www.connecare.eu/>	Chronic care management.
H2020	HELMO	<http://cordis.europa.eu/result/rcn/185297_en.html>	Cardiovascular diseases diagnosis.

Continued on next page

Moreover, several systematic literature reviews evidence the advances in researches on ICT[7]-based treatment of chronic diseases (and related co-morbidities) at home. These reviews investigate the impact (from economic and stakeholders point of view) on using HSH systems for self-management of patients conditions are presented in (GAIKWAD; WARREN, 2009; PARÉ

6 Chronic Obstructive Pulmonary Disease
7 Information and Communication Technologies

et al., 2010; KITSIOU; PARÉ; JAANA, 2013; REEDER *et al.*, 2013). From a technological perspective, Toumpaniaris and Iliopoulou(2014) present an overview of different sensors and systems that can be used in HSH systems. Specifically, such study classifies sensors to support monitoring at home of heart failure condition, pulmonary diseases, diabetes, hypertension, Alzheimer's disease, and Parkinson's disease. Similarly, Amiribesheli and Benmansour(2015) reviewed sensors, communication technologies, data processing techniques, and user interfaces considerations to develop Smart Home systems, with a focus on health care. An important contribution of Amiribesheli and Benmansour's work is the evidence offered about several techniques used in Smart Homes in the context of data processing and knowledge engineering, such as decision trees, fuzzy logic, artificial neural networks, support vector machines, naive Bayes classifier, Hidden Markov model and its variants, conditional random field, emerging patterns, ontological modelling, and context-aware reasoning.

In a similar effort, Reeder and Meyer(2013) conducted a systematic literature review to identify HSH projects and assess them regarding their maturity and application with real patients. This study aims to obtain evidences to propose HSH systems as an approach to support public health interventions for achieving independent ageing. In such study, 31 HSH projects were analysed. Capabilities offered by such HSH systems were classified as: (i) physiological monitoring, as the function for collecting and analysing physiological measurement data such as vital signs;(ii) functional monitoring/emergency detection and response, as the capacity of collecting and analysing functional measurement data such as general activity level or falls; (iii) safety monitoring and assistance, as a function of collecting and analysing environmental hazard data, such as flooding and notification of floods; (iv) security monitoring and assistance, as detection of intruders and notification of identified threats; (v) social interaction monitoring and assistance, as the collection and analysis of data pertaining to social interactions and technologies that facilitate social interactions; and (vi) cognitive and sensory assistance, offering support to compensate for sensory deficits, giving reminders or task instructions.

In the same perspective, Avila et al.(2017) conducted a literature review aiming to obtain evidence on the use of SOA for developing HSH systems. Specifically, this study classified 29 HSH projects regarding managed physiological and environmental data. Physiological data measured by HSH systems are related to blood pressure, heart rate, body temperature, SpO2[8], body composition (e.g, weight, body mass index), electrical activity of heart, blood flow to skin, breathing rate, and body movements. Environmental data measured by HSH systems are related to environment temperature, humidity, dust concentration, and pollution. Moreover, the study of Avila et al. analysed approaches to overcome interoperability in HSH projects. For instance, intermediate agents, mid-tier agents, ontologies mappers, and mediator layer were some strategies used in HSH systems for interoperability purposes. Regarding overcome security issues, the work of Avila et al. found that data encryption, authentication mechanisms, DPWS

[8] Oxygen saturation in the blood.

(Device Profile for Web Services), and HTTPS (Hyper Text Transfer Protocol Secure) protocols are some approaches used by HSH systems.

Potential areas of future research in health care in Smart Home are given by Bennet and Rokas(2017). This study offers a broader view on past, present, and new challenges for developing HSH systems. Such study details important characteristics of HSH systems found after analysing 10 of those systems, such as: (i) Environmental sensors located within doors, chairs, walls, and floor to measure light, temperature, air quality, weather conditions, and pressure; (ii) Personal sensors, such as smart glasses, belts, caps, armbands, pulse trap, smart bra, and other wearable sensors, to track steps, floors climbed, sleep, heart rate, and patient location; (iii) Controllers and gateways to aggregate and distribute data gathered from all environmental and personal sensors to external services; (iv) actuators and smart appliances to control heating, lighting, turn on/off appliances; and (v) software implementing home automation and integrating healthcare services. Moreover, Bennet and Rokas' study offers a reclassification of common Activities of Daily Living (AoDL)[9] in six categories: (i) Cognitive orthotics and mental augmentation, such as reminders, navigation assistance, reasoning assistance, planning tools, multimedia coaching, or information retrieval; (ii) Continuous vital sign monitoring, such as, blood pressure monitoring, heart rate monitoring, sleep monitoring, AoDL monitoring, habit monitoring, or movement detection; (iii) Social assistance, such as telephone calls, human interaction or on-line interaction; (iv) Medical intervention, such as tele-support, remote health assistance, or on-line medical tools; (v) Medical emergency detection, such as fall detection, abnormal behaviour detection, or hazard detection; and (vi) Physical assistance, as home automation, assistive tools, or predictive software.

3.3. Standards for Health Information Systems Interoperability

Interoperability is defined as the ability of two or more systems or components to exchange information and to correctly use such information (IEEE, 1990). In the healthcare domain, interoperability is a critical requirement for effective communication between Healthcare Information Systems (HIS). Some of benefits obtained at using interoperable HIS, is to ensure reliable access to patient information, the possibility of creating and accessing to medical knowledge, and to improve patients care (DUCROU, 2009). Interoperability in software systems can be seen as a set of layers built upon each other (KUBICEK; CIMANDER; SCHOLL, 2011). Such set is formed at least by four layers, namely, technical, syntactic, semantic, and organizational interoperability (KUBICEK; CIMANDER; SCHOLL, 2011). In the context of healthcare informatics, interoperability is also considered as a four-layer configuration constituted by technical, semantic, process, and clinical interoperability layers (BENSON; GRIEVE, 2016).

9 The first classification of AoDL was made in the AALIANCE project <http://www.aaliance2.eu/> .

In this section, definitions about each layer are given, detailing standards in the healthcare domain to realize each interoperability type.

Technical Interoperability: It is related to establish links and transmit data between components or systems independently of their distance. This type of interoperability is domain independent, and it does not know about the meaning of data exchanged (BENSON; GRIEVE, 2016). Hence, technical interoperability only guarantees the correct transmission of bits, but it does not tell anything about the meaning of these bits and what they represent (KUBICEK; CIMANDER; SCHOLL, 2011). This type of interoperability is often focused on communication protocols and infrastructure needed for those protocols to operate (VEER; WILES, 2006). In health informatics, messaging is the approach used to achieve technical interoperability between HIS (BENSON; GRIEVE, 2016).

Syntactic Interoperability: Syntax is related to the rules of what kind of data are exchanged, how to put together and in which order. In this context, syntactic interoperability is focused on provide data formats, well-defined syntax and encoding (e.g., message content structure, size of headers, size of message body, fields contained into a message). In healthcare domain, several standards, referred as clinical information models, were conceived to support syntactic interoperability through the definition of message syntaxes. Examples of standards are: Health Level 7 (HL7 in its versions 2 and 3)[10] and GALEN[11].

Semantic Interoperability: Semantics is related to the exact meaning of the exchanged data (e.g., clear definitions of clinical terminology or vocabulary). In this context, in semantic layer, data are conceived as information to be shared, processed, and well-understood (without ambiguity) by systems (IDABC, 2004). Semantic interoperability is specific to domain and context and requires the use of unambiguous codes and identifiers (BENSON; GRIEVE, 2016). In health informatics, there are more than one hundred classification of health data (SIMONET *et al.*, 2009), represented as clinical terminologies, taxonomies, and ontologies, which have been proposed to achieve semantic interoperability between HIS. The most representative clinical terminologies are SNOMED (Systematised Nomenclature of Medicine), ICD-10 (International Classification of Diseases) (WHO, 2016), CTv3 (Clinical Terms version 3), SNOMED CT (a combination of SNOMED and CTv3)[12], LOINC (Logical Observation Identifier Names and Codes)[13] and MeSH (Medical Subject Headings)[14]. Moreover, several ontologies have been proposed to represent and exploit health knowledge (SIMONET *et al.*, 2009), for instance, the ontology proposed in the context of heart failure (ECCHER *et al.*, 2006), the ontology established for interventions of patients with dementia (NAVARRO; RODRÍGUEZ; FAVELA, 2012), the specification of Parkinson's disease (GARCÍA-MAGARIÑO; GÓMEZ-SANZ, 2013), or the

[10] <http://www.hl7.org/implement/standards/>
[11] <http://www.opengalen.org/index.html>
[12] <http://www.snomed.org/>
[13] <https://loinc.org/>
[14] <https://meshb.nlm.nih.gov/>

K4Care Ontology for representing homecare services (CAMPANA *et al.*, 2008).

Business Processes Interoperability: Business processes can be seen as the interaction of multiple organizations to obtain their respective business goals (CHOPRA, 2008). Interoperability of business processes is focused on coordination of distributed work flows and activities that are well understood by systems, organizations or people interacting in such processes. Organizations participating on a business process must commit to perform several activities, and such commitments are specified in contracts agreed with other organizations that are taking part of the business process. In this perspective, a prominent approach to achieve business processes interoperability is the use of SOA, which allows a common description of inter-organizational processes (KUBICEK; CIMANDER; SCHOLL, 2011).

Regarding health informatics, important business processes are related to referral and consultation communication between clinicians, as such communications and consultations (e.g., between primary care physicians and specialists) are often inadequate, with negative consequences for patients (O'MALLEY; RESCHOVSKY, 2011). Hence, clinical interoperability must be considered to address full interoperability in HIS. In short, clinical interoperability is the capability of different physicians (e.g., with different knowledge of patient's health status) to refer patients, and provide them seamless care (BENSON; GRIEVE, 2016).

Clinical guidelines to treat patients can support the establishment of business processes, that in turns, can be realized using BPM (Business Process Management/Modelling) techniques to overcome clinical interoperability issues. Some examples of clinical guidelines are those proposed by Brazilian (MINSAUDE, 2013), French (HAS, 2014), Colombian (MINSALUD, 2009), Canadian (CANADA, 2017) and Australian (QUEENSLAND-GOVERMENT, 2015) governments to prevent and manage chronic conditions.

3.4. Quality of HSH Systems

HSH systems must address all layers of interoperability detailed in Section 3.3, to allow a better prevention and management of patient's diseases. However, overcoming interoperability issues in software systems in the healthcare domain is not enough to ensure a final solution with high quality. Hence, other quality requirements must be addressed to offer a quality-based systems to healthcare ecosystems.

Considering the critical nature of software systems in healthcare domain such as AAL and, in particular, HSH systems, it is important to understand how quality assurance is performed in such systems. In this perspective, we conducted a systematic mapping to provide a broad and detailed panorama on quality models (QMs) and quality attributes (QAs) that must be considered during all phases of AAL systems engineering, namely, requirement analysis, architecture design, development, testing, deployment, and evolution (GARCÉS *et al.*, 2017a). Since HSH systems are a sub-domain of AAL, as presented in Section 3.2, results obtained from this systematic

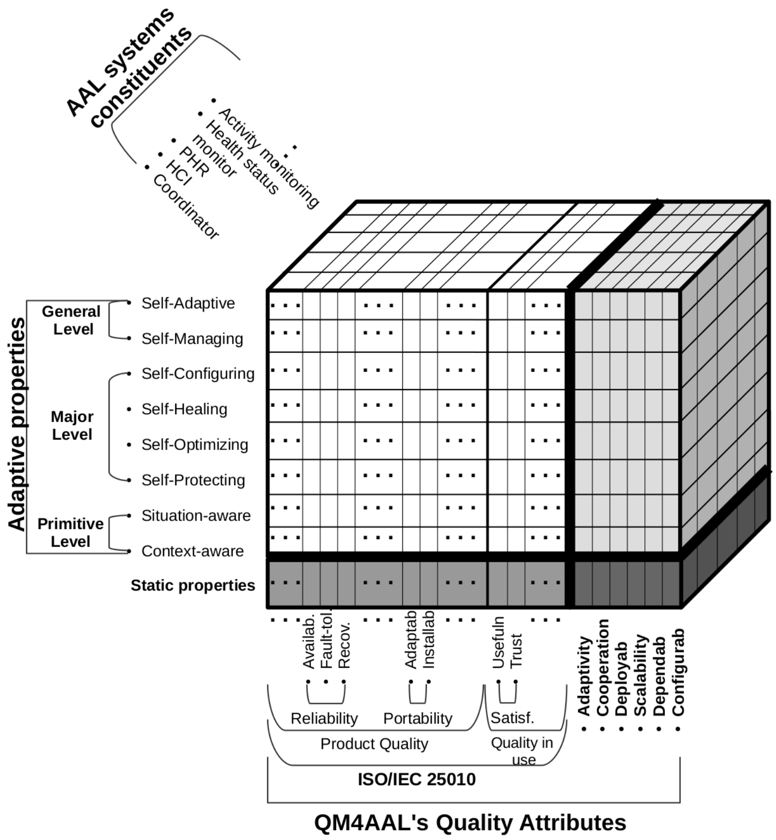


Figure 6 – The three dimensions of the QM4AAL. Source: (GARCÉS; OQUENDO; NAKAGAWA, 2016)

mapping could be applied to HSH systems engineering as well. In this mapping, 27 AAL systems were analysed to identify: (i) QMs and QAs requirements for such systems; (ii) the AAL sub-domains where quality has been considered; (iii) approaches to identify or establish QMs and QAs for AAL systems; and (iv) evaluation techniques used to asses quality in those systems. Ninety seven QAs were identified based on evidence provided by the investigated AAL systems. QAs were mapped into the standard ISO/IEC 25010 (ISO/IEC, 2011) to provide a common terminology of quality characteristics for AAL systems. Moreover, taxonomies of QAs for AAL sub-domain were defined. Furthermore, results of this mapping evidenced that it does not exist a QM to orient AAL systems engineering. In this perspective, the QM4AAL (Quality Model for AAL systems) (GARCÉS; OQUENDO; NAKAGAWA, 2016) was established and evaluated based on evidence gathered from the AAL systems found through the conduction of the mentioned mapping study. QM4AAL is composed of three dimensions, namely, AAL systems constituents, adaptive and static properties, and quality attributes. Dimensions are illustrated in Figure 6 and detailed as follows:

AAL system constituents dimension: It represents all possible constituent systems that can compose the AAL system and that are in some way involved in the QAs requirement specification.

Adaptive and static properties dimension: It covers behavioural properties of AAL systems and their constituents, and is classified in: (i) **adaptive properties**, that must be achieved at runtime and imply dynamic modifications during the system's life cycle. Most of adaptive properties can not be established at system's design time, but at executing time. Adaptive properties in QM4AAL were defined based on the classification of self-adaptive software proposed by Salehie and Tahvildari(2009) that defines three properties levels, namely, primitive level (considering context and situation aware), major level (including self-protecting, self-optimizing, self-healing and self-configuring), and general level (contemplating self-managing and self-adaptive); and (ii) **static properties**, that are predictable and can be addressed at design time without requiring modifications after the system's deployment.

Quality attributes dimension: It describes quality attributes requirements associated to achieve each system behaviour (adaptive or static). The QAs were structured according to the taxonomy showed in Figure 7, which is an extension of the standard ISO/IEC-25010 (ISO/IEC, 2011). QM4AAL contains 52 QAs requirements specifications for all levels of adaptive properties, and 123 for static properties.

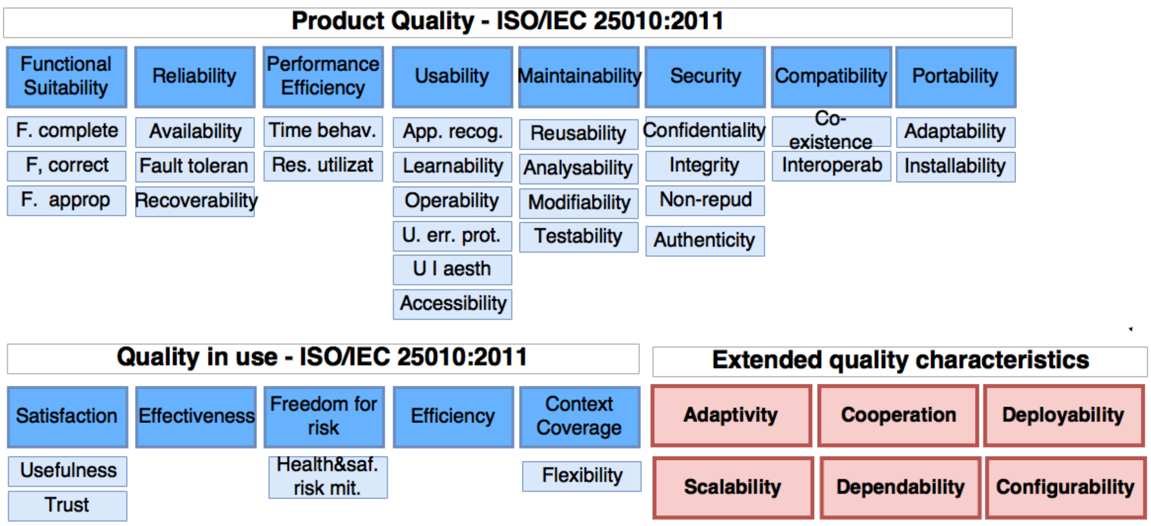


Figure 7 – Taxonomy of quality attributes of QM4AAL as an extension of the standard ISO/IEC 25010. Legend: Strong blue and light blue colours represent, respectively, characteristics and sub-characteristics of the standard, whilst red colour highlights extended quality characteristics for AAL systems.

Table 4 shows two examples of specifications of QAs requirements in QM4AAL. For each of the 175 QA requirements (i.e., 52 for adaptive properties and 123 for static properties) it is detailed: (i) the requirement ID; (ii) the adaptive property to which the requirement is described. If not presented, it is considered a requirement of static property; (iii) the name(s) of involved constituent(s) system(s), or, the AAL system when the requirement is oriented to the system as a whole; (iv) the principal QA related with the requirement; (v) the stakeholders engaged with the requirement; (vi) other QAs associated to the requirement; and (v) the requirement description.

In this perspective, QM4AAL allows the description of QAs requirements for the whole HSH system and for each of its constituent systems, providing structured specifications that can be further used in dependency analysis between constituents and the HSH system. Complete details about specifications for all QAs requirements of QM4AAL are given in (GARCÉS; OQUENDO; NAKAGAWA, 2017).

Table 4 – Examples of QAs requirements in QM4AAL

Requirement ID	QAR001
Adaptive property	Self-Managing
Constituent	AAL system
Quality Attribute	Configurability
Stakeholders	AAL system's constituents
Related QAs	-
Description	The system must provide update and delete function for applications/system components/constituents/services at runtime.
Requirement ID	QAR002
Adaptive property	-
Constituent	AAL system constituents
Quality Attribute	Confidentiality
Stakeholders	AAL system
Related QAs	Security
Description	Each part of the system (constituents) must be aware of maintaining the confidentiality of sensitive data, including controls on storage, handling, and sharing of data.

3.5. Final Considerations

In this chapter, results of conducting the first step of ProSA-RA, namely, domain analysis, were presented. As evidenced, HSH systems are large and complex systems involving a great variety of stakeholders classified as the home healthcare team, and different health services providers. To enable a continuous monitoring of patients at home, and improve their quality of life, HSH systems must consider the four interoperability layers (i.e., technical, semantic, syntactic, and process layers) presented in Section 3.3; therefore, those systems must be developed following health interoperation standards. However, interoperability is not the only type of requirement to ensure HSH systems quality. Other important QAs requirements that HSH systems could consider were detailed in Section 3.4, which are based on specifications given in QM4AAL. Finally, information presented in this chapter was used as basis to perform the architectural analysis (see Chapter 4), where domain models are presented, as well as, the architecturally significant requirements of *HomecARe*.

CHAPTER 4

ARCHITECTURAL ANALYSIS OF *HOMECARE*

This chapter presents results of performing the architectural analysis of *HomecARe*, as proposed in ProSA-RA. This analysis was based on the domain knowledge presented in Chapter 3. Section 4.1 details the following domain models of *HomecARe*: (i) missions models, represented as a tree of objectives (functional and non-functional) that HSH systems should be oriented to achieve; (ii) models describing types of constituents and their responsibilities to collaborate in the pursuit of HSH systems missions; (iii) models representing, for each constituent, all expected operational capabilities to be part of HSH systems; (iv) model of entities (data and events) required from constituents and managed by HSH systems; (v) models of required communications between constituent systems and HSH systems to achieve their missions; and (vi) models of those behaviours that can emerge because constituents and HSH systems interactions. Furthermore, based on domain models, the architecturally significant requirements of *HomecARe* were defined and are detailed in Section 4.2. This chapter ends with some final considerations in Section 4.3.

4.1. Domain Models for HSH systems

Domain models were obtained through the conduction of the process showed in Figure 8. Information and models flows are represented as dashed arrows, while the process flow is symbolized as bold arrows. This process explains how to identify missions, designate responsibilities, allocate operational and communicational capabilities to constituent systems, identify exchanged data entities, and determine emergent behaviours in reference architectures of SoS (GARCÉS; NAKAGAWA, 2017). Because architectural elements in a reference architecture must be more abstract than in concrete systems to allow their instantiation, in *HomecARe* , similar constituent systems (i.e., offering equivalent capabilities, and hence, contributing in an equal way to HSH

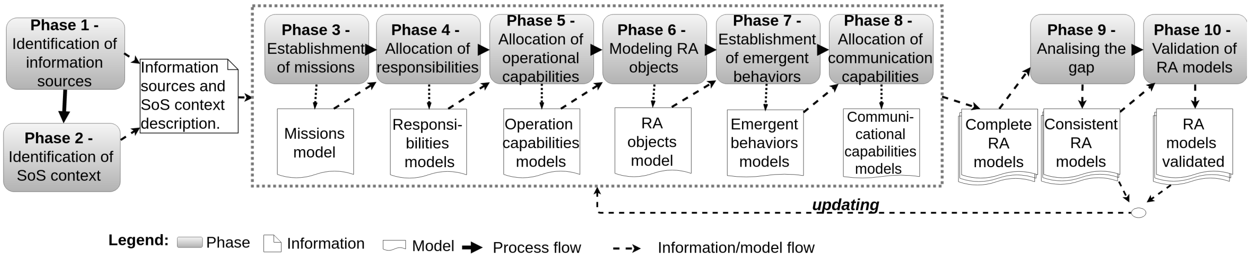


Figure 8 – Process to establish missions for reference architectures of SoS. Source: (GARCÉS; NAKAGAWA, 2017).

systems missions) were generalized as a class of constituent system in this reference architecture.

As an initial phase (Phase 1), domain information sources were identified and studied to obtain a well-consolidated knowledge on HSH systems, as presented in Chapter 3. The second phase (Phase 2) aims to identify which kind of SoS (i.e., directed, acknowledged, collaborative, or virtual) the systems represented by the reference architecture are. In this thesis, HSH systems were considered as collaborative SoSs, since each constituent system maintains its operational and managerial independence, but they cooperate with the HSH system to achieve and evolve global missions. Examples of constituent systems are smart homes, domotic systems, electronic health records (EHR) systems, monitoring systems, smart devices, and rehabilitation systems. Technical details of HSH systems and the variety of their possible constituent systems were presented previously in Section 3.2.

The remainder of this section presents the domain models generated through the conduction of Phases 3 to 10 of the process presented in Figure 8. Domains models were designed using the mKAOS tool (SILVA; BATISTA; OQUENDO, 2015).

Missions for HSH Systems

In the Phase 3, refinement and abstraction strategies were applied in parallel and iteratively to establish a hierarchical missions tree of HSH systems for considering them in *HomecARe*. These strategies were proposed initially in the GORE (Goal Oriented Requirement Engineering) area to facilitate the definition of goals and requirements of software systems (LAMSWEERDE, 2001). An excerpt of the missions tree is depicted in Figure 9. Refinement strategy consisted in the missions identification through the discrimination on HOW high-level missions (e.g., the HSH system satisfying stakeholders functional and non-functional needs in Figure 9) can be reached. Missions refinement is performed until reaching low-level missions that can be addressed by constituent systems. Abstraction strategy aimed to identify high-level missions, justifying WHY low-level missions are important to allow the accomplishment of more abstract missions. An excerpt of low-level missions, related with the high-level mission *Chronic disease managed successfully*, is depicted in Figure 10.

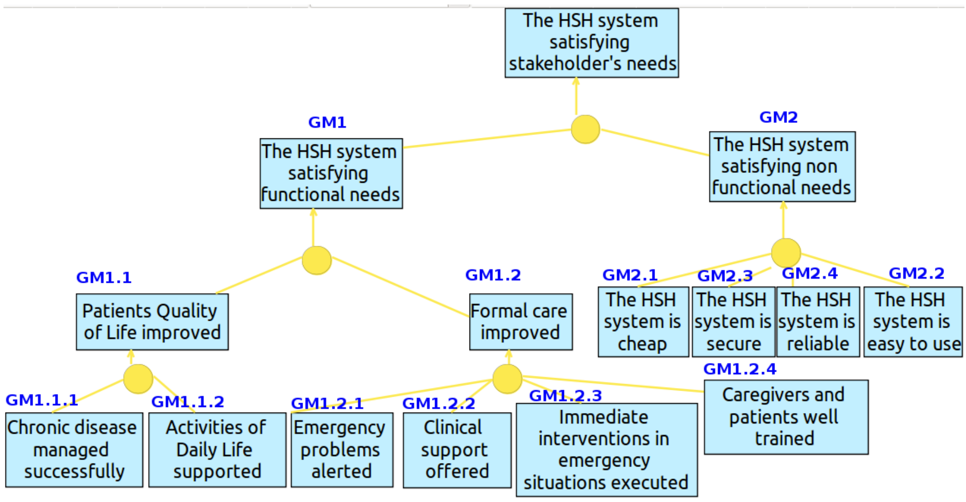


Figure 9 – Excerpt of the mission models of *HomecARe*. Source: (GARCÉS; NAKAGAWA, 2017).

As result, all high-level missions have at least two sub-missions, and all low-level missions are the individual missions that were allocated to at least one type of constituent system (see Table 6). A total of 283 missions was established for *HomecARe*, corresponding to 37 high-level missions for HSH systems and 246 low-level or individual missions for their constituent systems. All diagrams related to missions models of *HomecARe* can be found in (GARCÉS; NAKAGAWA, 2017; GARCÉS; OQUENDO; NAKAGAWA, 2017). Table 5 lists all missions considered in *HomecARe* for both HSH systems (i.e., high-level missions are highlighted in grey colour) and for their constituents (i.e., missions without background colour). The first column contains the mission IDs, which are used as a code to be referred in other models, and the second column describes each mission.

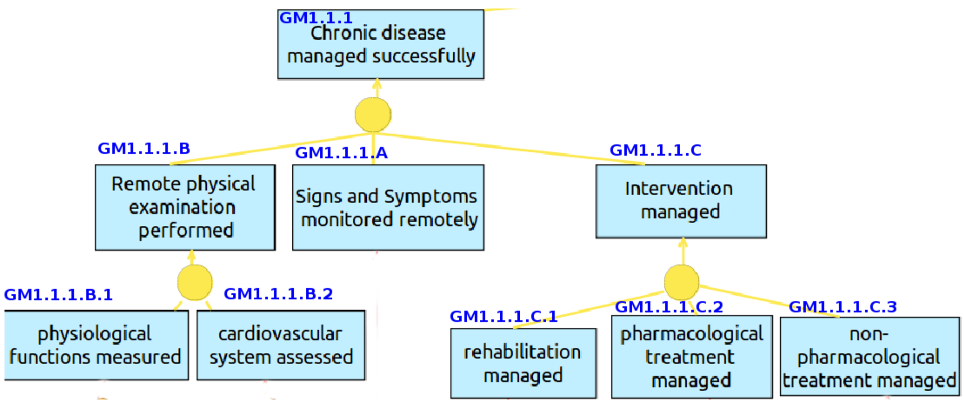


Figure 10 – Excerpt of the mission models of *HomecARe* presenting low-level missions. Source: (GARCÉS; NAKAGAWA, 2017).

It is important to highlight that missions dependencies (refinement and abstraction) were identified based on the study of well-structured clinical ontologies and taxonomies (SIMONET *et al.*, 2009; ECCHER *et al.*, 2006; NAVARRO; RODRÍGUEZ; FAVELA, 2012;

GARCÍA-MAGARIÑO; GÓMEZ-SANZ, 2013; CAMPANA *et al.*, 2008), as well as, guidelines (MINSAUDE, 2013; HAS, 2014; MINSALUD, 2009; CANADA, 2017; QUEENSLAND-GOVERMENT, 2015) representing the adequate knowledge, processes and activities to offer home healthcare services to chronic patients. Moreover, missions models were reviewed by members of the healthcare team, namely, registered and licensed practical nurses.

Table 5 – Full tree missions of *HomecARe*. Coloured missions are of high-level corresponding to HSH systems, and the remainder are individual missions of constituent systems.

Mission ID	Description
GM	**The HSH system satisfying stakeholder's needs.**
GM1	**The HSH system satisfying functional needs.**
GM1.1	Patients Quality of Life improved.
GM1.1.1	Chronic diseases successfully managed.
GM1.1.1.A	Signs and symptoms remotely monitored.
GM1.1.1.A.1	General signs and symptoms monitored.
GM1.1.1.A.2	Neurological, psychological and behavioral signs and symptoms monitored.
GM1.1.1.A.3	Vascular and cardiac signs and symptoms monitored.
GM1.1.1.A.4	Ocular signs and symptoms monitored.
GM1.1.1.A.5	Pulmonary signs and symptoms monitored (e.g., apnea, hyperventilation, cough).
GM1.1.1.A.6	Muscle-skeletical and bone signs and symptoms monitored.
GM1.1.1.B	Remote physical examination performed.
GM1.1.1.B.2	Cardiovascular system assessed.
GM1.1.1.B.3	Nervous system assessed.
GM1.1.1.B.4	Thorax and thoracic organs assessed.
GM1.1.1.B.5	Pheripheral arteries assessed.
GM1.1.1.B.6	Abdomen assessed.
GM1.1.1.B.7	Varicosity and ulcer assessed.
GM1.1.1.B.8	Presure ulcer assessed.
GM1.1.1.C	Intervention managed.
GM1.1.1.C.1	Rehabilitation managed.
GM1.1.1.C.1.1	Motor rehabilitation (physical therapy) assisted.
GM1.1.1.C.1.2	Language rehabilitation assisted.
GM1.1.1.C.1.3	Cognitive rehabilitation assisted.
GM1.1.1.C.1.4	Occupational therapy assisted.
GM1.1.1.C.2	Pharmacological treatment managed.
GM1.1.1.C.3	Non-pharmacological treatment managed.
GM1.1.1.C.3.1	Patient positioning assessed.
GM1.1.1.C.3.2	Environmental interventions performed.
GM1.1.1.C.3.3	Dietetic prescriptions performed.
GM1.1.2	Activities of Daily Life supported.
GM1.1.2.A	Patient's environment monitored.
GM1.1.2.B	Social activities assisted.
GM1.1.1.B.1	Write assisted.
GM1.1.1.B.2	Calls to relatives assisted.
GM1.1.1.B.3	Check email assisted.
GM1.1.1.B.4	Internet social networking assisted.
GM1.1.2.C	Personal care activities assisted.
GM1.1.1.C.1	Take a shower assisted.
GM1.1.1.C.2	Eat assisted.
GM1.1.1.C.3	Physical exercise assisted.
GM1.1.1.C.4	Go to toilet assisted.
GM1.1.1.C.5	Sleep monitor.

Continued on next page

Table 5 – *Continued from previous page*

Mission ID	Description
GM1.1.2.D	Leisures activities assisted.
GM1.1.1.D.1	Read assisted.
GM1.1.1.D.2	Watch tv assisted.
GM1.1.1.D.3	Game assisted.
GM1.1.2.E	Domestic activities assisted.
GM1.1.1.E.1	Cook assisted.
GM1.1.1.E.2	Clean assisted.
GM1.1.1.E.3	Open doors/windows assisted.
GM1.2	Formal care improved.
GM1.2.1	Emergency problems alerted.
GM1.2.1.A	Activity of daily life problem alerted.
GM1.2.1.B	Sign and symptom abnormality alerted.
GM1.2.1.C	Intervention problem alerted.
GM1.2.1.D	Physical examination abnormality alerted.
GM1.2.2	Patient healthcare profile managed.
GM1.2.3	Immediate interventions in emergency situations executed.
GM1.2.3.A	Emergency services contacted.
GM1.2.3.B	Relatives contacted.
GM1.2.3.C	Carers contacted.
GM1.2.4	Caregivers and patients well trained.
GM2	**The HSH system satisfying non-functional needs.**
GM2.1	The HSH system is cheap.
GM2.1.1	The HSH system allowing multiple patients disease management with the same infrastructure.
GM2.1.1.A	Multi purpose sensing platform offered.
GM2.1.1.B	Constituent systems adapted dynamically to patients profiles.
GM2.1.2	Constituent systems services and devices are cheap.
GM2.1.3	Low cost at maintaining or evolving the HSH system offered.
GM2.1.3.A	Constituent systems are automatically integrated.
GM2.1.3.B	New missions are automatically configured.
GM2.1.3.C	Constituent systems must be interoperables.
GM2.1.3.C.1	Constituent systems data must follow well-stablished interoperation standards (i.e., OpenEHR).
GM2.2	The HSH is easy to use.
GM2.2.1	User interfaces adapted to patients profiles.
GM2.2.1.A	Intuitive user interfaces provided.
GM2.2.1.B	User interfaces adapted in interaction time.
GM2.2.1.C	Authentication mechanisms easy to use.
GM2.2.1.D	Highly responsive user interfaces.
GM2.2.1.E	Health information presented in a meaningful way.
GM2.2.2	Physical measures obtained in a comfortable way.
GM2.2.2.A	Non invasive constituent systems.
GM2.2.2.B	Constituent systems easy to use.
GM2.3	The HSH system is secure.
GM2.3.1	Patient's data confidentially managed.
GM2.3.1.A	Patient data are protected.
GM2.3.1.A.1	Stored patient data protected.
GM2.3.1.A.2	Transferred patient data protected.
GM2.3.1.B	Unauthorized access to patient's data prevented.
GM2.3.1.B.1	Constituent systems services authenticated.
GM2.3.1.B.2	Users authenticated.
GM2.3.2	Non-repudiation mechanisms implemented.
GM2.4	The HSH system is reliable.
GM2.4.1	The HSH system is robust.

Continued on next page

Table 5 – *Continued from previous page*

Mission ID	Description
GM2.4.1.A	Fault-tolerant mechanisms implemented.
GM2.4.1.B	Bottlenecks in constituents communication prevented.
GM2.4.1.C	Robust constituent systems operations.
GM2.4.1.D	Robust communication infrastructure.
GM2.4.2	Reliable patient data collected.
GM2.4.2.A	Trusted constituent systems.
GM2.4.2.B	Trusted communication infrastructure.

Responsibilities of Constituent Systems

During the conduction of Phase 4 of our process, low-level or individual missions, i.e., leaf nodes missions in the missions model, were allocated to abstract constituent systems. In short, an abstract constituent is a generalization of a set of possible constituent systems that can realize a specific low-level mission. For instance, blood pressure, heart rate and respiratory rate monitors are specializations of a *physiological functions monitor*, since each of them is responsible to measure specific body functions. Figure 11 depicts an excerpt of responsibilities model, detailing the abstract constituent systems (e.g., physiological functions monitor, cardiovascular system monitor, and signs and symptoms diagnostic system) responsible of achieving low-level missions (e.g., GM1.1.1.B.1, GM1.1.1.B.2, and GM1.1.1.A). For instance, the low-level mission of *GM1.1.1.B.1 - physiological functions measured* is under responsibility of the abstract constituent system *physiological functions monitor*; therefore, for a HSH system achieve this mission, it is required the participation of constituents offering patient's physiological functions information, for instance, blood pressure (GM1.1.1.B.1.1), heart rate (GM1.1.1.B.1.2), or respiratory rate (GM1.1.1.B.1.3). Missions and their codes were detailed in Table 5.

An individual mission must be of responsibility of only one abstract constituent system. However, one abstract constituent system can have several individual missions under its responsibility. Abstract constituent systems composing HSH systems are detailed in Table 6. Moreover, codes of individual missions (previously detailed in Table 5), which are expected to be achieved for each abstract constituent system are listed in the last column of Table 6.

Table 6 – Abstract constituent systems and their responsibilities in HSH systems.

ID	Name	Responsibility	HSH mission
CS001	General signs and symptoms diagnostic system	To detect and inform any sign or symptom related with changes in health function experienced by a patient (e.g., abdominal pain, vertigo)	GM1.1.1.A.1.1 - GM1.1.1.A.1.31
CS002	Neurological signs and symptoms diagnostic system	To detect and inform any condition (e.g., aggression, convulsions) caused by a dysfunction in part of the brain or nervous system.	GM1.1.1.A.2.1 - GM1.1.1.A.2.71

Continued on next page

Table 6 – *Continued from previous page*

ID	Name	Responsibility	HSH mission
CS003	Vascular and cardiac signs and symptoms diagnostic system	To detect and inform any sign or symptom (e.g., arrhytmia, bradycardia) associated with vascular and cardiac system.	GM1.1.1.A.3.1 - GM1.1.1.A.3.7
CS004	Ocular signs and symptoms diagnostic system	To detect and inform any sign or symptom (e.g., blindness, saccade) related with ocular system	GM1.1.1.A.4.1 - GM1.1.1.A.4.8
CS005	Pulmonary signs and symptoms diagnostic system	To detect and inform any irregular pulmonary condition (e.g., cough, apnea)	GM1.1.1.A.5.1 - GM1.1.1.A.5.12
CS006	Muscle-skeletical signs and symptoms diagnostic system	To detect and inform any problem regarding muscles and bones (e.g., muscle cramps, abnormal posture)	GM1.1.1.A.6.1 - GM1.1.1.A.6.11
CS007	Physiological functions monitor	To measure and inform physiological functions (e.g., blood pressure, heart rate)	GM1.1.1.B.1.1 - GM1.1.1.B.1.12
CS008	Cardiovascular system monitor	To measure and inform cardiovascular functions (e.g., heart sound, gallop rhythm)	GM1.1.1.B.2.1 - GM1.1.1.B.2.7
CS009	Nervous system monitor	To calculate nervous system measures (e.g., gait, tremor)	GM1.1.1.B.3.1 - GM1.1.1.B.3.12
CS010	Thorax and thoracic organs monitor	To measure and inform thorax and thoracic functions (e.g., percussion, breathing sounds)	GM1.1.1.B.4.1 - GM1.1.1.B.4.6
CS011	Pheripheral arteries monitor	To measure and inform pheripheral arteries functions (e.g., carotid murmur, femoral pulse)	GM1.1.1.B.5.1 - GM1.1.1.B.5.4
CS012	Abdomen monitor	To measure and inform abdomen status (e.g., bowel sounds, abdominal pain)	GM1.1.1.B.6.1 - GM1.1.1.B.6.7
CS013	Varicosity and ulcer monitor	To inform varicosity and leg ulcers	GM1.1.1.B.7.1 - GM1.1.1.B.7.2
CS014	Pressure ulcer monitor	To inform pressure ulcer status (e.g., location, dimension, length)	GM1.1.1.B.8.1 - GM1.1.1.B.8.11
CS015	Rehabilitation management system	To manage rehabilitation (e.g., motor, language, cognitive rehabilitations or occupational therapy) treatments.	GM1.1.1.C.1.1 - GM1.1.1.C.1.4
CS016	Pharmacological treatment management system	To assist the management of pharmacological treatment.	GM1.1.1.C.1.4
CS017	Non-pharmacological treatment management system	To assist the management of non-pharmacological treatment (e.g., dietetic prescriptions, patient positioning, environmental interventions).	GM1.1.1.C.3.1 - GM1.1.1.C.3.3
CS018	Social activities management system	To assist social activities, such as, writing and social networking	GM1.1.1.B.1 - GM1.1.1.B.4
CS019	Personal care activities support system	To assist personal care activities at home, such as, taking a shower, eating, physical exercising.	GM1.1.1.C.1 - GM1.1.1.C.5
CS021	Leisure activities support system	To assist leisure activities, such as, reading, watching tv, gaming.	GM1.1.1.D.1 - GM1.1.1.D.3
CS022	Domestic activities support system	To assist domestic activities, such as, cooking, cleaning, open and closing doors or windows.	GM1.1.1.E.1 - GM1.1.1.E.3
CS023	Emergency system	To alert problems (i.e., problems regarding patient's activities, detected signs and symptoms, physical examintions abnormalities) to emergency systems	GM1.2.1.A - GM1.2.1.D; GM1.2.3
CS024	Electronic Health Record system	To assist the management of patient healthcare profile.	GM1.2.2
CS025	Homecare training system	To support training for disease management of patients and caregivers.	GM1.2.4

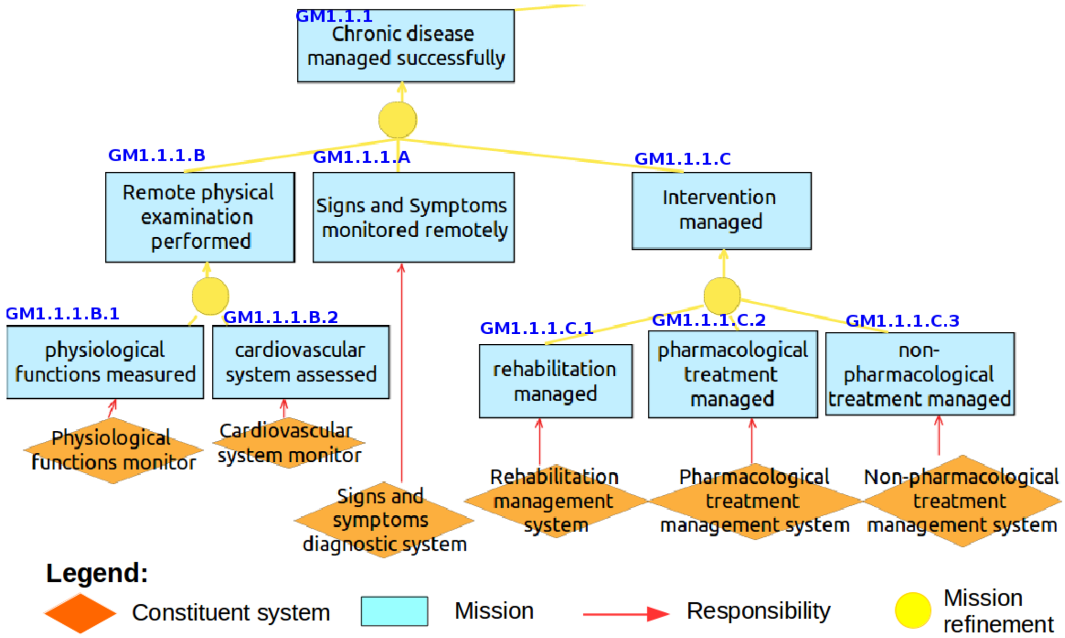


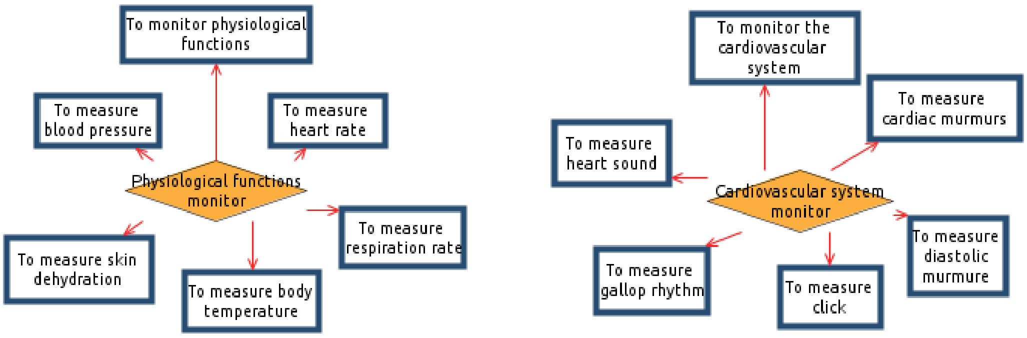
Figure 11 – Responsibilities allocation of low-level missions to abstract constituent systems. Source: (GARCÉS; NAKAGAWA, 2017).

Operational Capabilities of Constituent Systems

In the Phase 5, operational capabilities of each abstract constituent system were modelled. An operational capability is associated to a responsibility (of an abstract constituent system), which is expected by the HSH system to perform its missions. Since HSH systems are collaborative SoS, i.e., they are operational independent of the HSH system, all capabilities offered by these constituents can not be entirely defined. Hence, capabilities model for each abstract constituent system contains just the expected responsibilities for each system. Figure 12 presents the capabilities models for two abstract constituent systems, namely, *Physiological Functions Monitor* (CS007), and *Cardiovascular System Monitor* (CS008). For example, instances of the abstract constituent system *Cardiovascular System Monitor* are expected to contribute with HSH systems offering capabilities to measure patient's heart sound, cardiac murmurs, gallop rhythm, click, diastolic murmur, and to monitor the entire patient's cardiovascular system. Entire capabilities models for all abstract constituent systems can be consulted in (GARCÉS; OQUENDO; NAKAGAWA, 2017).

Entities Model

In Phase 6, data entities and events required for HSH systems operations (i.e., behaviours that allow to achieve missions) were modelled. Entities implement data structures that represent elements of the HSH system and patient's situation, and events are related to specific circumstances for which HSH system must react (e.g., alerts on patient health or environmental problems). Figure 13 presents the entities and events model defined in *HomecARe*.



(a) Capabilities model for a *Physiological Functions Monitor*.

(b) Capabilities model for a *Cardiovascular System Monitor*.

Figure 12 – Capabilities models for abstract constituent systems in *HomecARe*. Source: (GARCÉS; OQUENDO; NAKAGAWA, 2017).

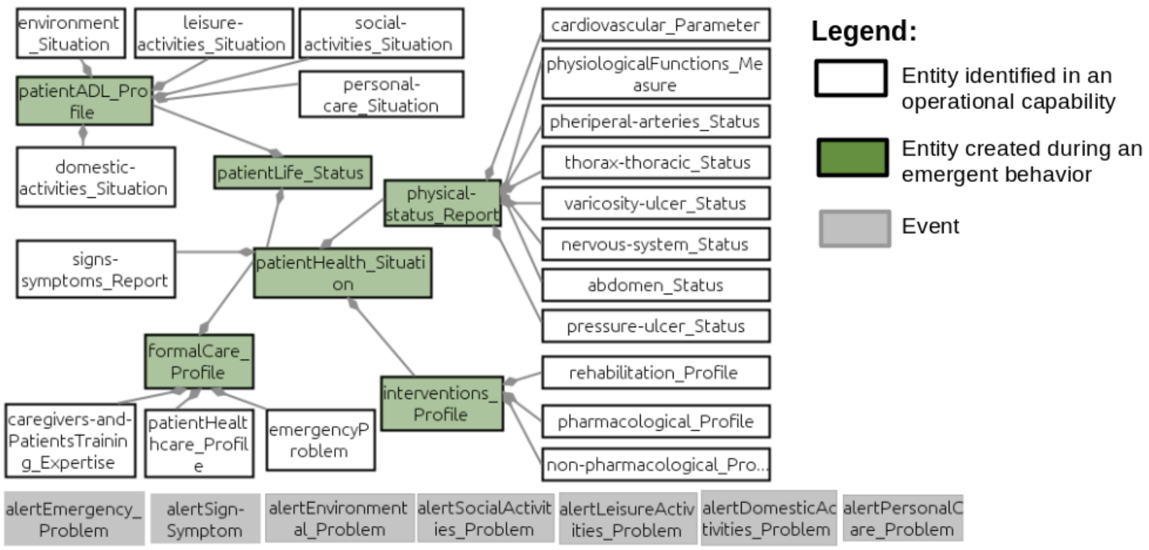


Figure 13 – Entities and events defined in *HomecARe*. Source: (GARCÉS; OQUENDO; NAKAGAWA, 2017).

For each operational capability expected to be provided by each abstract constituent system, data entities (showed as white colour boxes) were identified and modelled (GARCÉS; NAKAGAWA, 2017). For example, the entity *cardiovascular_Paramenter* is provided by operational capabilities offered by the abstract constituent system *Cardiovascular system monitor (CS008)*.

Moreover, during HSH systems operations are expected some emergent behaviours requiring and providing additional data entities, which are presented as green colour boxes in Figure 13. Those entities are needed to accomplish HSH systems missions. For instance, to support the mission of *CM1.1.1.C - Intervention managed* it is required the establishment of patient's interventions profile, i.e., the creation of the data entity *interventions_Profile*, that depends of three data entities (i.e., *rehabilitation_Profile, pharmacological_Profile*, and *non-*

pharmacological_Profile) provided, respectively, by the abstract constituent systems: *CS015 - rehabilitation management system*, *CS016 - pharmacological treatment management system*, and *CS017 - non-pharmacological treatment management system*.

Finally, events (depicted as grey colour boxes) can be notified by abstract constituent systems, or as result of emergent behaviours during HSH systems operations.

Emergent Behaviours of HSH Systems

In Phase 7, the identification and modelling of possible emergent behaviours was made. Missions of HSH systems are fulfilled through the emergence of positive behaviours, i.e., emergent behaviours that contribute to fulfil HSH systems missions. These type of behaviours are consequence of interactions among constituent systems; therefore, emergent behaviours can not be designated as an operational capability of individual constituent systems. In this perspective, the knowledge about an emergent behaviour is spread across the entire SoS and exists only at the macro-level, this is, at realizing high-level missions (MITTAL; RAINEY, 2015). In consequence, it is possible new knowledge exists at a higher level of abstraction itself and there are not enough means to break this new knowledge to be attributed to any constituent system capability (MITTAL; RAINEY, 2015).

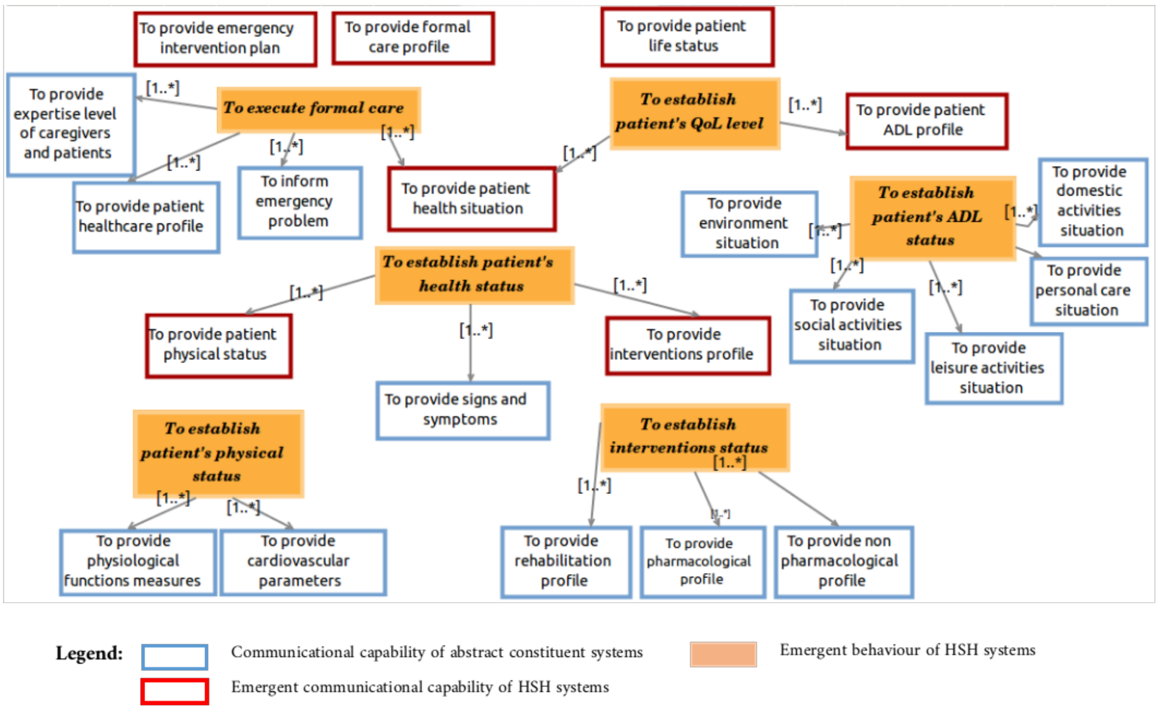


Figure 14 – Expected emergent behaviours of HSH systems defined in *HomecARe*. Source: (GARCÉS; OQUENDO; NAKAGAWA, 2017).

In the context of HSH systems, some emergent behaviours can be defined at design time. Most of these behaviours are considered weak, simple or strong, and can be predicted using deterministic or stochastic approaches (MITTAL; RAINEY, 2015). In HSH systems, emergent behaviours are related to: (i) the aggregation of data, e.g., construction of patient profiles obtaining data from different constituent systems, as detailed in the entities model, Figure 13; and (ii) the prediction of emergency situations, represented as events, e.g., the inference of patient injuries, falls or lost of consciousness. Emergency situations can be inferred using different measures provided by heterogeneous constituent systems through stochastic approaches, such as machine learning or Hidden Markov models (AMIRIBESHELI; BENMANSOUR; BOUCHACHIA, 2015). Figure 14 presents the expected emergent behaviours, defined at the reference architecture level, that are required to accomplish each HSH systems missions. For instance, for a HSH system to address its mission of *GM1.1.1.B - Remote physical examination performed*, the system must to establish and communicate the patient's physical status (which is considered as emergent behaviour), which also contributes to establish patient's health status, allowing the achievement of the mission *GM1.1.1 - Chronic disease managed successfully*. Therefore, the emergence of multiple (sometimes) dependent behaviours permits to reach HSH systems higher missions.

Communicational Capabilities of Constituent Systems

The Phase 8 of our process orients the establishment and modelling of communicational capabilities of abstract constituent systems. To allow emergent behaviours, constituent systems need to interact among them and with the HSH system. This type of interactions are considered as communicational capabilities for those systems. For each entity provided by each abstract constituent systems, it was associated a communicational capability to represent constituents involved in the entity sending and reception. Since these interactions represent processes that need to be supported by HSH systems, stakeholders activities, domain guidelines, and documentation of some HSH systems identified in Section 3.2 were important information sources to identify interactions between constituent systems.

Figure 15 presents an example of communicational capability model in *HomecARe*, which represents interactions among two abstract constituent systems, namely, *CS001 - Signs and symptoms diagnostic system*[1] and *CS024 - Electronic health record system*, that are participating in a HSH system. The CS001 is responsible for creating the entity *signs-symptoms_Report*. For this, CS001 uses the entity *physical-status_Report* that is created by the HSH system through the emergence of the behaviour *to establish patient's physical status*. These entities, *physical-status_Report* and *signs-symptoms_Report*, together with the *interventions_Profile* are communicated to CS024, to establish and communicate the *patientHealthcare_Profile*. The entire models associated to other emergent behaviours can be consulted in (GARCÉS; OQUENDO; NAKAGAWA, 2017).

[1] The *Signs and symptoms diagnostic system* is an abstraction of constituent systems CS001 - CS006 in Table 6

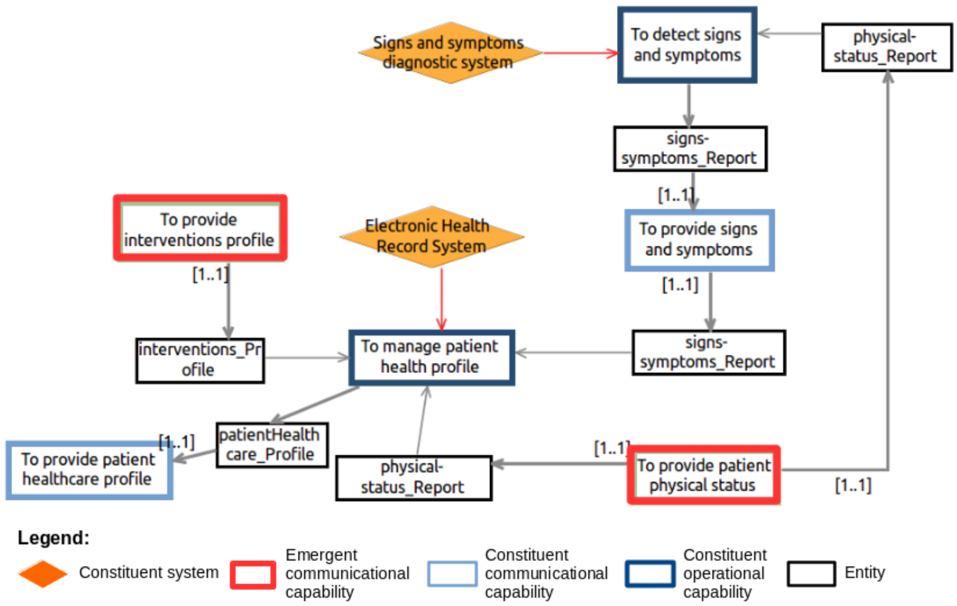


Figure 15 – Communicational capabilities involved to manifest the emergent behaviour *To provide patient healthcare profile*. Source: (GARCÉS; OQUENDO; NAKAGAWA, 2017).

Analysing the Gap and Validating the Domain Models

In Phase 9, there were analysed possible gaps between all domain models, presented in Figures 9 to 15, for ensuring their consistency in *HomecARe*. Therefore, the following mappings were performed (GARCÉS; NAKAGAWA, 2017): (i) *Emergent behaviour - missions models:* to check if for each high-level mission, *HomecARe* exhibits an emergent behaviour that allows its execution; (ii) *Communicational capabilities - Emergent behaviours models:* to verify if all communicational capabilities required to trigger all emergent behaviours were considered; (iii) *Abstract constituent system - Communicational capabilities models:* to ensure that for each communicational capability it was allocated just one abstract constituent system responsible for it; (iv) *Entity/event - Operational/communicational capabilities models:* to determine if all entities/events defined in *HomecARe* were considered in at least one operational/communicational capability; (v) *Entities/events - Objects models:* to examine if all entities/events established in *HomecARe* were defined in the object model; and (vi) *Individual mission - Operational capability models:* to check if each individual mission was designated under responsibility of at least one constituent system.

Finally, in the last phase of our process, Phase 10, it was investigated the viability of domains models established in *HomecARe* to support the domain analysis for concrete HSH systems. Therefore, those models were instantiated to represent real scenarios as those defined in (CAMARINHA-MATOS *et al.*, 2011). Optimist results were obtained from using *HomecARe* models to identify missions, constituents (and their operational and communicational capabilities), data and events, and possible emergent behaviours to define HSH systems for those real scenarios.

4.2. Architecturally Significant Requirements of *HomecARe*

The architectural significant requirements (ASRs) considered in *HomecARe* were established based on domain knowledge obtained through the analysis of several information sources (as presented in Chapter 3), the representation of such knowledge in domain models (exhibited in Section 4.1), and the selection of important quality attributes requirements for HSH systems based on the QM4AAL (described in Section 3.4).

Table 7 lists requirements that were considered important for the construction of *HomecARe*. The ASRs of *HomecARe*, are an abstraction of missions (both functional and non-functional) specified in Section 4.1, Table 5. Hence, this version of *HomecARe* is focused to support only the achievement of those missions, leading other requirements, e.g., performance, for further versions. In this perspective, ASRs of *HomecARe* were classified in five categories: (i) **Domain requirements (DR)**, detailing functionalities that systems resulted from the reference architecture must accomplish to achieve HSH systems missions; (ii) **Adaptivity requirements (AR)**, specifying modifications and reconfigurations that HSH systems must done to allow the correct accomplishment of their missions; (iii) **Security requirements (SR)**, establishing constraints on how HSH systems must manage patient information; (iv) **Interoperability and integration requirements (IIR)**, describing capacities that HSH systems must consider to allow interoperation of their constituent systems; and (v) **Reliability requirements (RR)**, enumerating means that HSH systems must offer to allow reliable interactions between their constituent systems.

Table 7 – Architecturally significant requirements of *HomecARe* .

ID	Description
	Domain requirements (DR)
DR01	The reference architecture must enable the development of HSH systems that, remotely, estimate and provide the patient physical status at any time, e.g., informing the status of patient through the analysis of his/her physiological functions and body systems (e.g., cardiovascular, nervous, respiratory) signs and symptoms.
DR02	The reference architecture must enable the development of HSH systems that, remotely, establish and provide the status of patient interventions at home, e.g., informing the status of patient pharmacological or rehabilitation treatments.
DR03	The reference architecture must enable the development of HSH systems that, remotely, establish and provide patient health situation at any time, based on knowledge of current situation of patient physical status, sign and symptoms and interventions.
DR04	The reference architecture must enable the development of HSH systems that estimate and provide the status of patient activity of daily life, e.g., consolidate status of patient environment, social activities, personal care activities (i.e., taking a shower, eating, exercising), and domestic activities (i.e., cleaning, cooking).
DR05	The reference architecture must enable the development of HSH systems that establish and provide the level of patient quality of life, based on his/her health status and activities of daily life situation.
DR06	The reference architecture must enable the development of HSH systems that allow the identification of emergency situations regarding patient health status and his/her environment situation.

Continued on next page

Table 7 – *Continued from previous page*

ID	Description
DR07	The reference architecture must enable the development of HSH systems that establish communication with emergency services (e.g., ambulance, caregivers, fire department, police department) when an emergency situation occurs, according with pre-defined emergency interventions plan.
DR08	The reference architecture must enable the development of HSH systems that allow patient monitoring, independently of his/her autonomy level.
	Adaptivity requirements (AR)
AR01	The reference architecture must enable the development of HSH systems that allow monitoring new signs and symptoms for a patient without the need of manual configuration.
AR02	The reference architecture must enable the development of HSH systems that allow modifications of pharmacological treatments without the need of manual intervention.
AR03	The reference architecture must enable the development of HSH systems that allow modifications of rehabilitation treatments without the need of manual intervention.
AR04	The reference architecture must enable the development of HSH systems that allow modifications of non-pharmacological treatments (e.g., patient diet) without the need of manual intervention.
AR05	The reference architecture must enable the development of HSH systems that allow modifications of patient activities to be monitored without the need of manual intervention.
AR06	The reference architecture must enable the development of HSH systems that allow modifications of emergency procedures without the need of manual intervention.
AR07	The reference architecture must enable the development of HSH systems that allow modifications of clinical guidelines to manage patient diseases without the need of manual intervention.
AR08	The reference architecture must enable the development of HSH systems that execute reconfigurations of their architectures when services provided by constituent systems present malfunctioning or unavailability, without the need of manual intervention.
AR09	The reference architecture must enable the development of HSH systems whose modifications in their behaviours do not impact the functionality of their constituent systems.
	Security requirements (SR)
SR01	The reference architecture must enable the development of HSH systems that protect patient information from unauthorized access.
SR02	The reference architecture must enable the development of HSH systems that implement non-repudiation mechanism for users and constituent systems.
	Interoperability and integration requirements (IIR)
II01	The reference architecture must enable the development of HSH systems that allow interoperable communication between constituent systems that are distributed, heterogeneous, and operational and managerial independent.
II02	The reference architecture must enable the development of HSH systems that allow coordination and collaboration between constituent systems following standards for interoperability in health care, e.g., HL7 v2, HL7 v3, SNOMED, ICD, AIS, LOINC and ISO 13606.
II03	The reference architecture must enable the development of HSH systems that allow interoperable coordination and collaboration between constituent systems to execute procedures established in clinical guidelines for chronic disease management.
	Reliability requirements (RR)
RR01	The reference architecture must enable the development of HSH systems that offer fault-tolerant mechanisms for constituent systems interactions.
RR02	The reference architecture must enable the development of HSH systems that offer fault-tolerant mechanisms for control message exchanging.
RR03	The reference architecture must enable the development of HSH systems that avoid bottlenecks in constituent systems communications.
RR04	The reference architecture must enable the development of HSH systems that avoid bottlenecks in control message exchanging.
RR05	The reference architecture must enable the development of HSH systems that offer trusted operations (i.e., requesting, providing, translating, encrypting, sending, reading and storing) on patient information.

4.3. Final Considerations

Domain knowledge presented in Chapter 3 was the basis to conduct the architectural analysis for *HomecARe*, which was presented in this chapter. As result of this analysis, important domains models were defined, representing missions (i.e., 37 high-level missions were defined for HSH systems, and 246 low-level were allocated for their constituent), operational capabilities of constituents, required data and events, possible emergent behaviours, and communicational capabilities of HSH systems and their constituents. All models were defined following a ten-phases systematic process to establish and validate domain models in reference architectures of SoS (GARCÉS; NAKAGAWA, 2017). Based on the consolidated knowledge about how to provide high-quality health care at home for chronic patients, the ASRs of *HomecARe* were established to ensure high-level missions of HSH systems. Both domain models as ASRs were used to define the architectural synthesis of *HomecARe*, whose results are presented in next chapter.

CHAPTER

5

ARCHITECTURAL SYNTHESIS OF *HOMECARE*

In this chapter, the architectural synthesis of *HomecARe* resulted from conducting the Step 3 of ProSA-RA is presented. *HomecARe* is described as a set of architectural views that gives understanding about how architectural decisions (e.g., architectural patterns, styles, and tactics) in *HomecARe* allow to accomplish its ASRs, and as consequence, HSH system missions (described in Section 4.1). Views were established following three viewpoints, namely, conceptual, missions, and quality viewpoints. The *conceptual viewpoint*, in Section 5.1, represents all required architectural elements and the layered structure adopted in *HomecARe* to allow the construction of HSH systems. The *missions viewpoint*, in Section 5.2, details the essential architectural configurations in *HomecARe* to allow HSH systems to accomplish their missions. The *quality viewpoint*, in Section 5.3, describes how non-functional requirements of reliability, security, interoperability, and adaptivity, are achieved by architectural decisions presented in *HomecARe*. This chapter ends with some final considerations in Section 5.4.

5.1. Conceptual Viewpoint

The conceptual viewpoint of *HomecARe* aims to describe, in a general way, important concepts of the HSH domain that were considered in the reference architecture. *HomecARe* is based on the Service-Oriented Solution Stack (S3) reference architecture (ARSANJANI *et al.*, 2007). In *HomecARe* , the bottom layer contains services provided by constituent systems and that are of interest of HSH systems to achieve their missions. In this perspective, the operational systems and service component layers defined in the S3 architecture were not considered in *HomecARe*, since they are related, respectively, to IT operating environment (e.g., package applications and solutions running in an enterprise environment) where a constituent system is being executed and to components implementations realizing services offered by a constituent system (ARSANJANI *et al.*, 2007). These both layers were not contemplated due to in SoS it is not possible to understand the internal structure and executing environment of most of constituent systems, since they are, by nature, managerial and operational independent from the SoS. The conceptual viewpoint is composed only of a layered view that was structured following S3 and it is detailed as follows.

Layered View

HomecARe is structured in seven layers as presented in Figure 16: (i) *Services layer* composed of two sub-layers, the *Constituent systems services layer* that contains all services offered by constituent systems and that are relevant to HSH systems, and the *Control services layer* comprising services to allow orchestration of other services; (ii) *Business process layer* that encompasses composed services to realize business processes; (iii) *Consumer layer* that contains services to handle interaction with users, i.e., through their interaction with services in the *Home healthcare team layer* and *System management layer*, and with other systems in the e-Health ecosystem, i.e., through services located in the *Health organizations layer*; (iv) *Health Service Bus layer* that integrates services contained in the aforementioned first three layers; (v) *Quality of service layer* that monitors, logs, and signs non-compliance of quality in HSH systems; (vi) *Information architecture layer* that contains all information, knowledge or meta data needed in HSH systems to achieve their missions; this layer is subdivided into the *Interoperation standards layer* and *Healthcare plans layer*; and (vii) *Governance and policies layer* that covers guidance and policies for managing HSH systems operations. Services identification (ID) are used in the remainder of this chapter to facilitate cross references between viewpoints. Specifically, the prefix **CSSv**, **CSv**, and **BPSv** are used to identify *Constituent systems services*, *Control services*, and *Business process services*, respectively. More details of each layer are provided in the remainder of this section.

Services Layer

It is the lowest level layer, containing two types of services, named constituent systems services and control services. *Constituent Systems Services* (identified with the prefix CSSv), also referred as basic services in S3, consist of all services that constituent systems offer and that are needed to perform HSH systems missions. HSH missions were introduced in Section 4.1. Services of this type implement abstract specifications of the operational capabilities detailed in Section 4.1. Table 8 lists the 12 types of services provided by constituent systems. First column specifies the ID of each service (CSSv), followed by its name (also showed in Figure 16) and a description of capabilities realized by each service. Last column details data classes from the standard SNOMED-CT that represent data communicated by each service.

It is important to highlight that, due to such services are offered by constituent systems, they are not under the control of HSH systems, but they operate and are managed independently

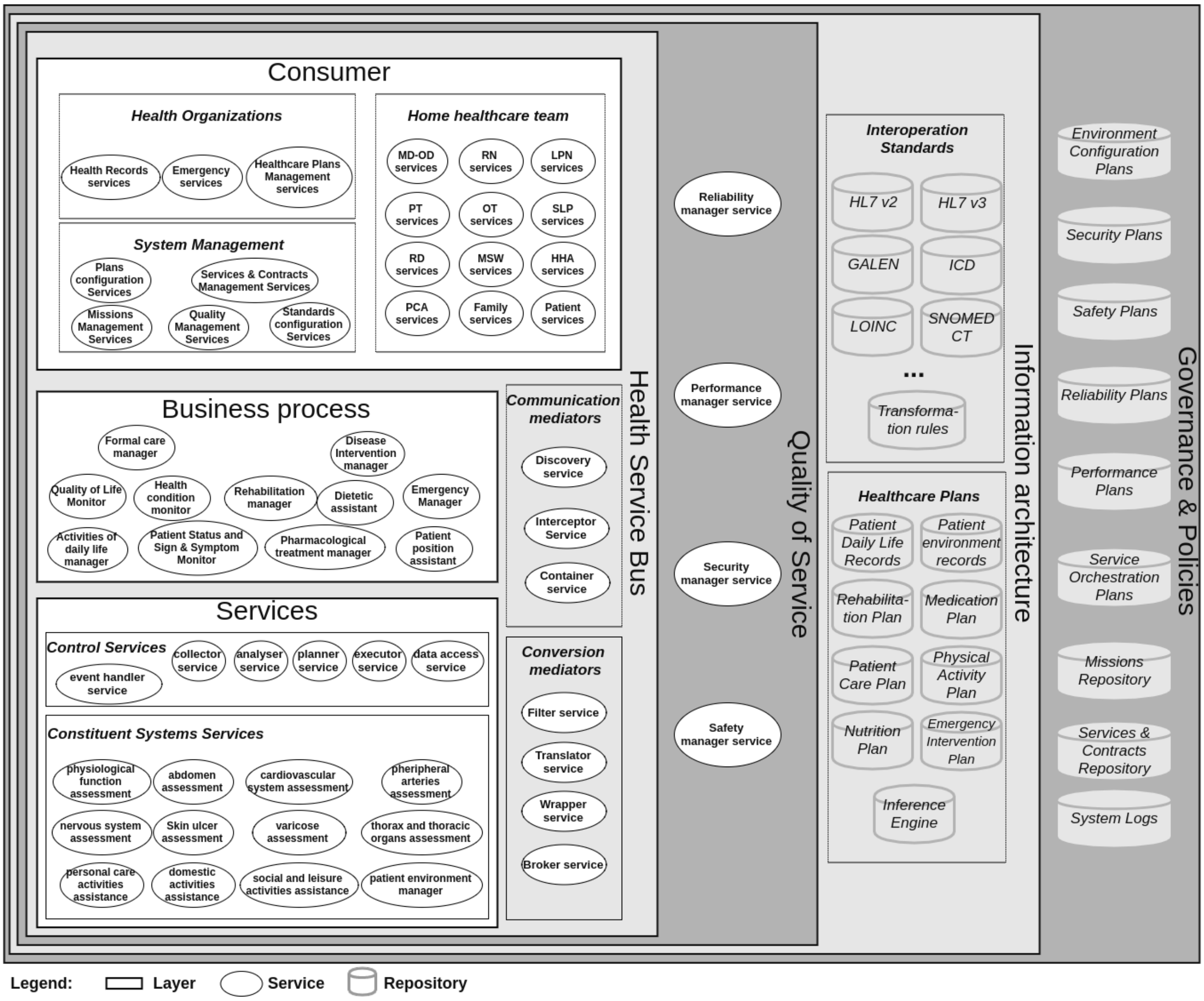


Figure 16 – Layered View of *HomecARe*.

by external organizations.

Table 8 – Constituent Systems Services in *HomecARe*.

Service ID	Service name	Description	SNOMED-CT ID
CSSv001	Physiological function assessment	To offer measures on patient vital signs, such as, body temperature (BT), blood pressure (BP), pulse or heart rate (HR), and respiration rate (RR).	363789004 - General characteristic of patient (observable entity).
CSSv002	Cardiovascular system assessment	To offer measures on patient cardiovascular functions, such as, heart sound, cardiac flow or systole function.	70337006 - Cardiovascular function (observable entity).
CSSv003	Abdomen assessment	To offer information on abdominal findings such as, abdominal aorta, mass or rigidity.	609624008 - Finding of abdomen; 249273002 - Finding of urinary tract.
CSSv004	Nervous system assessment	To offer measures on patient involuntary movements, such as, excessive blinking, tremors or spasms.	267078001 - Involuntary movement finding.
CSSv005	Varicose assessment	To offer measures on patient varicose or skin ulcer.	271652003 - Varicose vein finding.

Continued on next page

Table 8 – *Continued from previous page*

Service ID	Service name	Description	SNOMED-CT ID
CSSv006	Skin ulcer assessment	To offer measures on patient skin ulcer, such as, pressure ulcer stage, surface area of ulcer.	439744001 - Ulcer observable (observable entity).
CSSv007	Peripheral arteries assessment	To offer measures on findings in systemic arterial, such as, arterial bruit, carotid bruit or pulse.	301139003 - Systemic arterial finding; 54718008 - Peripheral pulse, function (observable entity)
CSSv008	Thorax and thoracic organs assessment	To offer measures regarding abnormal breathing, lung capacity or respiration difficulty.	106048009 - Respiratory finding.
CSSv009	Social and leisure activities assistance	To offer information regarding patient activities in social context, such as, shopping, reading, using telephone.	300574001 - Community living activity (observable entity).
CSSv010	Personal care activities assistance	To offer information on personal care activities, such as, dressing, personal hygiene, or taking medications.	285592006 - Personal care activity (observable entity)
CSSv011	Domestic activities assistance	To offer information on domestic activities, such as, doing housework or preparing meals.	272387007 - Domestic activity (observable entity).
CSSv012	Patient environment manager	It intends to establish the patient environment situation, through monitoring odour, infestation, water supply and other environment characteristics, and define an environment intervention plan.	224153006 - Local environment and neighbourhood details (observable entity). 224249004 - Characteristics of home environment ; 129841008 - Finding related to environmental risk factor.

A second category of services in the Service Layer is related with *Control Services* (identified with the prefix CSv) that offer capabilities to the orchestration of basic services and their composition into business process services. In *HomecARe*, control services were proposed to the construction of autonomic managers as manager services located in the *Business process layer* and *Quality of service layer*. Hence, each control service performs specific functionalities of elements in the MAPE-K pattern. This pattern was introduced in Section 2.2 as an architectural solution for SoS.

In *HomecARe*, CSv are under the control of HSH systems, hence, they can be instantiated or modified at runtime, as reconfigurations may be required during the HSH systems operation. Table 9 details all control services defined in *HomecARe*. First column indicates service ID (CSv) for each control service, followed by its name and description of its capabilities.

Table 9 – Control Services in *HomecARe*.

Service ID	Service name	Description
CSv001	Event handler service	To offer data/event handling functionalities when data is exchanged or events are published in the HSB.
CSv002	Collector service	To collect data from the HSB for monitoring purposes.

Continued on next page

Table 9 – *Continued from previous page*

Service ID	Service name	Description
CSv003	Analyser service	To infer situations based on collected data.
CSv004	Planner service	To interpret situations and define plans to change HSH systems configurations or behaviours.
CSv005	Executer service	To distribute re-configurations plans to involved entities.
CSv006	Data access service	To offer data access to repositories.

Business Process Layer

Services in this layer, called business process services (identified with the prefix BPSv), represent macro-flow activities specified in business processes. Here, a business process is related to those work-flow required to bring healthcare services at patient's home, e.g., activities performed by the healthcare team to successfully manage patient's condition. Business processes were identified and defined based on clinical guidelines, missions, and other domain knowledge obtained during the domain analysis of *HomecARe* as presented in Chapter 3.

BPSv are composed of basic services from the services layer. BPSv realizes the logic to orchestrate basic services to execute a business process. In *HomecARe*, business process services are responsible for performing high-level missions, as defined in Section 4.1. For instance, the service *BPSv010 - Formal care manager* choreographs CSSv, CSv, and other BPSv to achieve the global mission *GM1.2 - Formal care managed*. BPSv are under the control of the HSH systems unlike CSSv that are independently controlled.

Table 10 presents all BPSv defined in *HomecARe*, providing service ID (BPSv), name, description, and missions that are under responsibility of each BPSv. Last column maps type of data managed by BPSv into data classes defined in the standard SNOMED-CT. Moreover, considering the importance of BPSv for achieving HSH systems missions, services architectures for these services are detailed in the missions viewpoint in Section 5.2.

Table 10 – Business Process Services

Service ID	Service name	Description	Related Mission	SNOMED-CT ID
BPSv001	Patient status and sign & symptom monitor	Its aim is to establish patient physical status, through remote patient physical examination,and monitoring her/his sign and symptoms.	GM1.1.B; GM1.1.1.A.	404966002 - Physical ageing status (observable entity)
BPSv002	Activities of daily life manager	It aims to provide the AoDL profile, detailing the status of patient environment and his/er social, personal care, domestic and leisure activities.	GM1.1.2	370885003 - Activities of daily living management

Continued on next page

Table 10 – *Continued from previous page*

Service ID	Service name	Description	Related Mission	SNOMED-CT ID
BPSv003	Rehabilitation manager	Its aim is to establish the status of patient rehabilitation regarding her/his care plan.	GM1.1.1.C.1	722138006 - Physiotherapy
BPSv004	Pharmacological treatment manager	Its aim is to establish the status of patient pharmacological treatment regarding her/his medication plan.	GM1.1.1.C.2	416608005 - Drug therapy.
BPSv005	Dietetic assistant	Its aim is to establish the status of patient diet, regarding her/his nutrition plan.	GM1.1.1.C.3.3	185495006 - In-house dietetics.
BPSv006	Patient position assistant	Its aim is to establish the situation of patient position regarding her/his position intervention plan.	GM1.1.1.C.3.1	225430005 - Procedures relating to mobility.
BPSv007	Health condition monitor	It aims to predict the situation of a patient health condition based on her/his physical status, sign and symptoms, and intervention profile.	GM1.1.1	405157008 - Personal health status (observable entity)
BPSv008	Disease intervention manager	Its aim is to establish the patient intervention profile, considering information of patient rehabilitation, and pharmacological and non-pharmacological interventions.	GM1.1.1.C	386053000 - Evaluation procedure
BPSv009	Quality of life monitor	It aims to establish the patient situation regarding her/his quality of life, considering informations from patient AoDL profile and health conditions.	GM1.1	709503007 - Assessment of quality of life (procedure)
BPSv010	Formal care manager	It intends to consolidate the patient formal care profile that can be analysed by care team for assessing its adequacy to patient situations.	GM1.2	392134007 - Manage health care
BPSv011	Emergency manager	Its purpose is to predict an emergency situation and to establish an emergency plan to be executed.	GM1.2.1; GM1.2.3	281694009 - Finding of a risk; 225314003 - Risk management.

Consumer Layer

Services in the consumer layer offer information about HSH systems to specific three types of stakeholders, namely, health organizations, managers of HSH systems, and members of the home healthcare team. Hence, the consumer layer contains three specialized layers offering services for each type of stakeholder:

- Health Organizations Layer: This layer contains the following services offered by health organizations: (i) *emergency services*, offering pre-hospital services, as ambulance and paramedic services; (ii) *health records services*, allowing access to patient information contained in electronic health records systems; and (iii) *healthcare plans management services*, granting the management of patient care plans. Services contained in this layer allow HSH systems to be part of a broader e-Health ecosystem. In this context, for each service offered by external health organization, a service interface, contract, and protocol that allow its integration to the HSH system must be defined.

- System Management Layer: This layer offers services to manage configurations and modifications of the HSH systems, for instance: (i) *plans configuration services*, offering functionalities to modify plans located in the governance and policies layer, e.g., environment configuration plans; (ii) *services and contracts management services*, allowing the insertion, modification or deletion of specifications of services, contracts, and protocols that are allowed in the HSH systems; (iii) *missions management services*, providing means to add, modify or delete missions specifications of the HSH systems; (iv) *quality management services*, allowing the management of quality plans, i.e., security and reliability plans in the governance and policies layer; and (v) *standards configuration services*, offering functionalities to modify, update, and add interoperation standards and transformation rules located in the information architecture layer. Through services offered in this layer, it is possible to follow an evolutionary development during the HSH systems life cycle, ensuring their sustainability over time.

- Home Healthcare Team Layer: This layer handles interactions with members of the healthcare team. See Section 3.1 for details about types of stakeholders for HSH systems. Home healthcare team services are the following:

- Physician (MD-DO) services, which support analysis of formal patient care, management of patient care plans, such as, pharmacological treatments and consultation of patient profiles following medical specialities requirements;
- Registered Nurse (RN) services, which facilitate the management of patient care, allowing assessment of patient health situations. These services also support monitoring of team members supervised by the RN, such as LPNs, HHAs, and PCAs;
- Licensed Practical Nurse (LPN) services, which support activities performed by LPNs, for instance manual register of patient vital signs, collecting biological samples from patient, and performing of assessment protocols;
- Physical Therapist (PT) services, which assist the PT to record evolutions of rehabilitation therapies and perform assessment protocols of patient physical status;
- Occupational Therapist (OT) services, which orient to OTs for performing their activities when assisting a patient at home through the use of assistive technologies, and guide patients' education about their disabilities;
- Speech-Language Pathologist (SLP) services, which support SLP at managing speech therapy to patients;
- Registered Dietician (RD) services, which allow nutrition plans management and the monitoring of how-well a patient follows the prescribed nutrition plan;

- Medical Social Worker (MSW) services, which support a social worker when offers emotional support or counselling to patients and their families for financial management or community services requesting;
- Home Health Aide (HHA) services, which support HHAs to assess patient self-care activities and her/his independence level;
- Personal Care Aide (PCA) services, which support PCAs to analyse patient environmental conditions and patient independence level at performing domestic activities;
- Family services, which inform patient relatives about patient conditions and emergency situations. Moreover, these services support relative-patient communication; and
- Patient services, which allow patient interactions with the HSH system.

The specialization of services for each member of the home healthcare team can facilitate the specification of profiles, roles, and permissions for each member, improving security mechanisms of authentication and authorization as defined in the quality viewpoint in Section 5.3.

Health Service Bus

The Health Service Bus (HSB) is a specialization of the ESB and is analogous to the integration layer specified in the S3 reference architecture (ARSANJANI *et al.*, 2007). Capabilities offered by the HSB are mediation, routing, and transportation of services messages and events. The HSB also allows message and protocol transformation to grant services interoperability and integration. Moreover, HSB is designed to improve flexibility and evolution of HSH systems, since it is possible to add, remove, modify or reconfigure services during HSH systems operation.

For communication, collaboration, and coordination purposes, HSB defines three mediators: (i) a *discovery service*, responsible for detecting possible providers of services requested by services in the consumer layer (e.g., home healthcare team services), and for verifying if contracts are correctly addressed by participating services; (ii) an *interceptor service*, responsible for routing messages in the HSB, verifying permissions of services over operations and data, and recording all operations occurred in the HSB into the system logs repository; and (iii) a *container service*, which is in charge of updating services registers, such as capabilities offered, implemented standards, availability level, roles, profiles and quality metrics.

For integration and interoperation ends, HSB offers four mediators: (i) a *filter service*, responsible to decode a message and select specific sets of information contained in a message; (ii) a *translator service*, which aims to achieve semantic interoperability, interpreting information contained in a message and translating the information to be interpreted by other participants; (iii) a *wrapper service*, which intends syntactic interoperability, encoding information in specific message format to be understood by other participants; and (iv) a *broker service* that coordinates services offered by filter, translator, and wrapper services to offer a composed service for collecting messages from a provider and transforming and routing messages to consumers in an interoperable way.

To achieve confidentiality and integrity of patient's data, filters, wrappers, and brokers can include operations to implement end-to-end security mechanisms, e.g., encrypting and decrypting of messages. More details of HSB are presented in the *Integration and interoperability view* in Section 5.3. Additional information about mediators can be consulted in Appendix C.

Quality of Service Layer

As defined in the S3 reference architecture (ARSANJANI *et al.*, 2007), the Quality of Service (QoS) layer is responsible for capturing, monitoring, logging, and signalling non compliance with non-functional requirements that relate to the service qualities. This layer observes the service, consumer, and HSB layers and emits events when it detects or anticipates non compliance of reliability, performance, security, and safety requirements. Henceforth, the aim of QoS layer is to ensure the HSH systems meet their non-functional requirements.

In this perspective, *HomecARe* defines several quality managers, i.e., reliability, performance, security and safety managers, which are realized using control services (CSv in Table 9). These managers are responsible for monitoring quality metrics, analysing non compliance of quality specifications, planning and executing required reconfigurations to maintain the desired quality level of the HSH systems, as specified in the quality plans repositories located in the governance and policies layer. Quality managers are presented in detail in the quality viewpoint in Section 5.3.

Information Architecture Layer

This layer contains knowledge, represented as repositories, offering intelligence to HSH systems to achieve their missions. This layer is subdivided into the *Interoperation standards layer* and the *Healthcare plans layer*. In the former layer, protocols for health information exchange are allocated, e.g., HL7, ICD and SNOMED-CT, and transformation rules between protocols are provided. These rules are used by brokers to allow transformation of messages for interoperable purposes. More information about message transformations by brokers is presented in the integration and interoperability view in Section 5.3. The latter layer includes all information required to allow the accomplishment of HSH systems missions, hence, it allocates the business logic that is represented as *Healthcare Plans*. This information can be stored as meta data content, using ontologies or well-defined repositories. Moreover, rules to allow knowledge discovering that enable prediction of HSH systems behaviour (considered as emergent behaviour in the SoS context) are defined in this layer. Additionally, data transformation rules are also placed in this layer.

Governance and Policies Layer

This layer covers all aspects of managing the HSH systems operations life cycle. It provides guidance and policies for managing service-level agreements, including performance, security, and monitoring (ARSANJANI *et al.*, 2007). Therefore, this layer is applicable to all other layers of the reference architecture for HSH systems. Specifically, quality plans, reconfiguration plans, repositories of missions, services and contracts specifications, and system logs are contained in this layer. Modifications of information comprised in this layer can be made using the *System management services* located in the consumer layer.

5.2. Missions Viewpoint

The mission viewpoint proposed for *HomecARe* is composed by six mission views (MV), as presented in Table 11. Each view aims to represent how HSH systems can achieve the high-level missions specified in Section 4.1, and how domain requirements (DR), defined in Section 4.2, are addressed in *HomecARe*. A first decision was to establish specific business process services (BPSv) responsible of executing all required activities to accomplish the missions. Hence, in *HomecARe*, all missions are ensured in a HSH system since it was defined at least one BPSv responsible for the accomplishment of each one. The eleven types of BPSv in *HomecARe* were introduced in Section 5.1. For each BPSv, a MV was defined, representing how a BPSv can be realized through the orchestration or choreography of participating services in HSH systems. A MV is described using the following models:

- **Business process model** gives a dynamic view of the work-flow needed to execute a business process. This diagram establishes which capabilities offered by participants are involved in the business process, and how such capabilities are used by other participants of the each process. A participant in a business process can be any service of those described in the conceptual view, Figure 16, i.e., constituent systems services (CSSv), control services (CSv), business process services (BPSv) and consumer services;
- **Capabilities model** details how required capabilities in a business process are related. Relationships between capabilities are identified using the business process diagram. Similarly, operations and data required to realize a capability are also determined from activities and tasks defined in the business process diagram;
- **Service interfaces model** specifies interfaces, containing the behaviour and semantics, of a service in order to offer a required capability in a business process;

Table 11 – Structure of the Missions Viewpoint of *HomecARe*

ID	Mission View Name	Requirement	Missions	Business Process Service	Models				
					Business Process	**Capabilities**	**Interfaces, Contracts and Protocols**	**Services Architecture**	**Participants**
MV1	Physical Status and Signs & Symptoms Monitored Remotely	DR01	GM1.1.B; GM1.1.1.A	BPSv001	Figure 17	Figure 18	Figures 19 - 24	Figure 25	No
MV2	Patient Intervention Managed	DR02	GM1.1.1.C; GM1.1.1.C.1; GM1.1.1.C.2; GM1.1.1.C.3.3; GM1.1.1.C.3.1	BPSv003; BPSv004; BPSv005; BPSv006; BPSv008	Figure 26	Figure 28	Figures 29 - 38	Figure 39	Figure 27
MV3	Activities of Daily Life Supported	DR04	GM1.1.2	BPSv002	Figure 40	Figure 42	Figures 43 - 49	Figure 50	Figure 41
MV4	Chronic Disease Managed and Patient Quality of Life Improved	DR03; DR05	GM1.1; GM1.1.1	BPSv007; BPSv009	Figure 51	Figure 52	Figures 53 - 56	Figure 57	No
MV5	Emergency Problem Alerted and Interventions in Emergency Situations Executed	DR06; DR07	GM1.2.1; GM1.2.3	BPSv011	Figure 58	Figure 59	Figures 60 - 65	Figure 66	No
MV6	Formal Care Improved	DR08	GM1.2	BPSv010	Figure 67	Figure 68	Figure 69	Figure 70	No

- **Contract model** details roles of services participating in a business process, such as providers and consumers of a service;
- **Protocols model** specifies constraints regarding how providers and consumers of a service can interact;
- **Services architecture model** describes participating services and contracts that must be accomplished to execute a business process, that in consequence, allows the accomplishment of specific missions. The service architecture diagram determines the internal configuration of business process services; and
- **Participants diagram** details participants involved in the execution of the business process. Participants, as well as their interactions, are identified based on the business process diagram.

Relations between MV, requirements, missions, and BPSv are shown in the first five columns of Table 11. Models illustrating the business process executed by each BPSv are linked in the last five columns.

To represent BPSv, business process models were created using BPMN (Business Process Management Notation). It is important to clarify that, models using BPMN represent fragments, since not all work, control, and message flows could be identified for BPSv, due to principally the uncertainty of some emergent behaviours.

Moreover, models using SoaML (Service Oriented Architecture Modelling Language) allowed the representation of capabilities, services interfaces, contracts, protocols, services architectures, and participants, required to realize BPSv. The remainder of this section gives more details about each mission view and its related models.

MV1 - Physical Status and Signs & Symptoms Monitored Remotely

This view aims to offer details of the composition of the business process service *BPSv001 - Patient status and sign & symptoms monitor*. This service aims to accomplish two high-level missions specified in *HomecARe*: *GM1.1.B - Remote physical examination performed*, and *GM1.1.1.A - Sign and symptoms monitored remotely*. Similarly, decisions presented in this view aid to address the domain requirement *DR01* demanded to *HomecARe* in Section 4.2. Diagrams used to represent this mission view are detailed as follows.

Business process diagram

The business process showed in Figure 17 depicts the work-flow needed to perform a remote physical examination of a patient at home, as well as, to monitor, in a remote way, her/his signs and symptoms. For executing this business process, four participants (services) must interact: the *Patient sign-symptom monitor*, the *Patient physical status analyser*, the *Inference engine*, and the *Patient DL records*. Details about each participating service are given as follows.

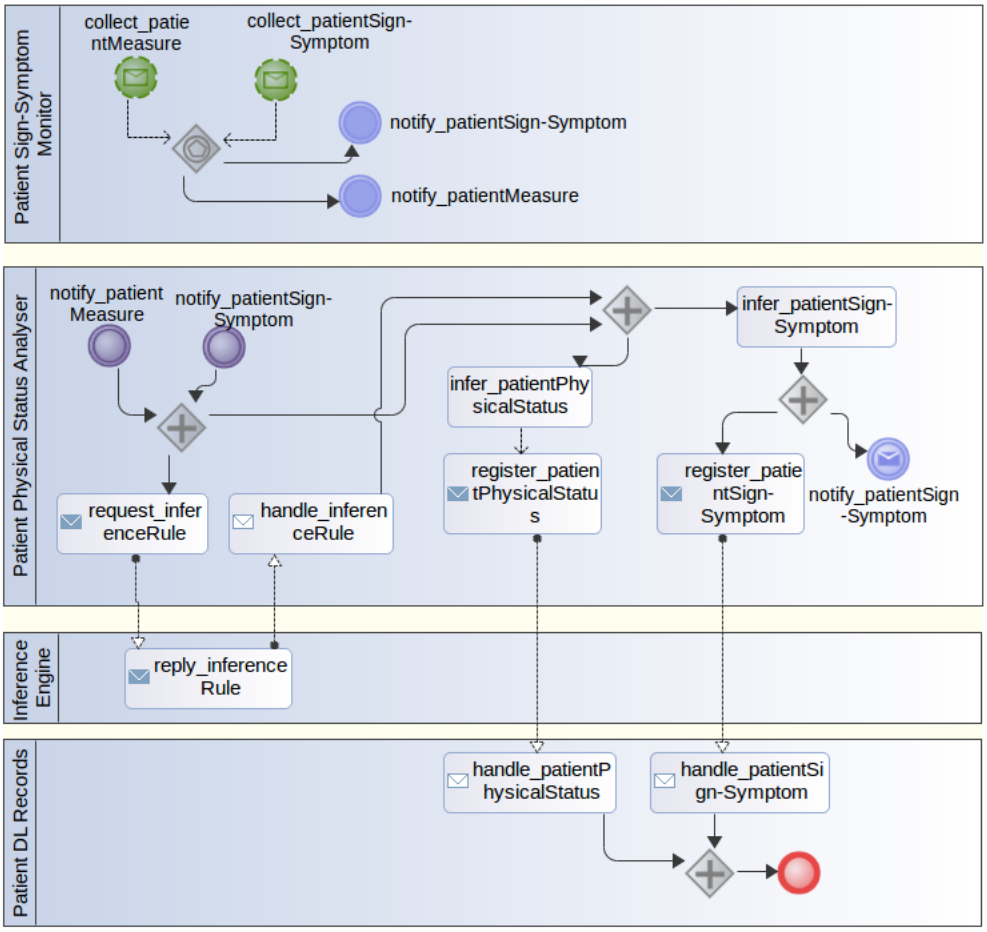


Figure 17 – Process diagram for the business process service *BPSv001 - Patient status and sign & symptoms monitor*

.

- **Patient sign-symptom monitor**, is an specialization of a *collector service* (a type of control service, CSv002, in Table 9) responsible for collect data notified or published by constituent systems services (CSSv). See a list of CSSv in Table 8. Collected data are measures of patient vital signs provided by the *Physiological function assessment service* (CSSv001), and patient sign and symptoms provided by others constituent systems services such as, for instance, the *CSSv004 - Nervous system assessment service* or the *CSSv002 - Cardiovascular system assessment service*. The *Patient sign-symptom monitor* publishes in the HSB the collected data in order to be processed by the *Patient physical status analyser*.

- **Patient physical status analyser**, is an specialization of an *analyser service* (a type of control service, CSv003, in Table 9) responsible for inferring, registering and notifying patient physical situation, and possible new sign and symptoms, based on data collected and published by the *Patient sign-symptom monitor*.

- **Inference engine** offers inference rules to the *Patient physical status analyser* for supporting the inference of patient physical situation and, possible new, sign and symptoms.

- **Patient DL records** maintains records about patient physical situation and sign and symptoms sent by the *Patient physical status analyser*.

Capabilities diagram

Capabilities were defined to allow the activities performed by participants in the business process in Figure 17. Figure 18 shows capabilities required to be offered by participating services for performing the business process in Figure 17. As such process is related to the business process service *BPSv001 - Patient status and sign & symptoms monitor*, capabilities in Figure 18 allow choreography of participating services to conform the BPSv001, and hence, to achieve its related high-level missions, i.e., GM1.1.1.A and GM1.1.B in Section 4.1.

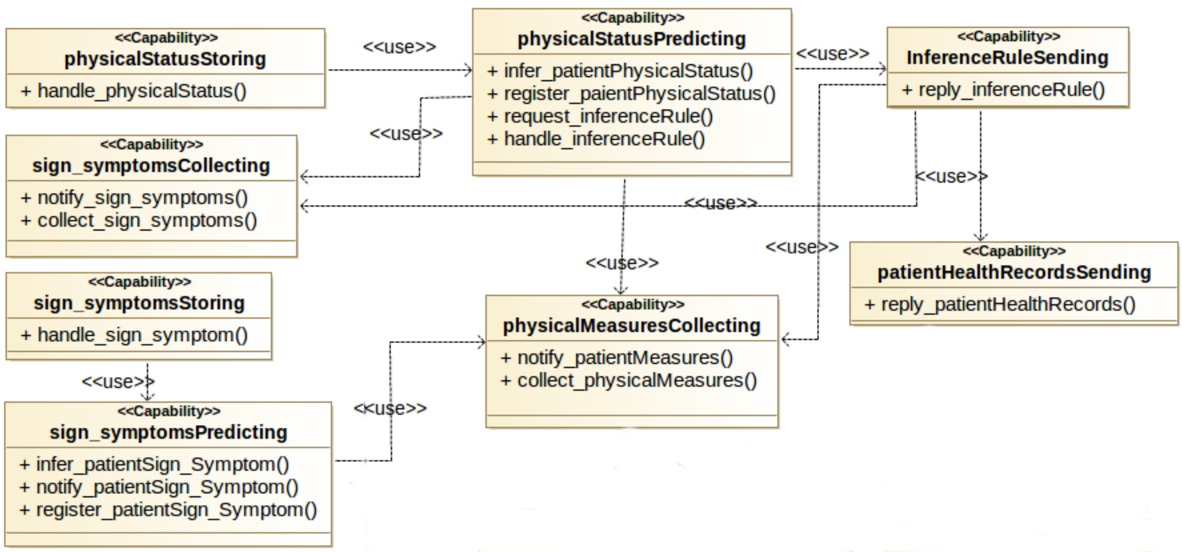
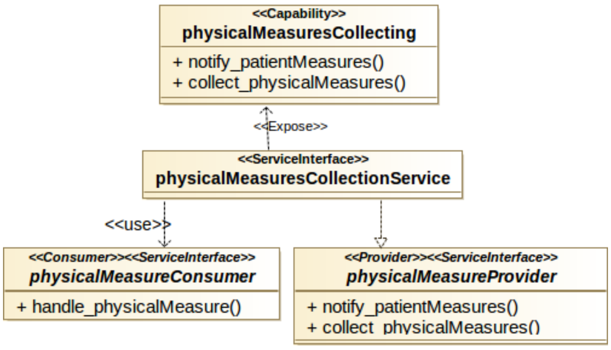


Figure 18 – Capability Diagram for the business process service *BPSv001 - Patient status and sign & symptoms monitor*

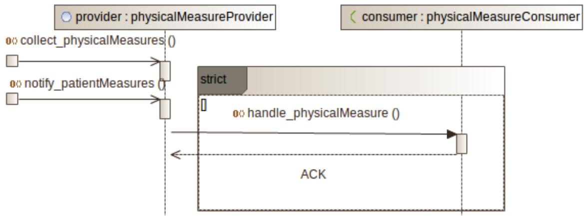
Services interfaces, contracts and protocols diagrams

For each capability in the capability diagram in Figure 18, a service interface, contract and protocol were established, aiming the choreography of participating services to conform the BPSv001 and executing the business process in Figure 17. Diagrams of service interface, contracts and protocols for each capability are presented as follows.

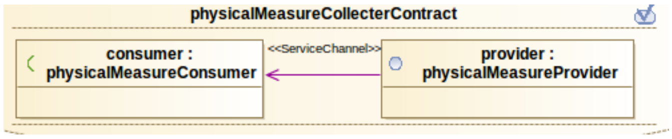
To realize the capability *physicalMeasuresCollecting*, the service interface *physicalMeasuresCollectionService* is proposed, as detailed in Figure 19a. This interface has associated a consumer named *physicalMeasureConsumer*, that in this case, can be the participating service *Patient physical Status analyser*. A provider named *physicalMeasureProvider*, can be a service of the type *CSSv001- Physiological function assessment*. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 19c, which is constrained by the interaction protocol in Figure 19b.



(a) Service interface.



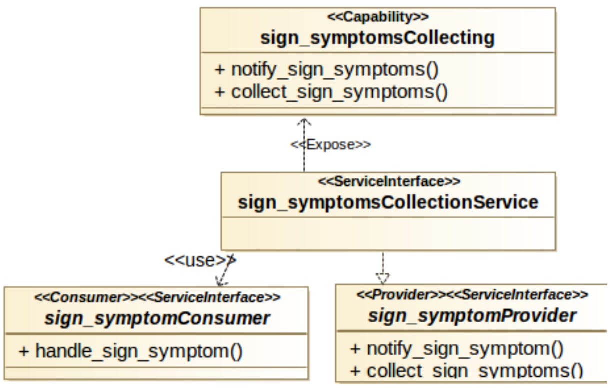
(b) Service protocol.



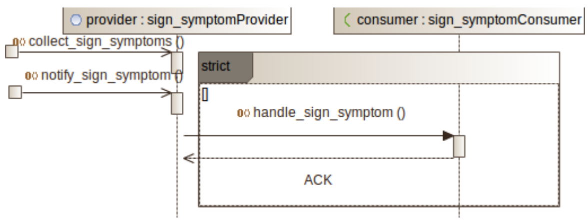
(c) Service contract.

Figure 19 – Service interface, contracts and protocol diagrams for the capability *physicalMeasuresCollecting*

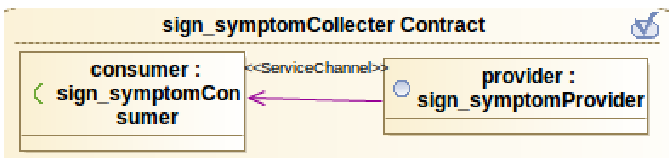
To realize the capability *sign_symptomsCollecting*, the service interface *sign_symptomsCollectionService* is proposed, as detailed in Figure 20a. This interface has associated a consumer named *sign_symptomConsumer*, that can be the participating service *Patient physical Status analyser*. A provider named *sign_symptomProvider*, can be any service of types *CSSv002-CSSv008* in Table 8 of Section 5.1. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 20c, which is constrained by the interaction protocol in Figure 20b.



(a) Service interface.



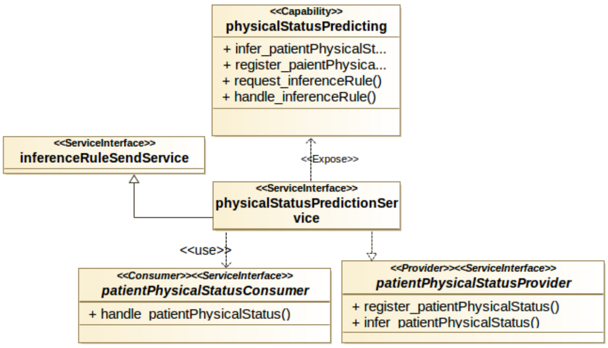
(b) Service protocol.



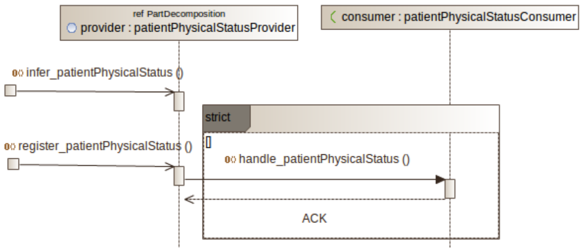
(c) Service contract.

Figure 20 – Service interface, contracts and protocol diagrams for the capability *sign_symptomsCollecting*

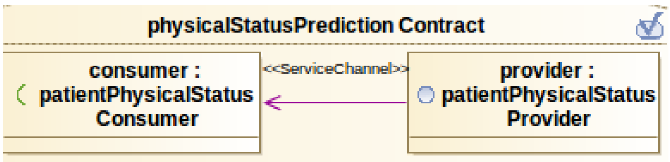
To realize the capability *physicalStatusPredicting*, the service interface *physicalStatus-PredictionService* is proposed, as detailed in Figure 21a. This interface has associated a consumer named *patientPhysicalStatusConsumer*, that can be the participating service *Patient DL Records*. A provider named *patientPhysicalStatusProvider*, is the participating service *Patient physical status analyser*. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 21c, which is constrained by the interaction protocol in Figure 21b.



(a) Service interface.



(b) Service protocol.



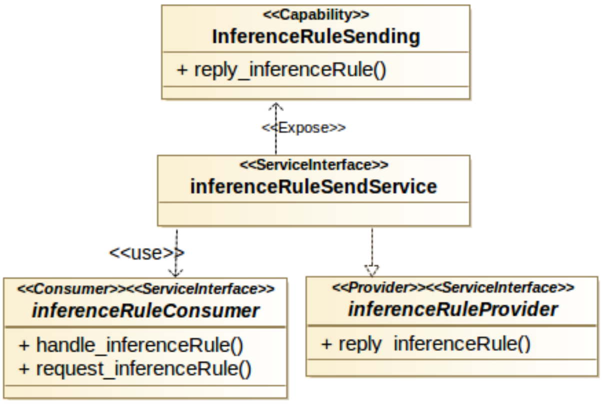
(c) Service contract.

Figure 21 – Service interface, contracts and protocol diagrams for the capability *physicalStatusPredicting*

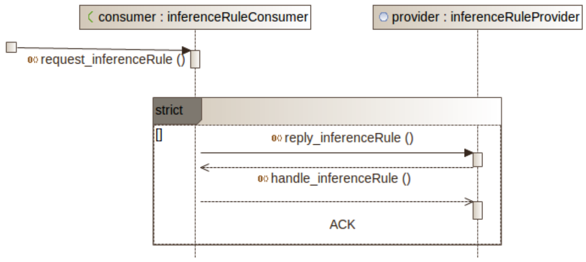
To realize the capability *inferenceRuleSending*, the service interface *inferenceRuleSend-Service* is proposed, as detailed in Figure 22a. This interface has associated a consumer named *inferenceRuleConsumer*, that can be the participating service *Patient physical status analyser*. A provider named *inferenceRuleProvider*, is the participating service *Inference engine*. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 22c, which is constrained by the interaction protocol in Figure 22b.

To realize the capability *sign_symptomsPredicting*, the service interface *sign_symp-tomsPredictionService* is proposed, as detailed in Figure 23a. This interface has associated a consumer named *sign_symptomsPredictionConsumer*, that can be an *Interceptor service* located in the HSB. A provider named *sign_symptomsPredictionProvider*, is the participating service *Patient physical status analyser*. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 23c, which is constrained by the interaction protocol in Figure 23b.

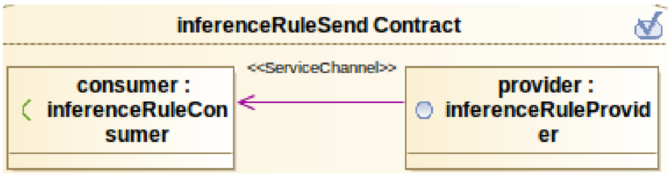
To realize the capability *sign_symptomsStoring*, the service interface *sign_symptoms-StoreService* is proposed, as detailed in Figure 24a. This interface has associated a consumer named *sign_symptomsConsumer*, that can be the participating service *Patient DL records*. A provider named *sign_symptomsProvider*, is the participating service *Patient physical status*



(a) Service interface.

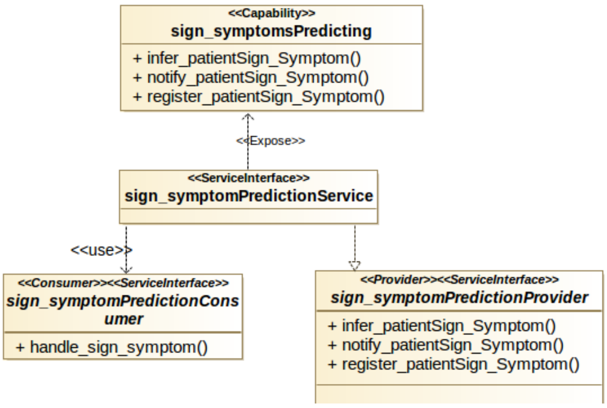


(b) Service protocol.

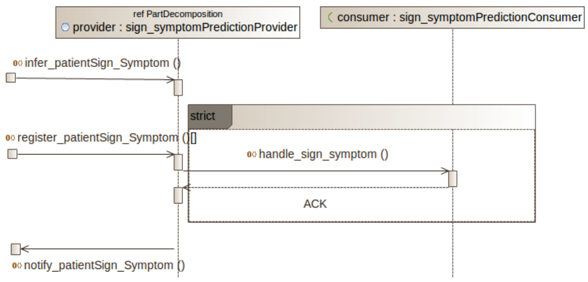


(c) Service contract.

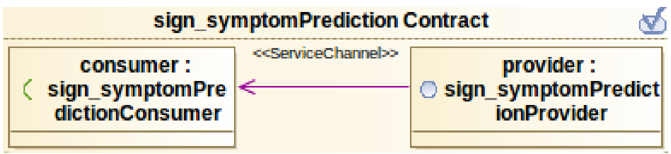
Figure 22 – Service interface, contracts and protocol diagrams for the capability *inferenceRuleSending*



(a) Service interface.



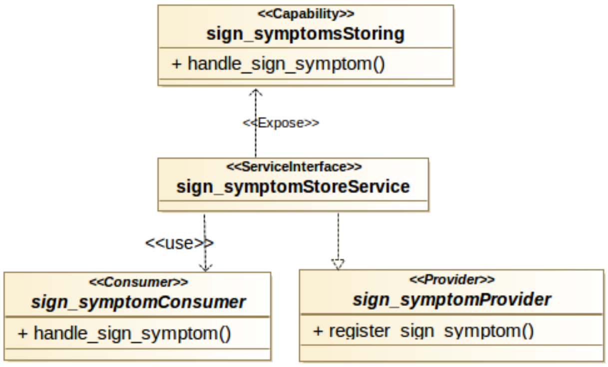
(b) Service protocol.



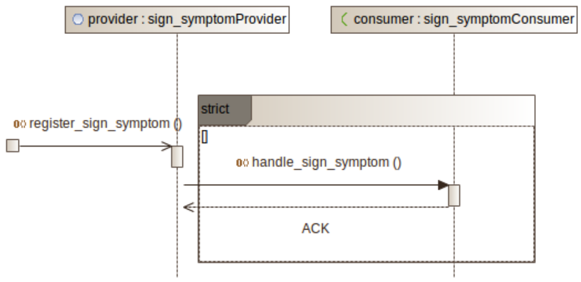
(c) Service contract.

Figure 23 – Service interface, contracts and protocol diagrams for the capability *sign_symptomsPredicting*

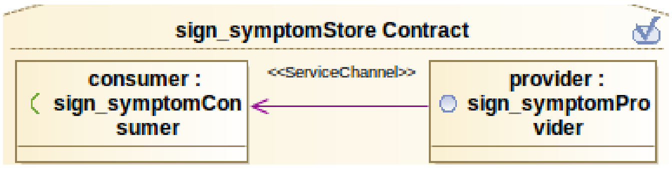
analyser. To be interoperable, the provider and consumer must accomplish the contract showed in Figure 24c, which is constrained by the interaction protocol in Figure 24b.



(a) Service interface.



(b) Service protocol.



(c) Service contract.

Figure 24 – Service interface, contracts and protocol diagrams for the capability *sign_symptomsStoring*

Services architecture diagram

Diagram presented in Figure 25 shows the services architecture defined in *HomecARe* for the business process service *BPSv001 - Patient status and sign & symptoms monitor*. Such architecture specifies how the four participating services must interact between them following specific contracts. Service interfaces and protocols associated to each contract were presented in Section 5.2. Moreover, the service architecture specifies roles for participating services, as defined in contract, in order to interact with other services, using their services interfaces. The interaction of participating services through contracts allow their choreography to conform the BPSv001, and hence, the accomplishment of the two missions of *HomecARe*: *GM1.1.B - Remote physical examination performed*, and *GM1.1.1.A - Sign and symptoms monitored remotely*.

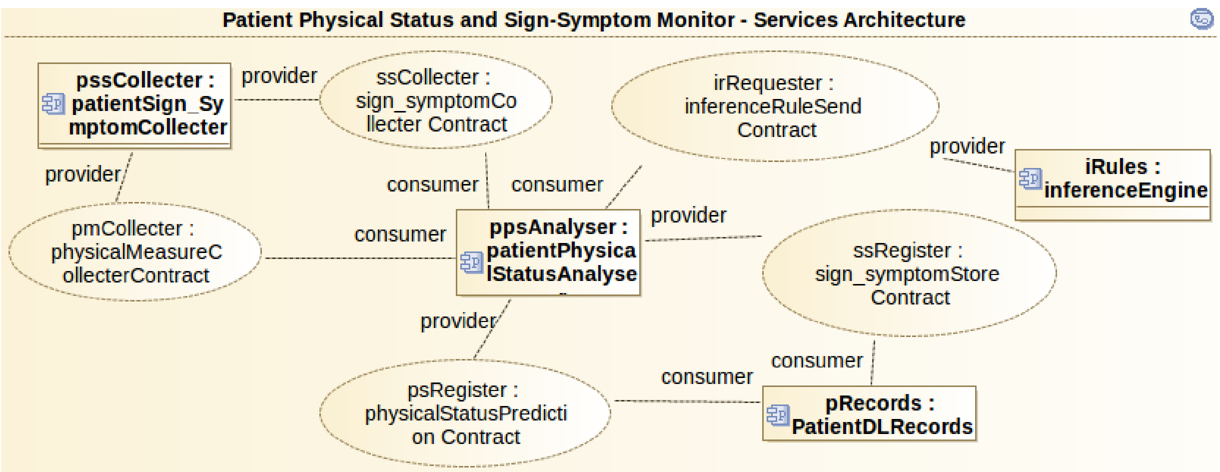


Figure 25 – Services architecture for the business process service *BPSv001 - Patient status and sign & symptoms monitor*.

MV2 - Patient Intervention Managed

This mission view offers knowledge about how *HomecARe* achieves the domain requirement *DR02* that demands the provision of the patient's intervention profile (See Table 7). For achieving such requirement, *HomecARe* must to accomplish the missions *(GM1.1.1.C) Intervention managed*; For this purpose, the business process service *(BPSv008) Disease intervention manager* was proposed. Specifically, this view describes the composition of BPSv008.

Relationships between missions, domain requirements of *HomecARe* and BPSv were introduced in Table 11. Models used to represent this mission view, i.e., capabilities, services interfaces, contracts, protocols and services architecture, are detailed as follows. Moreover, a participants model is also presented to offer a better representation of some generalizations made for the BPSv008.

Business process diagram

The diagram presented in Figure 26 details the work-flow required to establish a patient's intervention profile in *HomecARe*. This profile contains information about the situation of rehabilitation, pharmacological and non-pharmacological (e.g., diet, position) treatments, and of environment interventions specified in the patient care plan. Twelve participants are involved in the execution of the process presented in Figure 26, as follows.

- *Rehabilitation manager*, is responsible for inferring, recording and informing the situation of patient rehabilitation treatment. Such situation is used to analyse if the prescribed rehabilitation plan is being correctly followed, or if plan modifications are needed;

- *Pharmacological treatment manager*, is in charged of establishing the situation about patient pharmacological treatment, in order to define if prescriptions made in the medication plan are positively contributing to patient well-being;

- *Dietetic assistant*, monitors patient meals to infer if she/he is exactly following her/his nutrition plans. It records and communicates patient nutrition situation;

- *Patient positioning assistant*, advises when a patient position is inadequate for her/his condition and infers position situation to be communicated to interested entities;

- *Environment manager*, defines and communicates patient environment situation that is further used to establish an intervention plan, if needed;

- *Disease intervention manager*, establishes the patient intervention profile based on situations informed by rehabilitation manager, pharmacological manager, dietetic assistant, patient positioning assistant and environment manager. Such profile is recorded and communicated to interested parties;

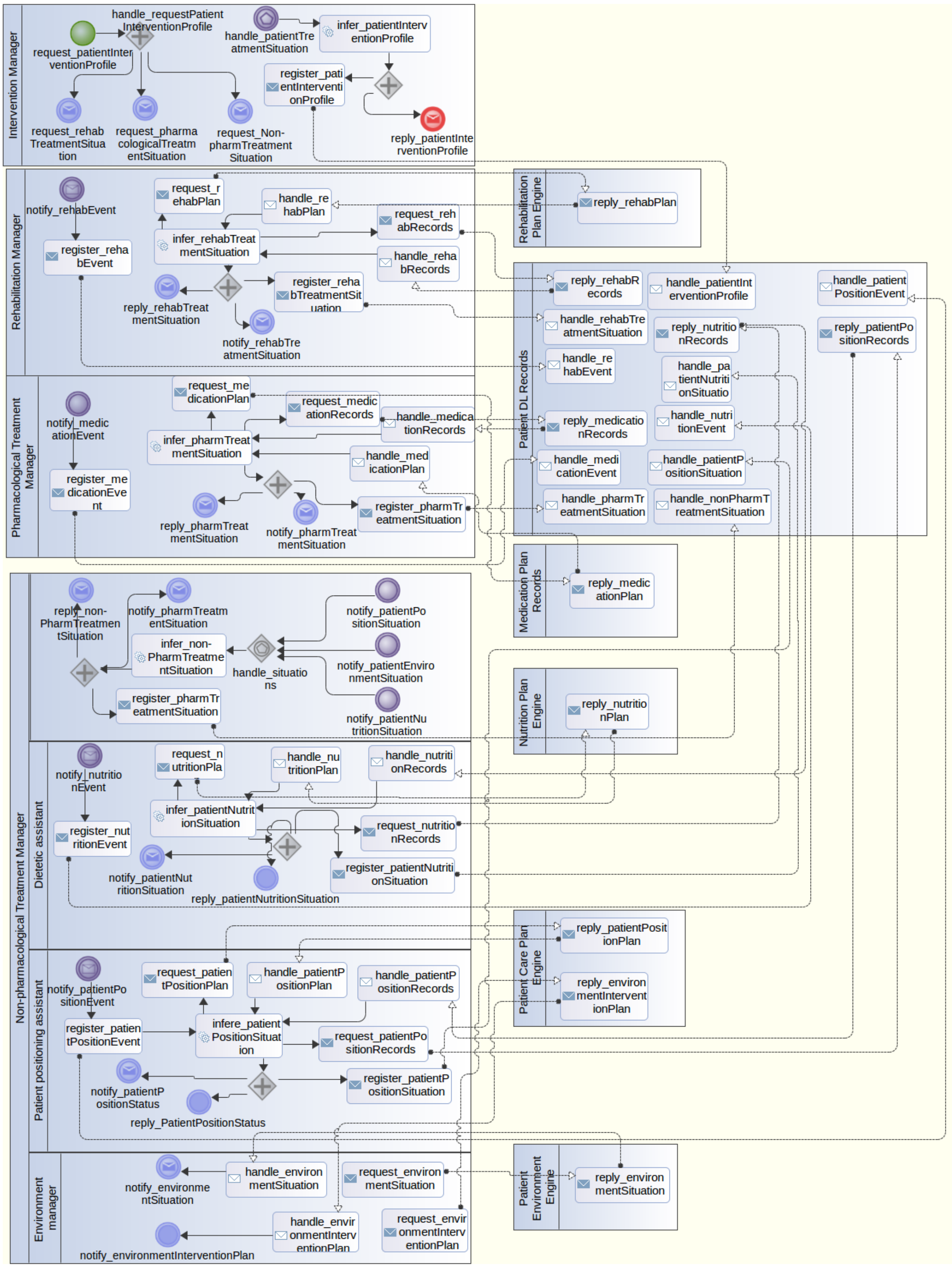


Figure 26 – Process diagram for the business process service *BPSv008 - Disease intervention manager*

.

- *Rehabilitation plan engine*, stores rehabilitation plans prescribed to the patient;
- *Patient DL records*, registers all information related to patient care;
- *Medication plan records*, maintains historical information about medication prescriptions of patient;

- *Nutrition plan engine*, records prescribed diet plans of patient;
- *Patient care plan engine*, stores additional care plans prescribed to patient by the care team; and
- *Patient environment engine*, registers patient environment information.

Participants diagram

Since some participating services perform similar activities to offer the information required to realize the business process in Figure 26, some generalizations of participants were made. As presented in Figure 27, the abstract participant *Intervention plan engine* generalizes the participants *Rehabilitation plan engine*, *Nutrition plan engine*, and *Medication plan records*. Moreover, the participants *Rehabilitation manager*, *Pharmacological treatment manager*, and *Non-pharmacological treatment manager* are generalized as the participant *Patient intervention assistant*. Similarly, the *Non-pharmacological treatment manager* is a generalization of the participants *Patient positioning assistant*, *Dietetic assistant*, and *Environment manager*.

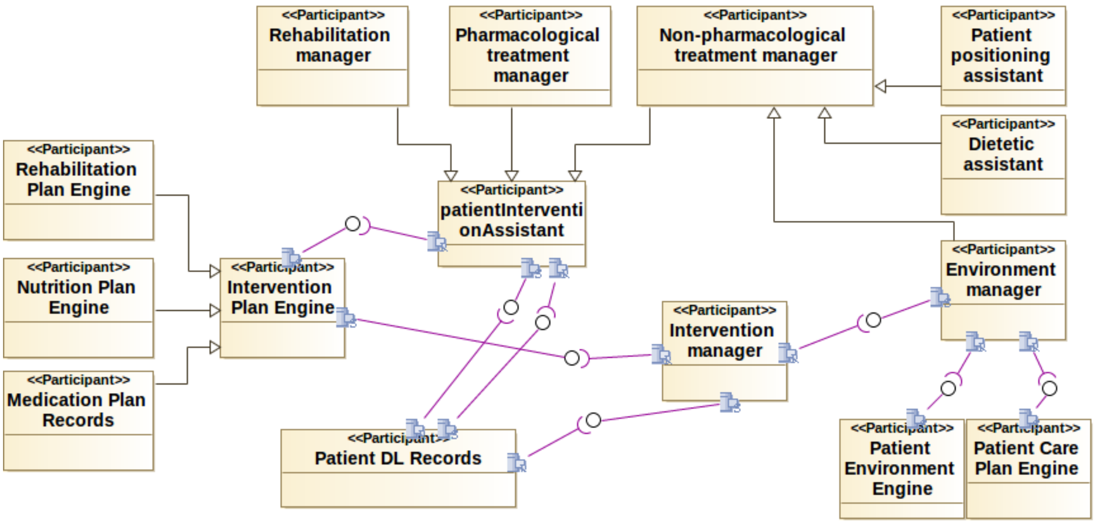


Figure 27 – Participants diagram related to BPSv008 - Disease Intervention Managed.

Capabilities diagram

Figure 28 shows all capabilities required to perform the business process in Figure 26, and consequently, to realize the BPSv008. Capabilities detail operations needed to execute activities described in the business process. Some capabilities in Figure 28 were proposed as an abstraction of similar activities performed by different participants, such as the capability *PatientInterventionSituationRequesting*, which is an abstraction of activities of request rehabilitation treatment situation or pharmacological treatment situations in Figure 26.

Relationships between capabilities are also described in the capability diagram. A participant can realize one or more capabilities, and each capability can be realized by one or more participants.

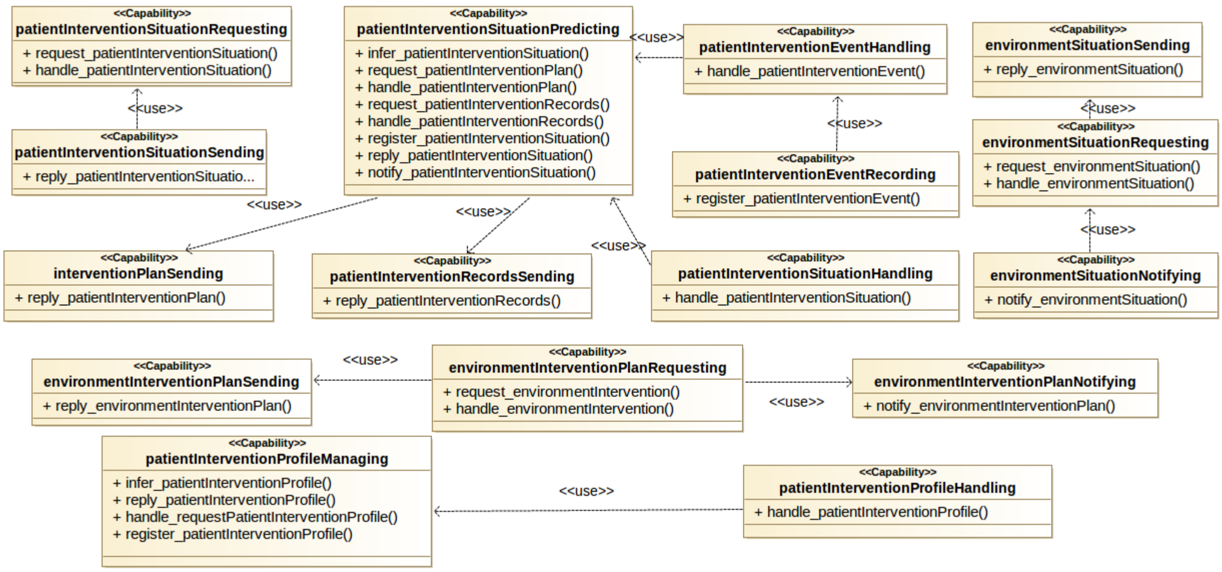


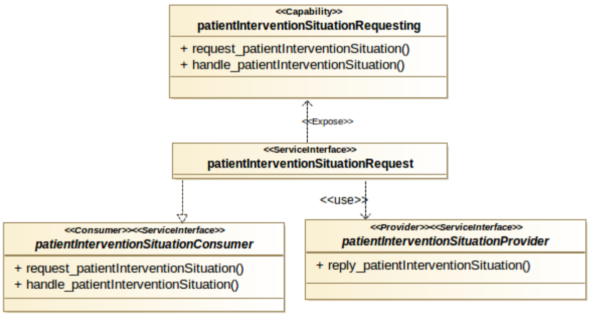
Figure 28 – Capability Diagram related to BPSv008 - Disease Intervention Managed

Services interfaces, contracts and protocols diagrams

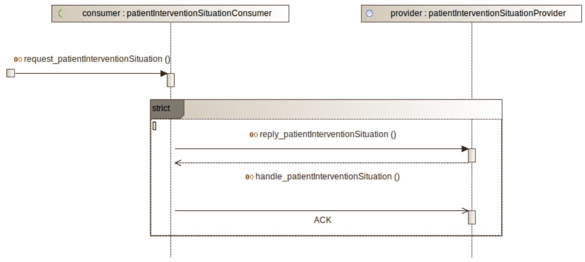
For each capability in the capability diagram in Figure 28, a service interface, contract and protocol were established, aiming the choreography of participants to conform the BPSv008 and executing the business process in Figure 26. Diagrams of service interface, contracts and protocols for each capability are presented as follows.

The capability *patientInterventionSituationRequesting* is accomplished through the service interface *patientInterventionSituationRequest* as showed in Figure 29a. The contract and protocol restraining interactions between consumer and provider using such interface are presented in Figures 29c and 29b, respectively. Consumers the interface *patientInterventionSituationRequest* can be the participating service *Disease intervention manager* (BSv008); whilst interface providers can be the *Rehabilitation manager* (BPSv003), *Pharmacological treatment manager* (BPSv004), *Dietetic assistant* (BPSv005), or *Patient position assistant* (BPSv006).

In Figure 30a, the service interface *patientInterventionProfileManagement* realizes the capability *patientInterventionProfileManaging*. The provider of a patient intervention profile is the service *Disease intervention manager*. Such profile can be consumed by any service related to the home healthcare team, e.g., *RN services* or *MD-OD services*. Home healthcare team services were presented in Section 5.1. Contracts and protocols associated to the service interface *patientInterventionProfileManagement* are presented in Figures 30c and 30b, respectively.



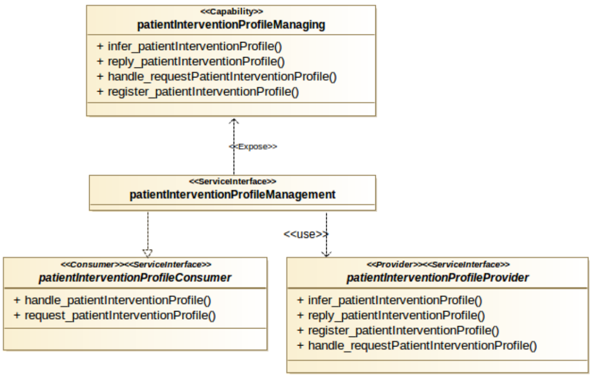
(a) Service interface.



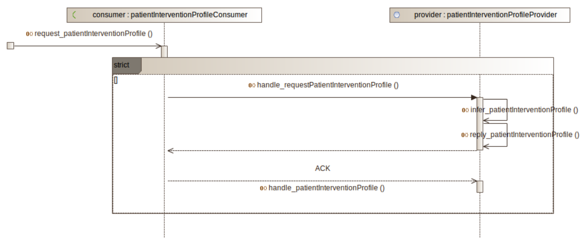
(b) Service protocol.

(c) Service contract.

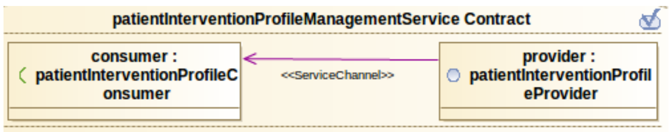
Figure 29 – Service interface, contracts and protocol diagrams for the capability *patientInterventionSituationRequesting*



(a) Service interface.



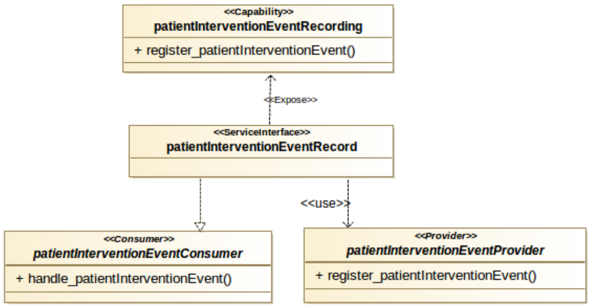
(b) Service protocol.



(c) Service contract.

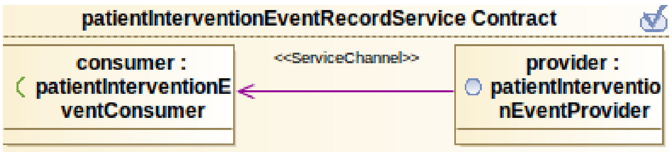
Figure 30 – Service interface, contracts and protocol diagrams for the capability *patientInterventionProfileManaging*

Figure 31a presents the service interface *patientInterventionEventRecord* associated to the capability *patientInterventionEventRecording*. The consumer of this capability in the BPSv008 is the *Patient DL Records* which needs to register all events related to any intervention, e.g., pharmacological events notifying medication consumption by patient. In this context, providers of intervention events can be the *rehabilitation manager*, *dietetic manager*, *pharmacological manager* or *environmental manager*. Contracts and protocols associated to the service interface *patientInterventionEventRecording* are presented in Figures 31c and 31b, respectively.



(a) Service interface.

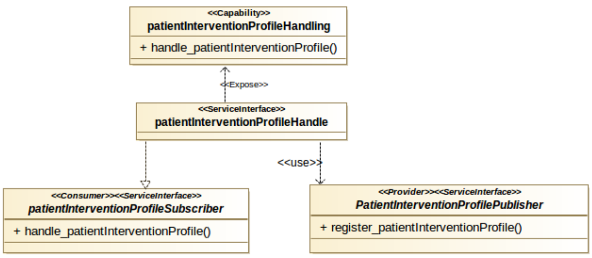
(b) Service protocol.



(c) Service contract.

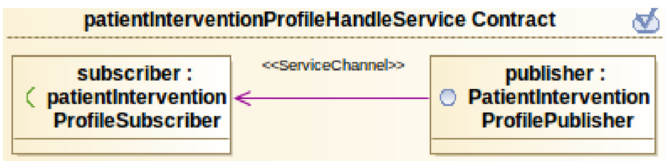
Figure 31 – Service interface, contracts and protocol diagrams for the capability *patientInterventionEventRecording*

Figure 32a exposes the service interface *patientInterventionProfileHandle* related to the capability *patientInterventionProfileHandling*. The patient intervention profile is provided by *BPSv008 - Disease intervention manager* and consumers of such profile can be: (i) the *Patient DL Records* for updating patient health records, or (ii) the *interceptor* in the HSB for routing the profile to interested parties. Contracts and protocols associated to the service interface *patientInterventionProfileHandle* are presented in Figures 32c and 32b, respectively.



(a) Service interface.

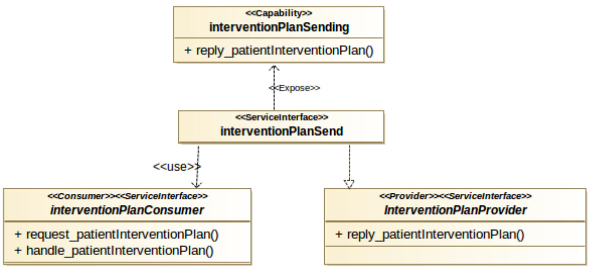
(b) Service protocol.



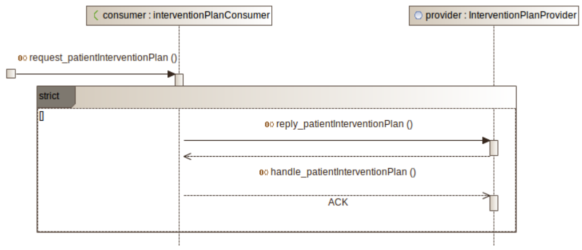
(c) Service contract.

Figure 32 – Service interface, contracts and protocol diagrams for the capability *patientInterventionProfileHandling*

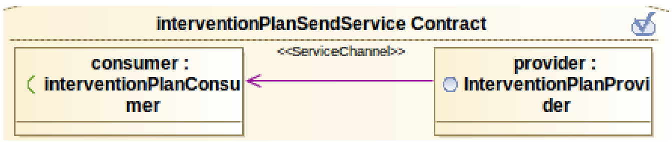
Figure 33a details the service interface *InterventionPlanSend* for the capability *InterventionPlanSending*. This interface can be used by participants to request and receive (consume) an intervention plan. An intervention plan contains activities and schedule to perform such activities in order to follow an intervention by both patient and healthcare team. Examples of intervention plans are rehabilitation, medication, nutrition, patient positioning or environmental interventions plans. Providers of intervention plans can be the *Patient DL Records*, or any plans repository or engine, such as *Nutrition Plan Engine* or *Medication plan records*. Contracts and protocols associated to the service interface *InterventionPlanSend* are presented in Figures 33c and 33b, respectively.



(a) Service interface.



(b) Service protocol.



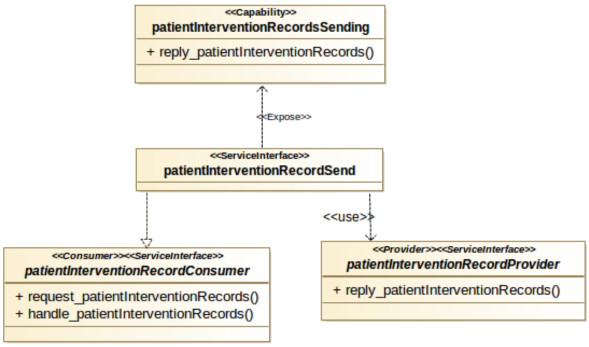
(c) Service contract.

Figure 33 – Service interface, contracts and protocol diagrams for the capability *InterventionPlanSending*

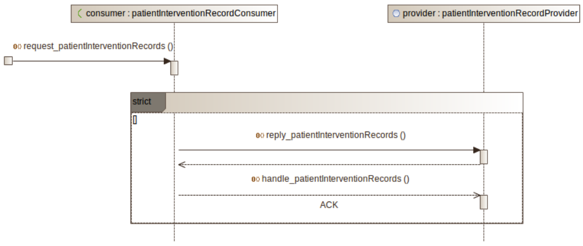
Figure 34a illustrates the service interface *patientInterventionRecordSend* for the capability *patientInterventionRecordSending*. Such interface can be used to request historical records of activities performed by patient and home team care regarding her/his prescribed interventions. An example of an intervention record consumer is the *Pharmacological treatment manager* that can request medication reports of patient over a determined time period. Hence, a provider can be the *Patient DL Records*. Contracts and protocols associated to the service interface *patientInterventionRecordSend* are presented in Figures 34c and 34b, respectively.

Figure 35a shows the service interface *patientInterventionSituationHandle* that is associated to capabilities *patientInterventionSituationSending* and *patientInterventionSituationHandling*. This interface can be used to provide a situation of a patient intervention. For instance, in Figure 26, the *Disease intervention manager* request the situations of patient rehabilitation and pharmacological treatments. Hence, the *rehabilitation manager* or *Pharmacological treatment manager* can use this interface to communicate the respective intervention situations. Contracts and protocols associated to the service interface *patientInterventionSituationHandle* are presented in Figures 35c and 35b, respectively.

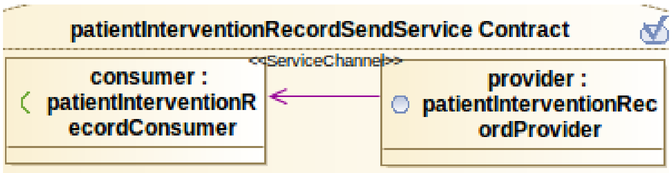
The service interface *environmentSituationSend*, presented in Figure 36a, allows an *environmentSituationConsumer* to request and handle situations of patient environment supplied



(a) Service interface.

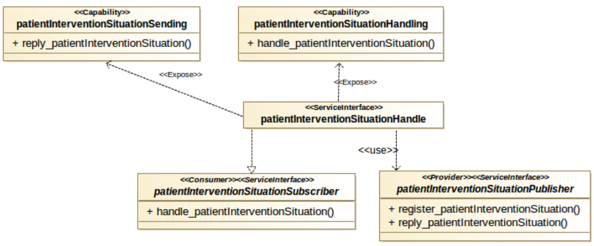


(b) Service protocol.

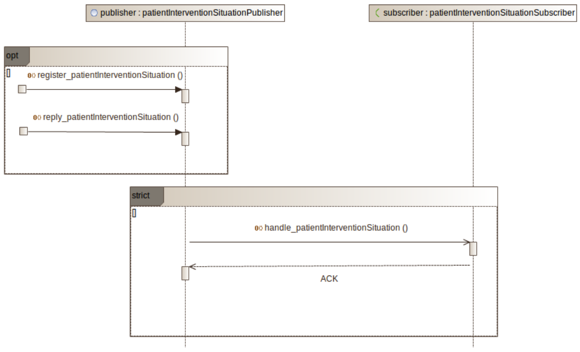


(c) Service contract.

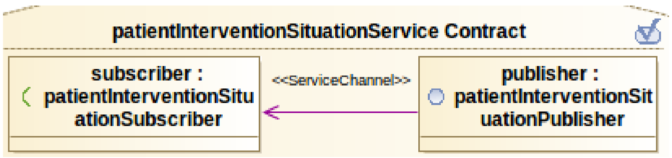
Figure 34 – Service interface, contracts and protocol diagrams for the capability *patientInterventionSituationSending*



(a) Service interface.



(b) Service protocol.

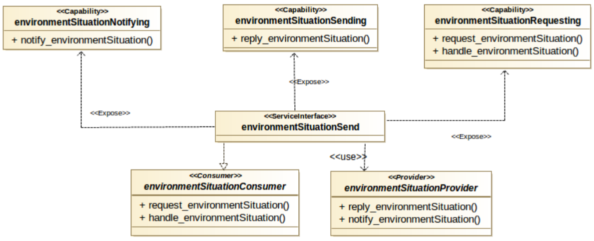


(c) Service contract.

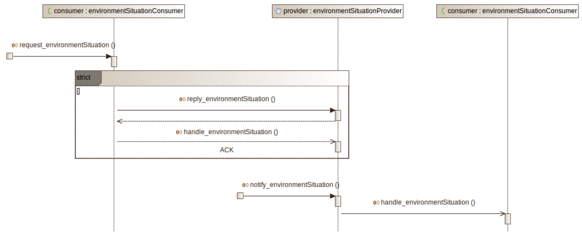
Figure 35 – Service interface, contracts and protocol diagrams for the capability *patientInterventionSituationSending*

by an *environmentSituationProvider*. A consumer of this interface can be the *Environment manager*, and the *Patient environment engine* can act as a provider. Contracts and protocols associated to the service interface *environmentSituationSend* are presented in Figures 36c and 36b, respectively.

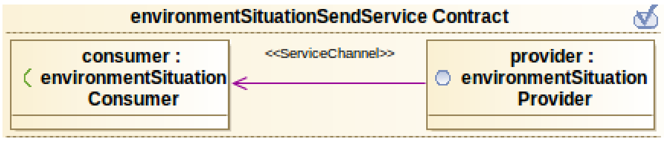
In Figure 37, is presented the service interface *environmentInterventionPlanSend*. This interface is related to capabilities *environmentInterventionPlanSending*, *environmentIntervention-*



(a) Service interface.



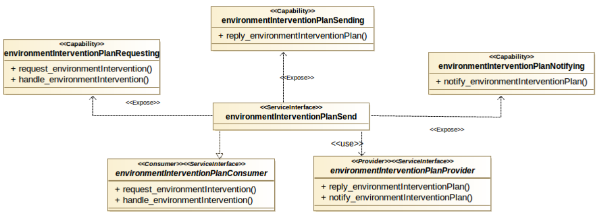
(b) Service protocol.



(c) Service contract.

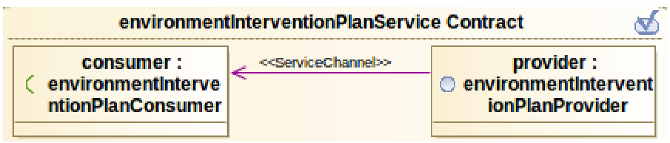
Figure 36 – Service interface, contracts and protocol diagrams for the capability *environmentSituation-Sending*

PlanRequesting and *environmentInterventionPlanNotifying*. An *environmentInterventionPlan-Consumer* requests and handles intervention plans for being executed in patient environment, that are supplied by an *environmentInterventionPlanProvider*. A consumer of such interface can be the *Environment manager* and the provider can be the *Patient environment engine*. Contracts and protocols associated to this service interface are presented, respectively, in Figures 37c and 37b.



(a) Service interface.

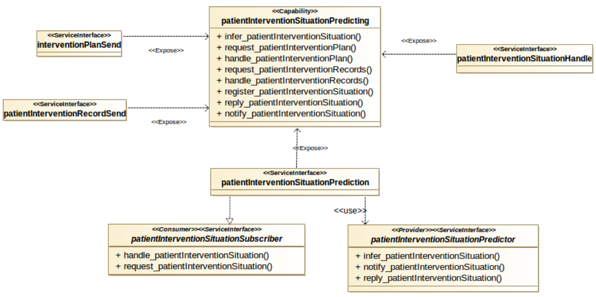
(b) Service protocol.



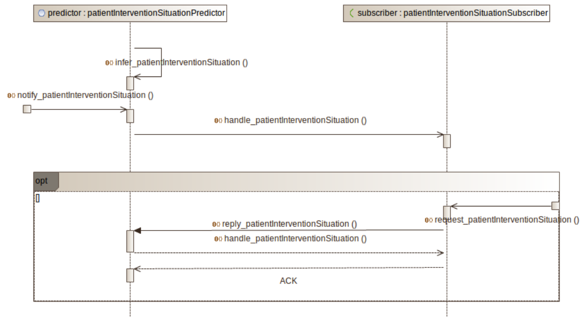
(c) Service contract.

Figure 37 – Service interface, contracts and protocol diagrams for the capability *environmentIntervention-PlanRequesting*

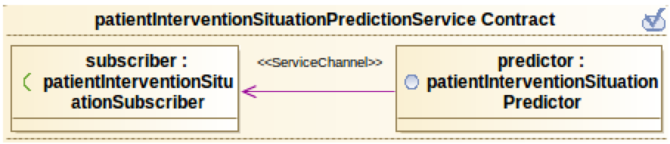
The service interface *patientInterventionSituationPrediction*, showed in Figure 38a, relates the capability *patientInterventionSituationPredicting*. Similarly, for achieving such capability, three additional services interfaces are needed, the *interventionPlanSend* in Figure 33a, the *patientInterventionRecordSend* in Figure 34a, and the *patientInterventionSituationHandle* in Figure 35a. A consumer, the *patientInterventionSituationSubscriber*, is subscribed to receive informations about the situation of patient interventions, e.g., pharmacological or rehabilitation treatments. For this interface, a consumer can be the *Disease intervention manager*. A provider, the *patientInterventionSituationPredictor*, infers and communicates the intervention situation. Providers with this interface can be the *Rehabilitation manager*, the *Pharmacological treatment manager* or the *Dietetic assistant*, among others participating services. Contract and protocol for this service interface are presented in Figures 38c and 38b, respectively.



(a) Service interface.



(b) Service protocol.



(c) Service contract.

Figure 38 – Service interface, contracts and protocol diagrams for the capability *patientInterventionSituationPredicting*

Services architecture diagram

The services architecture of the *BPSv008 - Disease intervention manager* is illustrated in Figure 39. The architecture shows all participating services needed to execute the business process showed in Figure 26, and hence, to achieve the high-level mission defined in *HomecARe*: *GM1.1.1.C - Intervention managed*. Participants are coordinated through specific services interfaces and constrained by contracts defined for each interface. Services interfaces, contracts and protocols relating with this services architecture were introduced in Section 5.2. Aiming the reuse of service interfaces, contracts and protocols, some participants were abstracted. For instance, the participating service *piAssistant:patientInterventionAssistant*, in the services architecture (Figure 39), is a generalization of the *Rehabilitation manager* and *Pharmacological treatment manager*, as showed in Figure 27.

MV3 - Activities of Daily Life Supported

This mission view provides knowledge about how *HomecARe* addresses the domain requirement *DR04*, and hence, the high-level mission *GM1.1.2 - Activities of daily life supported*. For achieving such requirement and mission, the business process service *BPSv002 - Activities of daily life manager* was proposed. In short, the BPSv002 aims to establish the profile of patient's

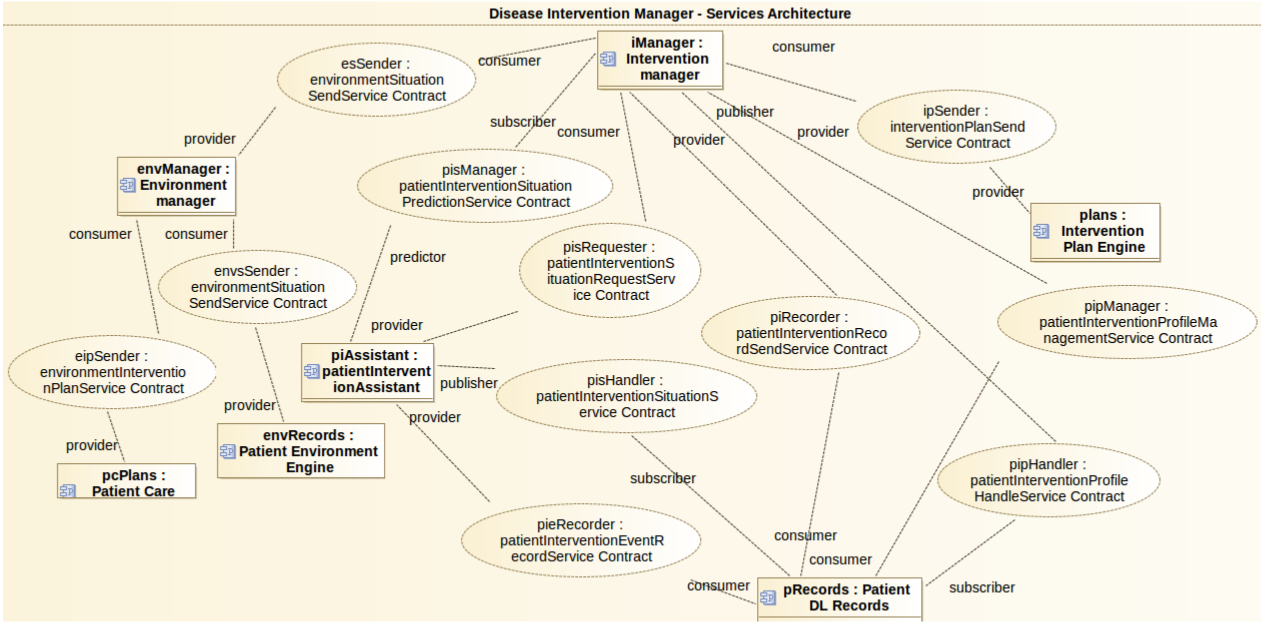


Figure 39 – Services architecture for the business process service *BPSv008 - Disease intervention manager*.

activities of daily life (AoDL) profile, containing information about patient's social, personal care, domestic and leisure activities. Moreover, such profile also considers the patient's environment status. Relationships between this mission view, with significant domain requirements, high-level missions of *HomecARe* and business process services were presented in Table 11. Models in SoaML used to represent this mission view, i.e., capabilities, services interfaces, contracts, protocols and services architecture, are detailed as follows. Additionally, a participants model is presented, detailing participating services in the realization of BPSv002. Such additional model was needed to better understanding some generalization made in this mission view.

Business process diagram

The business process presented in Figure 40 details the work-flow that must be executed by the BPSv002. Such process defines activities and interactions between participating services that are required to establish the patient's AoDL profile in *HomecARe*. In this context, to realize BPSv002 it is needed the collaboration of the following eight participants:

- *Activities of daily life monitor*, is responsible for establishing, storing and notifying the patient's AoDL profile based on activities situations notified by the social, personal care, domestic, leisure and environment managers and monitors.

- *Social activities monitor*, is responsible for monitoring social activities, such as, writing, calling, internet networking, and to communicate patient situation regarding those activities.

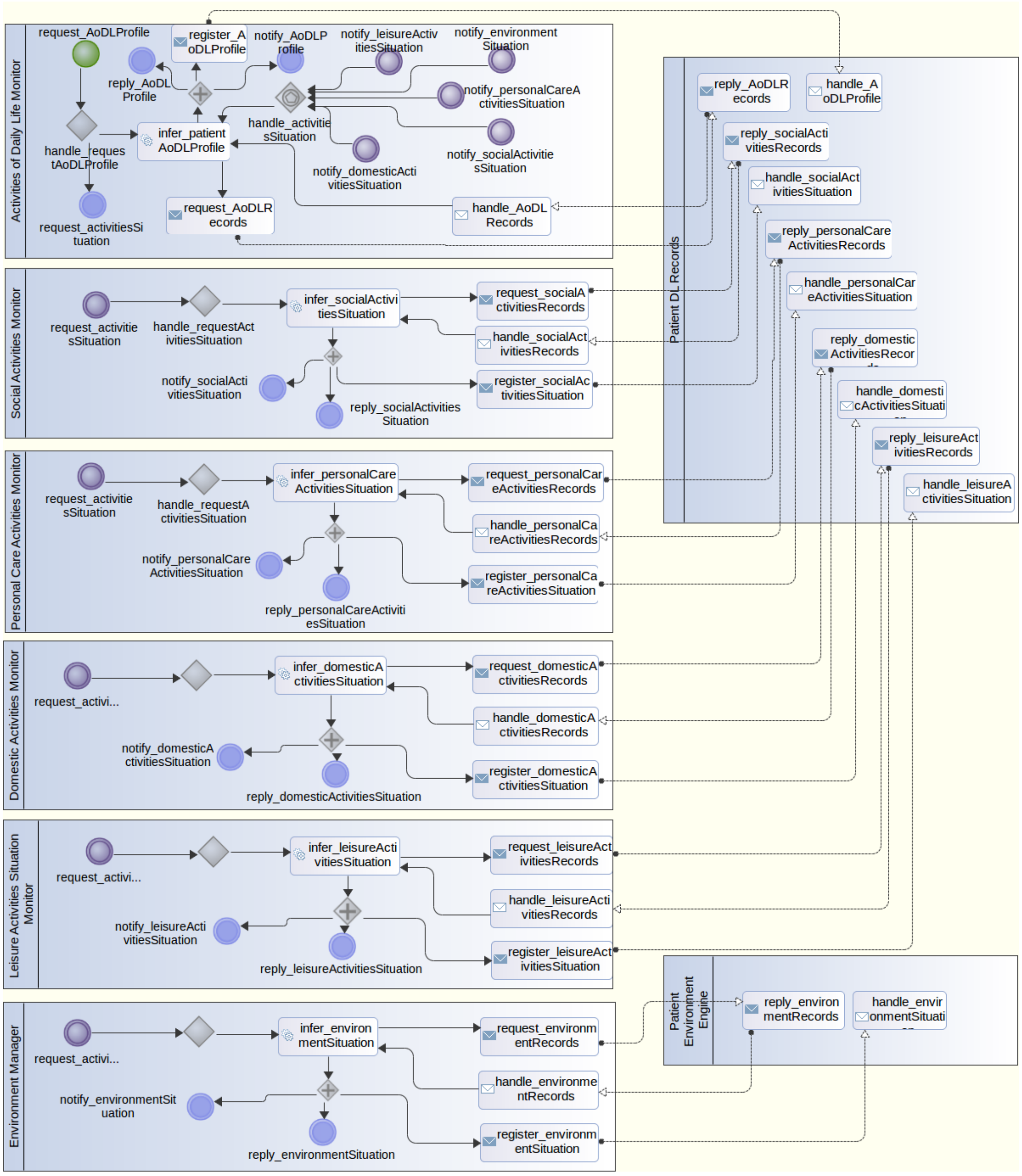


Figure 40 – Process diagram for the business process service *BPSv002 - Activities of daily life manager* .

- *Personal care activities monitor*, is in charge of monitoring personal care activities, such as, showering, eating, sleeping, and to communicate patient situation regarding those activities.

- *Domestic activities monitor*, is responsible for monitoring domestic activities, such as, cooking or cleaning, and to communicate patient situation regarding those activities.

- *Leisure activities monitor*, observes leisure activities, such as, reading, watching tv, gaming or playing an instrument, and establishes patient situation regarding those activities.

- *Environment manager*, is responsible for determining and communicating patient environment situation.

- *Patient DL records*, stores and provides patient activities records and situations, and
- *Patient environment engine*, stores and provides patient environment records and situations.

Participants diagram

Monitors of social, leisure, personal care and domestic activities were generalized as the participant *Patient activities monitor* as showed in Figure 41, since they share a similar internal activities flow to provide specific patient's activities situations.

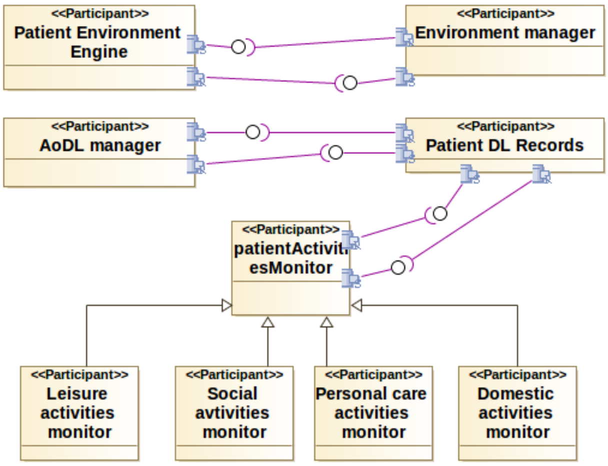


Figure 41 – Participants diagram related to the services architecture of BPSv002.

Capabilities diagram

Capabilities required to perform the business process in Figure 40, and consequently, to realize the BPSv002, are presented in Figure 42. Such capabilities must be offered by the aforementioned participating services. Capabilities were identified based on the activities performed by each participant in the business process in Figure 40.

Services interfaces, contracts and protocols diagrams

For each capability in the capability diagram in Figure 42, a service interface, contract and protocol were established, aiming the choreography of participants to conform the BPSv002, and hence, for executing the business process in Figure 40. Diagrams of service interface, contracts and protocols for each capability are presented as follows.

In Figure 43a is presented the service interface *activitiesRecordSend* related to the capability *activitiesRecordSending*. This interface allows transference of patient activities records, corresponding to social, personal care, domestic or leisure activities. An *activitiesRecordsConsumer* interface can be implemented by, for instance, the *Social activities monitor* or the *Domestic*

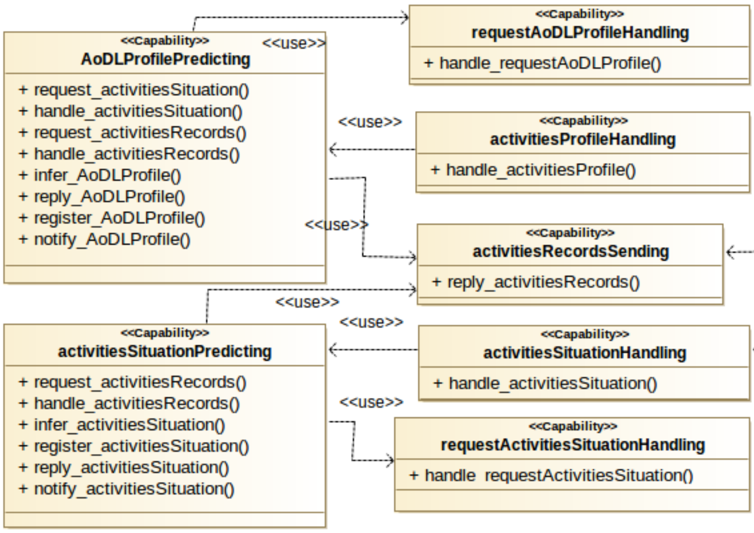
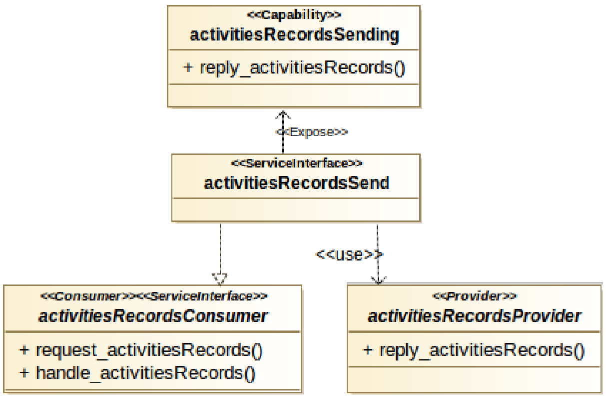
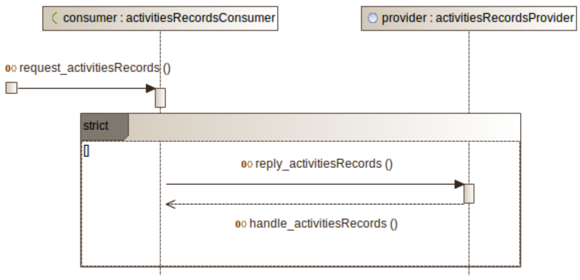


Figure 42 – Capabilities diagram for the business process service *BPSv002 - Activities of daily life manager*

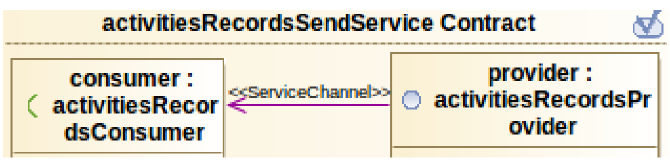
activities monitor. The *activitiesRecordsProvider* interface can be implemented by the participant *Patient DL records*. Contracts and protocols related with the service interface *activitiesRecord-Send* are presented in Figures 43c and 43b, respectively.



(a) Service interface.



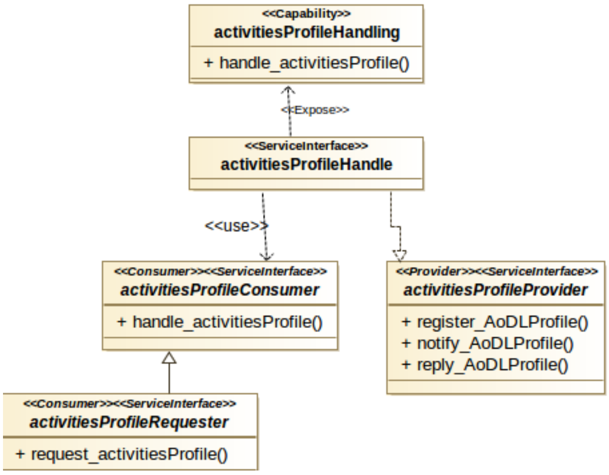
(b) Service protocol.



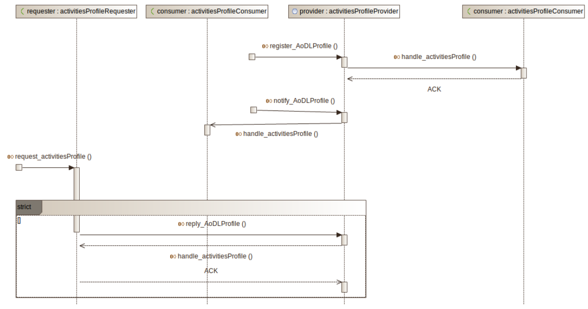
(c) Service contract.

Figure 43 – Service interface, contracts and protocol diagrams for the capability *activitiesRecordSending*

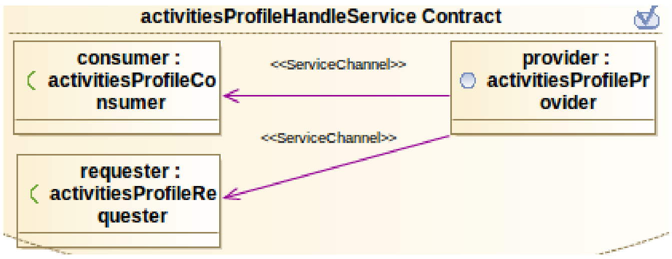
The service interface *activitiesProfileHandle* realizes the capability *activitiesProfileHandling*. The provider interface *activitiesProfileProvider* can be implemented by the *Activities of daily life manager* to communicate the patient profile of AoDL. The interfaces *activitiesProfileConsumer* or *activitiesProfileRequester* are used by consumers and requesters, respectively, to receive such profile. An example of consumer/requester can be the service *Quality of life monitor*, as presented in Section 5.2. Figures 44c and 44b depict the contract and protocol associated to the service interface *activitiesProfileHandle*.



(a) Service interface.



(b) Service protocol.



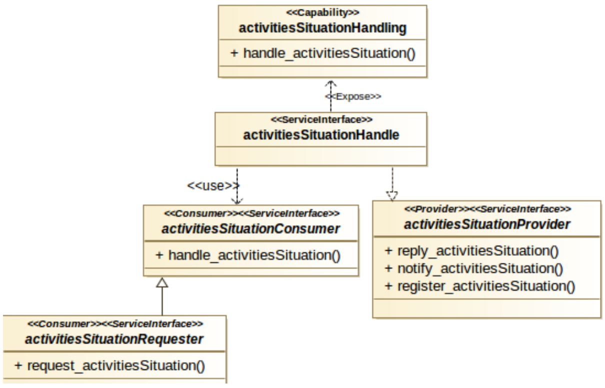
(c) Service contract.

Figure 44 – Service interface, contracts and protocol diagrams for the capability *activitiesProfileHandling*

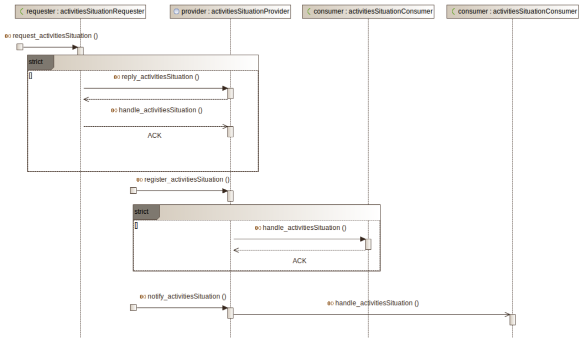
For allowing the capability *activitiesSituationHandling*, the service interface *activitiesSituationHandle* was proposed. Providers of activities situations, such as, *Personal care activities monitor* or *Domestic activities monitor*, use the interface *activitiesSituationProvider* to communicate a situation of a respective patient activity. The *Activities of daily life manager* uses the interfaces *activitiesSituationConsumer* or *activitiesSituationRequester* to receive patient activities situation. Contract and protocol related to the service interface *activitiesSituationHandle* are presented, respectively, in Figures 45c and 45b.

Figure 46a illustrates the service interface *environmentRecordSend* that allows the capability *environmentRecordSending*. The *Patient environment engine* offers patient environment records through the interface *environmentRecordsProvider*. The *Environment manager* receives such records through the interface *environmentRecordsConsumer*. Contract and protocol associated to the service interface *environmentRecordSend* are presented, respectively, in Figures 46c and 46b.

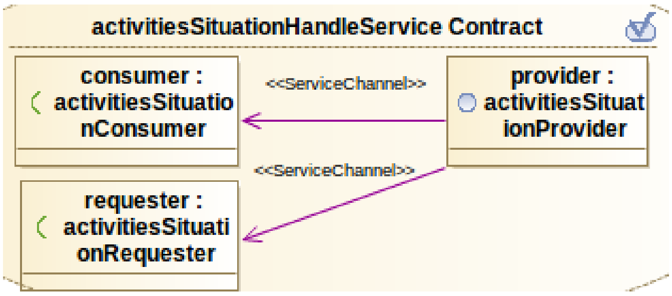
Figure 47a illustrates the service interface *requestHandle* that allows the capabilities *requestAoDLProfileHandling* and *requestActivitiesSituationHandling*. At implementing this



(a) Service interface.

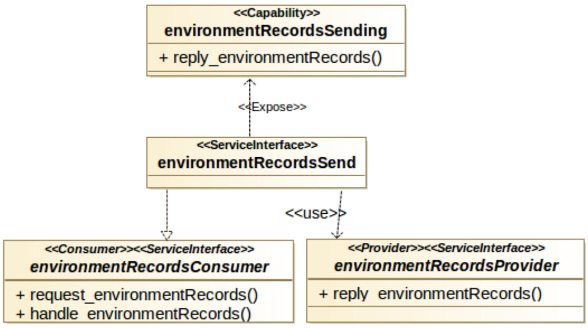


(b) Service protocol.

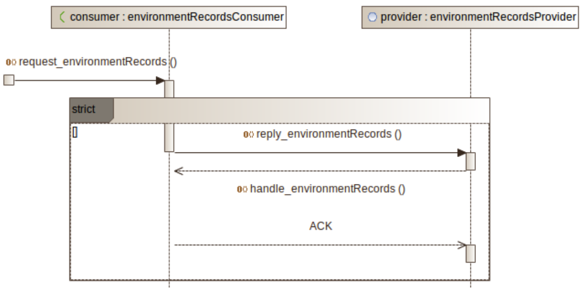


(c) Service contract.

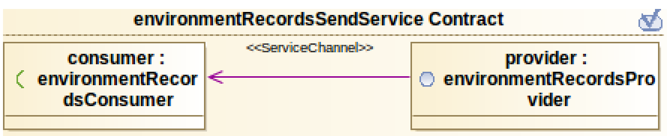
Figure 45 – Service interface, contracts and protocol diagrams for the capability *activitiesSituationHandling*



(a) Service interface.



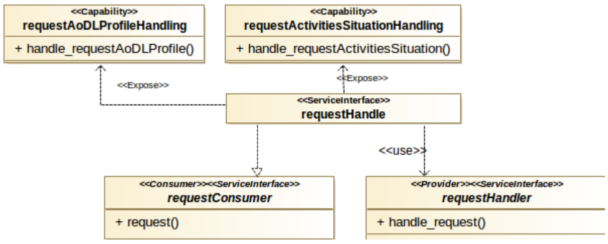
(b) Service protocol.



(c) Service contract.

Figure 46 – Service interface, contracts and protocol diagrams for the capability *environmentRecordSending*

service interface, a requester *requestConsumer* demands information, whether patient AoDL profile or some activities situations, to a provider, that must handle such request using the interface *requestHandler*. Contract and protocol of the service interface *requestHandle* are showed in Figures 47c and 47b, respectively.



(a) Service interface.

(b) Service protocol.

(c) Service contract.

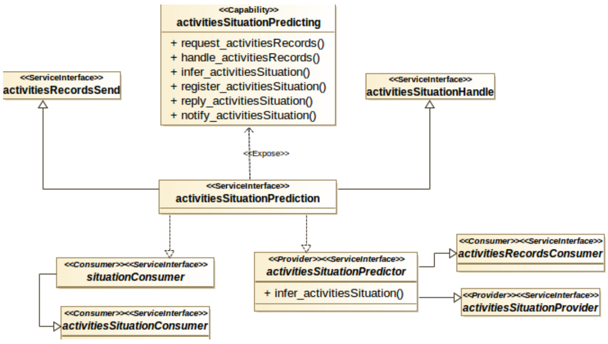
Figure 47 – Service interface, contracts and protocol diagrams for the capability *requestAoDLProfileHandling*

The capability *activitySituationPredicting* is allowed through the service interface *activitiesSituationPrediciton*, showed in Figure 48a, that inherits operations from two service interfaces: the *activitiesRecordsSend* and *activitiesSituationHandle* presented in Figures 43 and 45, respectively. A consumer, such as, the *Activity daily life manager* implements the interface *situationConsumer* to request and receive situations of patient activities, which inherits operations from the interface *activitiesSituationConsumer* introduced in Figure 45a. Providers of patient activities situations must implement the interface *activitiesSituationPredictor*, which inherits operations from two interfaces: the *activitiesRecordConsumer* and the *activitiesSituationProvider* detailed in Figures 43a and 45a, respectively. Contract and protocol of the service interface *activitiesSituationPrediciton* detailed in Figure 48a are illustrated in Figures 48c and 48b.

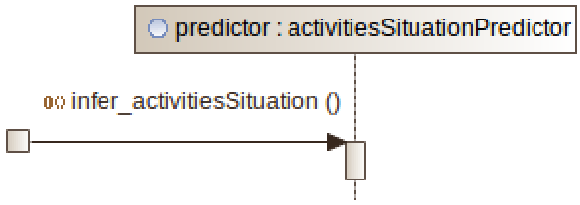
Figure 49a shows the service interface *activitiesProfilePrediction* that allows the capability *AoDLProfilePredicting*. This service interface inherits operations from other three service interfaces: the *activitiesSituationHandle*, the *activitiesRecordsSend* and the *activitiesProfileHandle*, which were introduced in Figures 45a, 43a and 44a, respectively. Hence, consumer *profileConsumer* and provider *activitiesProfilePredictor* of the service interface *activitiesProfilePrediction*, also inherits operation of consumer and provider interfaces in Figures 45a, 43a and 44a. Contract and protocol of the service interface *activitiesProfilePrediction* are presented in Figures 49c and 49b, respectively.

Services architecture diagram

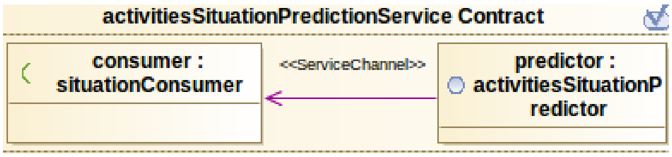
The services architecture for the business process service *BPSv002 - Activities of daily life manager* is illustrated in Figure 50. The service architecture details all participants required to execute the business process in Figure 40, and thus, to achieve the high-level mission defined in *HomecARe*: *GM1.1.2 - Activities of daily life supported*. Participants are coordinated through specific services interfaces and constrained by contracts and protocols defined for each inter-



(a) Service interface.

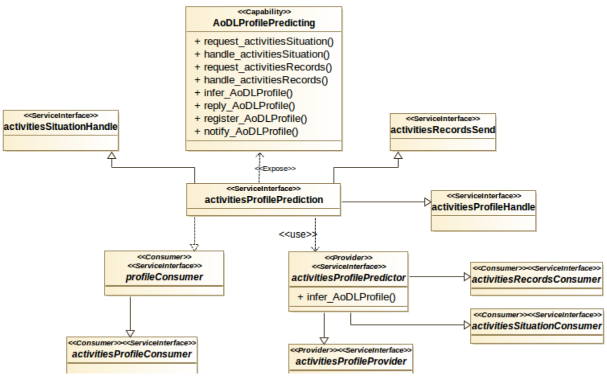


(b) Service protocol.

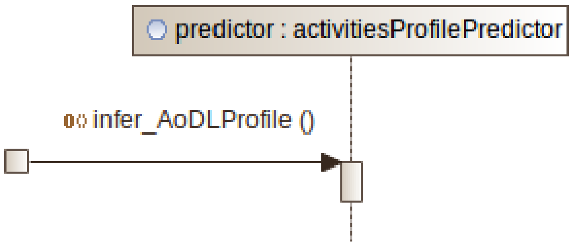


(c) Service contract.

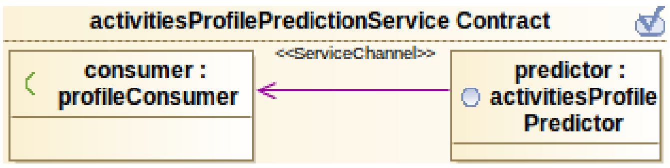
Figure 48 – Service interface, contracts and protocol diagrams for the capability *activitiesSituationPredicting*



(a) Service interface.



(b) Service protocol.



(c) Service contract.

Figure 49 – Service interface, contracts and protocol diagrams for the capability *AoDLProfilePredicting*

face. Services interfaces, contracts and protocols relating with this services architecture were introduced in the last section.

Aiming the reuse of service interfaces, contracts and protocols, the participating services *Leisure activities monitor*, *Social activities monitor*, *Personal care activities monitor* and *Domestic activities monitor* were abstracted into a more general participant, the *Patient activities monitor*, such as presented in the participants diagram in Figure 41. Such diagram simplifies relationships between participating services in the architecture of BPSv002.

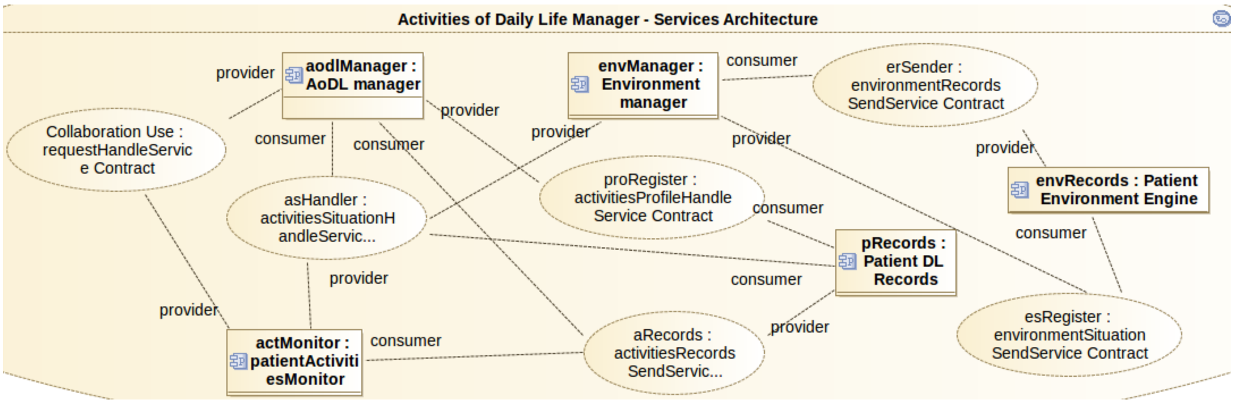


Figure 50 – Services architecture for the business process service *BPSv002 - Activities of daily life manager.*

MV4 - Chronic Disease Managed and Patient Quality of Life Improved

This mission view offers the knowledge contained in *HomecARe* for achieving the domain requirements *DR03* and *DR05* introduced in Section 4.2. Moreover, this view details how the two high-level missions of *HomecARe*, i.e., *GM1.1.1 - Chronic disease successfully managed* and *GM1.1 - Patients quality of life improved*, can be accomplished, respectively, by the two business process services *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor*. The BPSv007 aims to establish the situation of patient health condition based on her/his physical status, predicted sign and symptoms (that are notified by the BPSv001 in the mission view MV1 in Section 5.2), and interventions situation (provided by the BPSv008 in the mission view MV2 in Section 5.2). Meanwhile, the BPSv009 intends to determine patient situation regarding her/his quality of life, based on information obtained from patient AoDL profile (stated by the BPSv002 in the mission view MV3 in Section 5.2) and health conditions (established by the BPSv007 detailed in this mission view).

Relationships between this mission view, with significant domain requirements, high-level missions of *HomecARe*, and business process services were presented in Table 11. Diagrams used to represent this mission view, i.e., business process, capabilities, service interfaces, contracts, protocols and services architecture for BPSv007 and BPSv009 are detailed as follows.

Business process diagram

The business process presented in Figure 51 details the work-flow that must be executed by the business process services BPSv007 and BPSv009. Such process exposes activities and interactions between participating services to establish patient's situations regarding her/his health condition and QoL in *HomecARe*. As showed in Figure Figure 51, this process involves two additional participants besides the BPSv007 and BPSv009, the *Patient DL records*, which stores and provides the patient's health records, patient's health condition situation and patient's quality of life (QoL) situation, and the *Inference engine* which offers inference rules to BPSv007 and BPSv009 to predict patient's health condition situation and QoL situation.

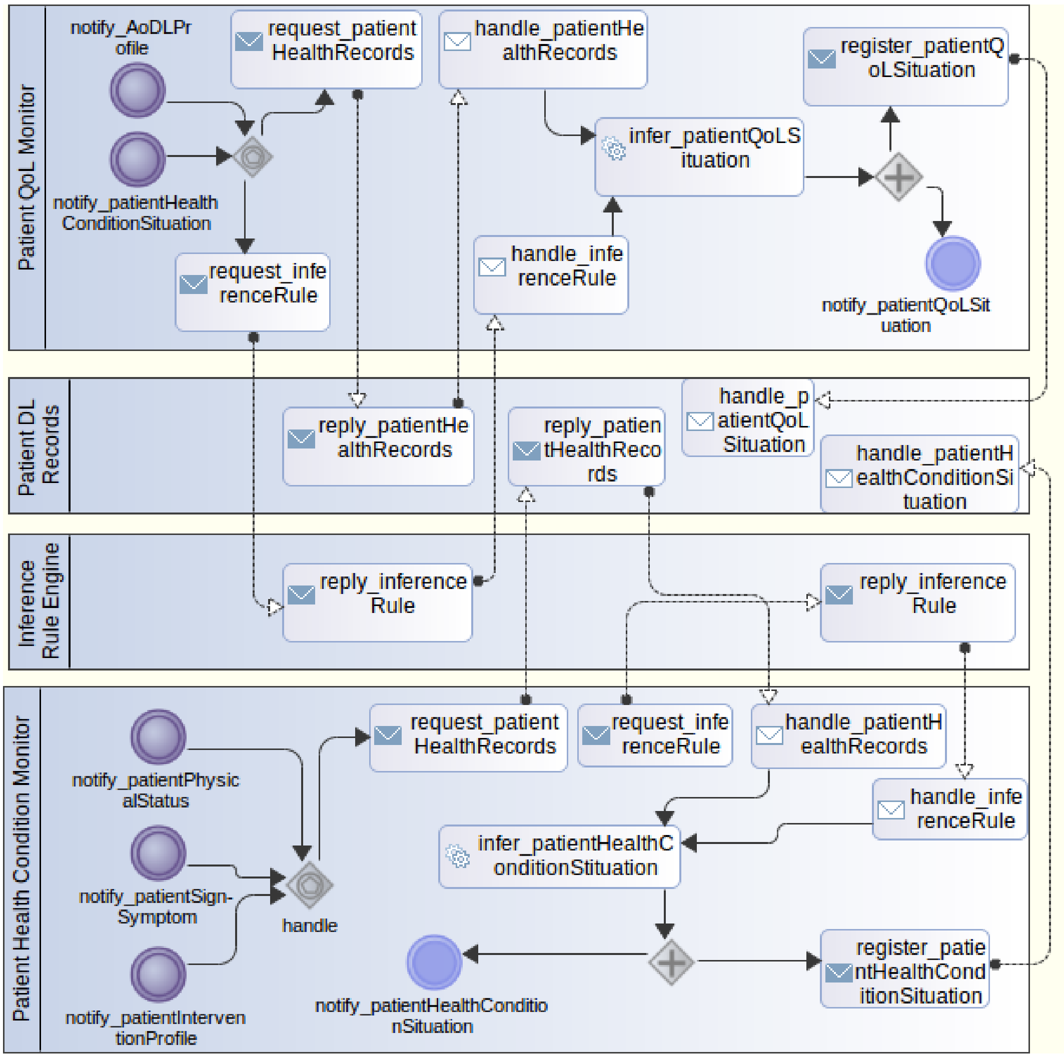


Figure 51 – Process diagram involving the business process services *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor*.

Capabilities diagram

Capabilities required to perform the business process in Figure 51, are depicted in Figure 52. Such capabilities allow the realization of the business process services BPSv007 and BPSv009. Capabilities in Figure 52 are related to activities performed by participating services as established in the business process in Figure 51.

Services interfaces, contracts and protocols diagrams

For each capability in the capability diagram in Figure 52, a service interface, contract and protocol were established, aiming the choreography of participants to execute the business process in Figure 51. Diagrams of service interfaces, contracts and protocols for each capability are presented as follows. It is worth mentioning that the capability *inferenceRuleSending* was introduced in the MV1, specifically in Section 5.2, Figure 22a, for this reason it is not detailed in this view.

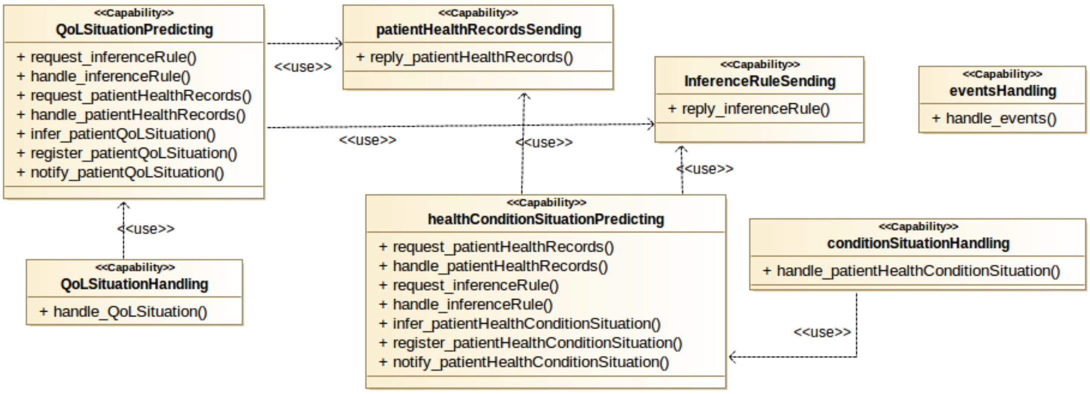
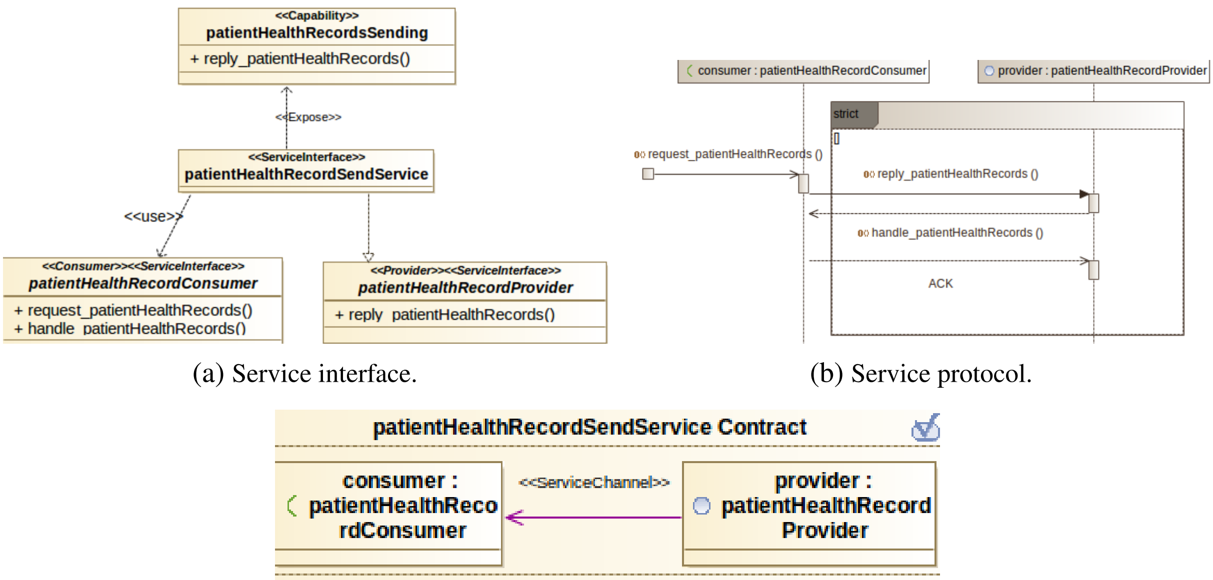


Figure 52 – Capabilities diagram of the business process services *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor*.

Figure 53a depicts the service interface *patientHealthRecordSendService* correspondent to the capability *patientHealthRecordsSending*. This service interface allows the transference of patient health records from a provider to a consumer, through the respective interfaces *patientHealthRecordProvider* and *patientHealthRecordConsumer*. For the business process in Figure 51, both business process services *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor* are consumers of patient health records. The provider of such records is the *Patient DL records*. Contract and protocol for the service interface *patientHealthRecordSendService* are presented in Figures 53c and 53b, respectively.



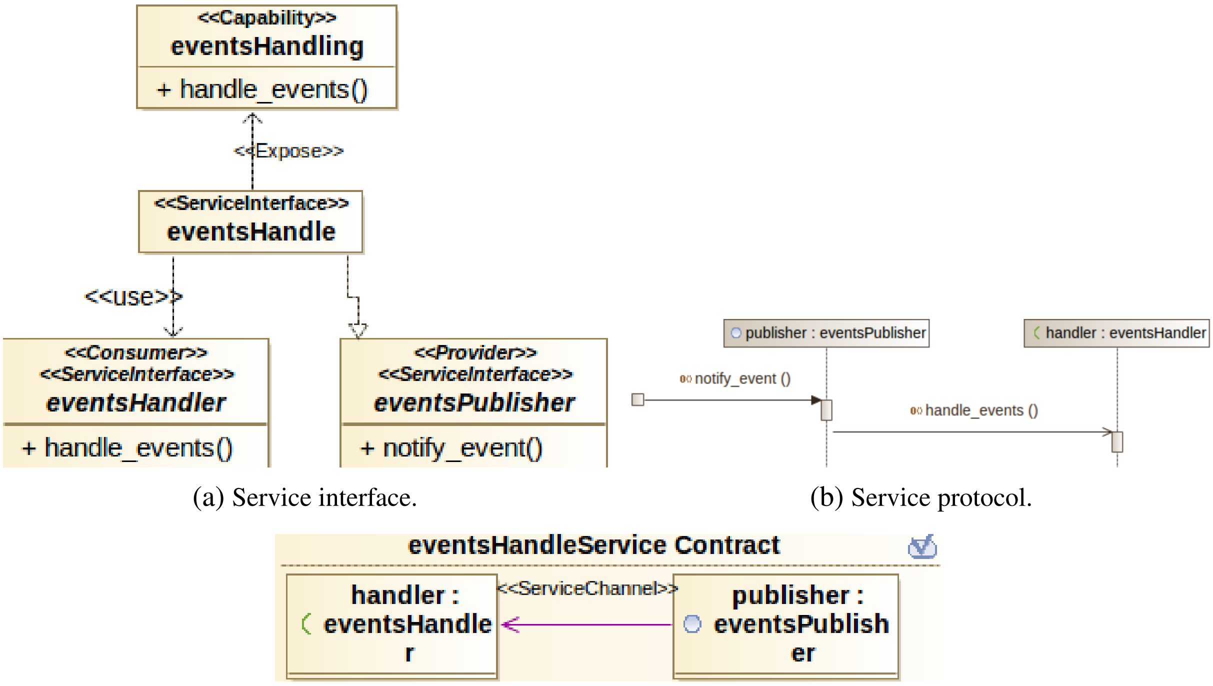
(a) Service interface.

(b) Service protocol.

(c) Service contract.

Figure 53 – Service interface, contracts and protocol diagrams for the capability *patientHealthRecordsSending*

For the capability *eventsHandling* the service interface *eventsHandle* is proposed in Figure 54a. The provider *eventsPublisher*, notifies an event to a consumer, *eventsHandler* that must manage such an event. For the business process in Figure 51, the *Health condition monitor* notifies the patient health condition situation, that must be handled by the *Quality of life monitor*. Contract and protocol for the service interface *eventsHandle* are depicted in Figures 54c and 54b.



(a) Service interface.

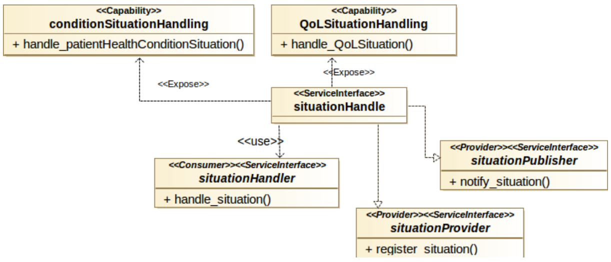
(b) Service protocol.

(c) Service contract.

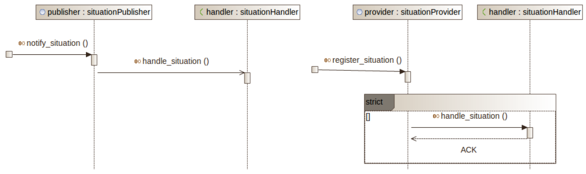
Figure 54 – Service interface, contracts and protocol diagrams for the capability *eventsHandling*

The service interface *situationHandle* is proposed to allow the capabilities *conditionSituationHandling* and *QoLSituationHandling*, as presented in Figure 55a. Through this service interface, a provider *situationPublisher* or *situationProvider* can notifies or registers a situation about patient health condition or patient quality of life. Whereas a consumer *situationHandler* can receive such situations. Contract and protocol for the service interface *situationHandle* are presented in Figures 55c and 55b, respectively.

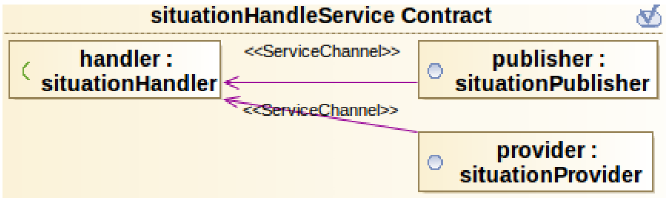
Figure 56a illustrates the service interface *situationPrediction* for the capabilities *healthConditionSituationPredicting* and *QoLSituationPredicting*. This service interface inherits operations from capabilities *patientHealthRecordsSendService* and *inferenceRuleSendService*, detailed in Figures 53a and 22a. Therefore, providers and consumers of the service interface *situationPrediction*, also inherits properties from providers and consumers from the service interfaces *patientHealthRecordsSendService* and *inferenceRuleSendService*. The service interface *situationPrediction* allows to a provider, *situationProvider*, infer a situation about patient health conditions or QoL, and communicate such situations to a consumer, *situationConsumer*, that must handle such situations. Contract and protocol to the service interface *situationPrediction* are illustrated in Figures 56c and 56b, respectively.



(a) Service interface.

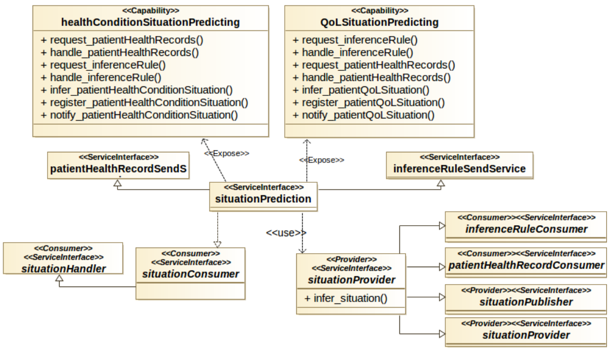


(b) Service protocol.

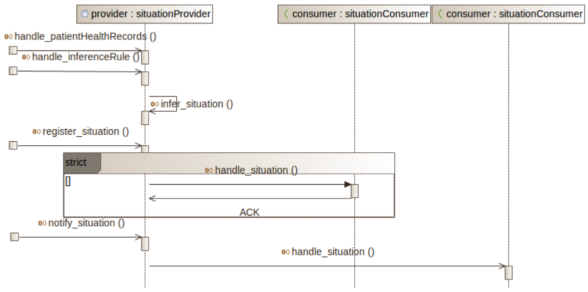


(c) Service contract.

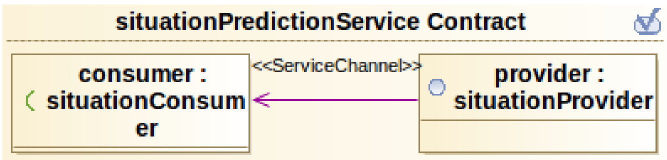
Figure 55 – Service interface, contracts and protocol diagrams for the capabilities *conditionSituationHandling* and *QoLSituationHandling*.



(a) Service interface.



(b) Service protocol.



(c) Service contract.

Figure 56 – Service interface, contracts and protocol diagrams for the capabilities *healhtConditionSituationPredicting* and *QoLSituationPredicting*.

Services architecture diagram

The services architecture presented in Figure 57 details all participants required to execute the business process in Figure 51. Specifically, such architecture was proposed to address the two high-level missions defined in *HomecARe*: *GM1.1.1 - Chronic disease successfully managed* and *GM1.1 - Patients quality of life improved*. For this, two business processes services participate of this architecture: *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor*. Moreover, the services architecture in Figure 57 describes how participants interact through services interfaces, and which contracts rule such interactions. Service interfaces, contract and protocols for this service architecture were presented in last section. As noticed, in the services architecture were included additional participating services, i.e., *AoDL manager*, *patient physical analyser*, and *intervention manager*, since they provide information required by the *QoL monitor* and the *health condition monitor* to execute their internal functionalities. The *AoDL manager*, *patient physical analyser*, and *intervention manager* were introduced in the mission views MV3, MV2 and MV1, respectively.

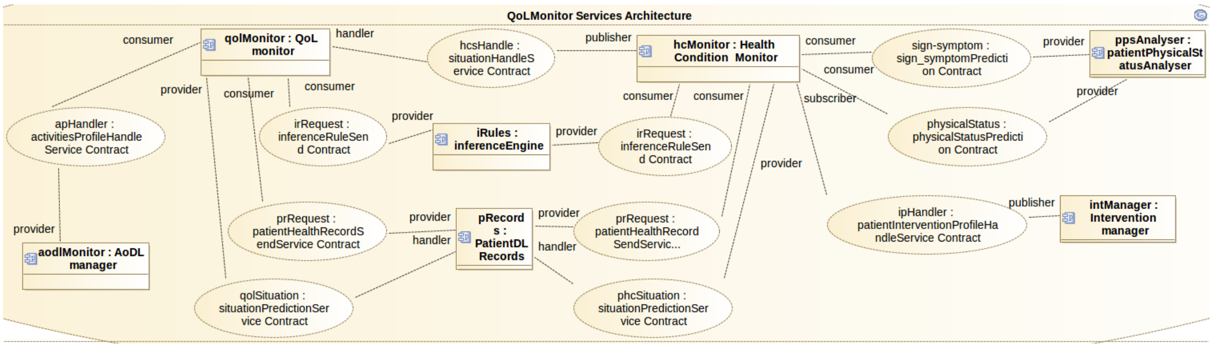


Figure 57 – Services architecture involving the business process services *BPSv007 - Health condition monitor* and *BPSv009 - Quality of life monitor*.

MV5 - Emergency Problem Alerted and Interventions in Emergency Situations Executed

This mission view aims to represent how *HomecARe* achieves the two domain requirements *DR06* and *DR07*, which are directly related with the high-level missions of *HomecARe*: *GM1.2.1 - Emergency problems alerted* and *GM1.2.3 - Immediate interventions in emergency situations executed*. Domain requirements of *HomecARe* were detailed in Section 4.2, and high-level missions in Section 4.1. To achieve the domain requirements, DR06 and DR07, and hence, the corresponding high-level missions, GM1.2.1 and GM1.2.3, the business process service *BPSv011 - Emergency manager* was proposed in *HomecARe*.

Relationships between this mission view, with significant domain requirements of *HomecARe*, high-level missions of *HomecARe*, and business process services were presented in Table 11. Diagrams used to represent this mission view, i.e., business process, capabilities, service interfaces, contracts, protocols and services architecture for BPSv011 are detailed as follows.

Business process diagram

The diagram presented in Figure 58 details the work-flow that the BPSv011 must execute to predict an emergency situation and to establish an emergency plan to be executed. In this context, for achieving the intended missions, collaborations between the following participants mus be allowed:

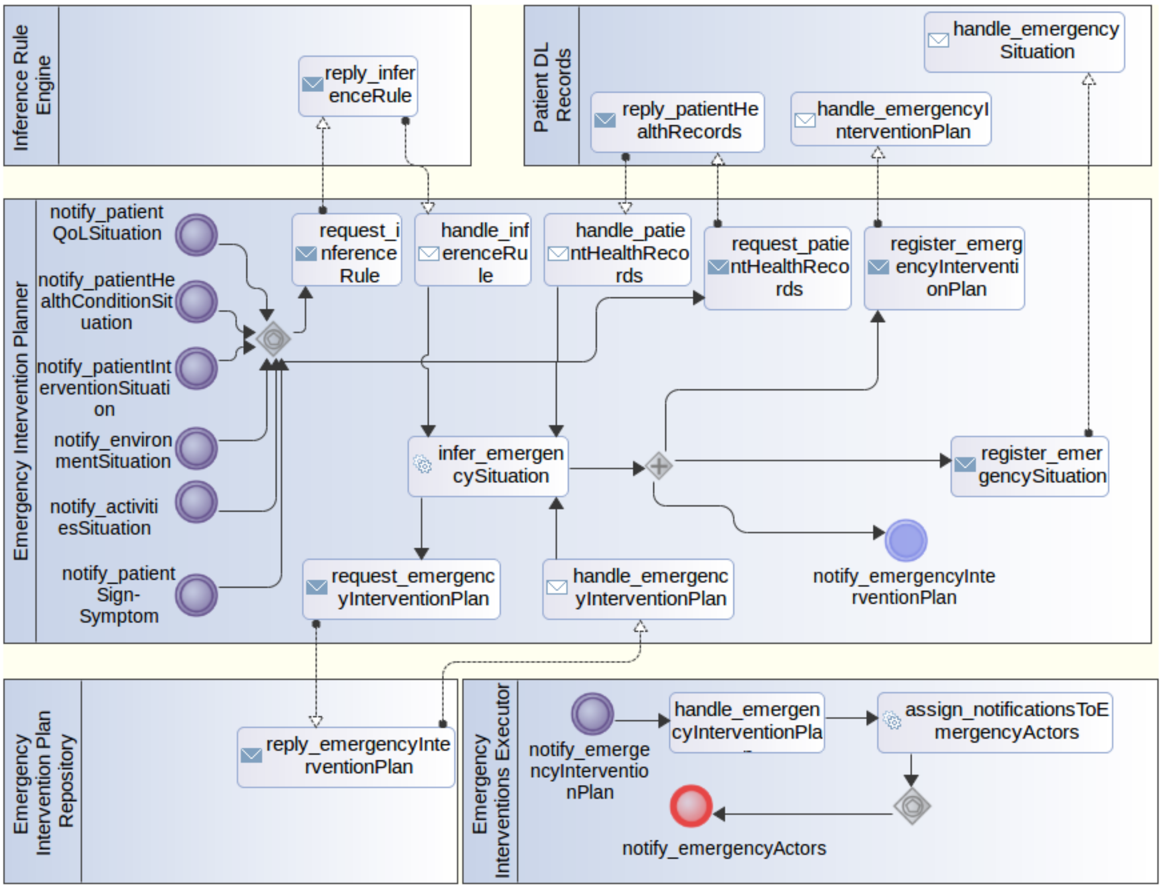


Figure 58 – Process diagram for the the business process service *BPSv011 - Emergency manager*.

- *Emergency intervention planner*, intends to define an intervention plan when an emergency occurs. Examples of emergency situations is a patient fall, injury, unconsciousness state, panic status, or when some environment problem arise, such as, fire, flooding, or gas leak. Emergency events must be identified by this participant based on situations by other services regarding QoL, health condition, medical interventions, activities, sign and symptoms, and environment of the patient.

- *Emergency interventions executor*, receives the intervention plan proposed by the *Emergency intervention planner*, and assigns specific notifications that are further send to emergency actors, e.g., ambulances, fire department, or homecare team members.

- *Emergency interventions plans repository*, stores intervention plans that provide actions to be executed considering the type of emergency situation occurred in patient's home.

- *Inference rule engine*, offers inference rules to predict which is the best intervention plan to be executed depending of the identified emergency situation, and

- *Patient DL records*, provides patient health records, and registers the detected emergency situations and the intervention plans proposed to attend the occurred emergency.

Capabilities diagram

Capabilities required to perform the business process in Figure 58, are depicted in Figure 59. Such capabilities allow the realization of the business process service BPSv011.

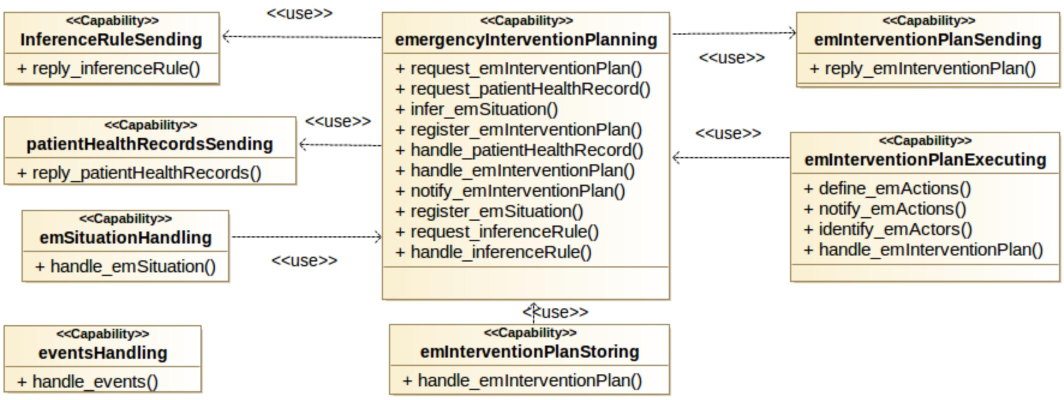


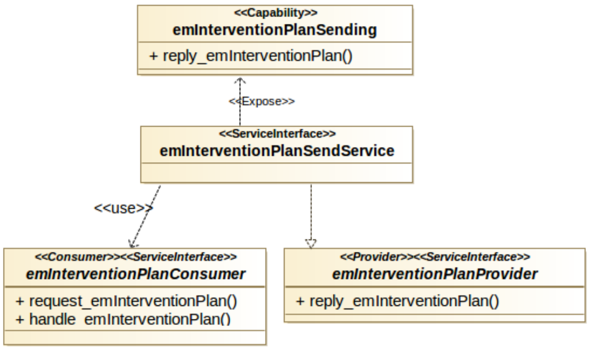
Figure 59 – Capabilities diagram for the business process service *BPSv011 - Emergency manager*.

Services interfaces, contracts and protocols diagram

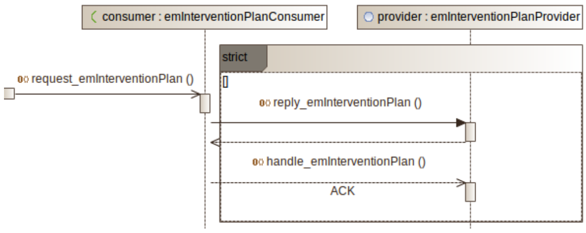
For each capability in the capability diagram in Figure 59, a service interface, contract and protocol were established, aiming the choreography of participants to execute the business process in Figure 58. Diagrams of service interfaces, contracts and protocols for each capability are presented as follows. It is worth mentioning that capability *inferenceRuleSending* was introduced in the MV1, specifically in Section 5.2, Figure 22. Moreover, capability *patientHealthRecordsSending* was described in the MV4, precisely in Section 5.2, Figure 53. Similarly, in the MV4, the capability *eventsHandling* was detailed, in Figure 54. Other capabilities showed in Figure 59, are reported as follows.

For realizing the capability *emInterventionPlanSending* was proposed the service interface *emInterventionPlanSendService* showed in Figure 60a. Using such interfaces, a provider *emInterventionPlanProvider* can send an intervention plan for emergency situations to a consumer *emInterventionPlanConsumer*. As showed in the business process in Figure 58, a possible consumer of this service interface can be the *Emergency intervention planner*, and the *Emergency intervention plan repository* can act as a provider. Contract and protocol for this service interface are presented in Figures 60c and 60b, respectively.

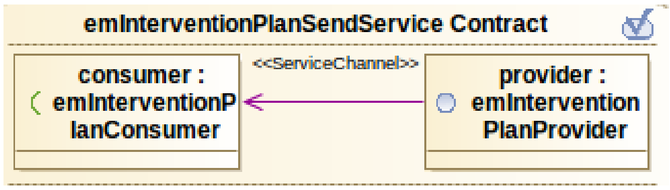
The service interface *emInterventionPlanStoreService* is proposed to realize the capability *emInterventionPlanStoring* showed in Figure 61a. A consumer of this service interface, *emInterventionPlanConsumer*, can be the *Patient DL records*, and a provider, *emInterventionPlanProvider* can be either the *Emergency intervention plan repository* or the *Emergency intervention planner*. Contract and protocol for this service interface are presented in Figures 61c and 61b.



(a) Service interface.

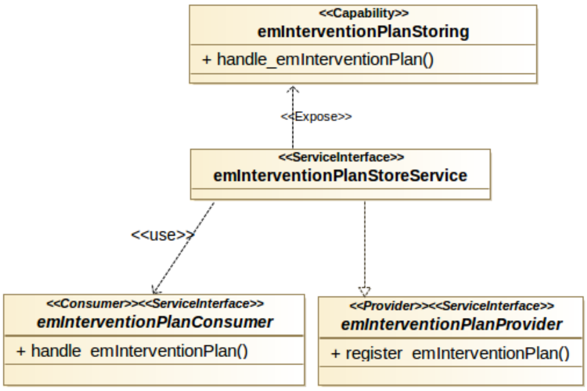


(b) Service protocol.

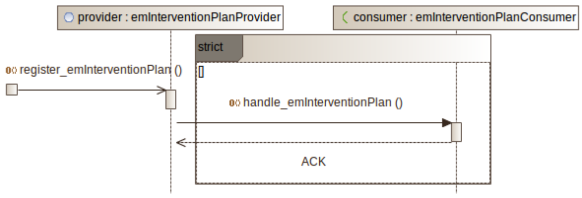


(c) Service contract.

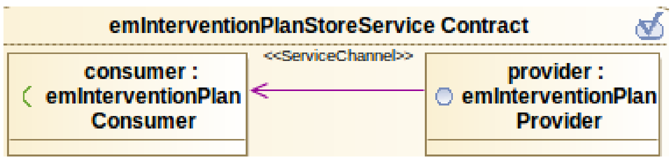
Figure 60 – Service interface, contracts and protocol diagrams for the capability *emInterventionPlanSending*



(a) Service interface.



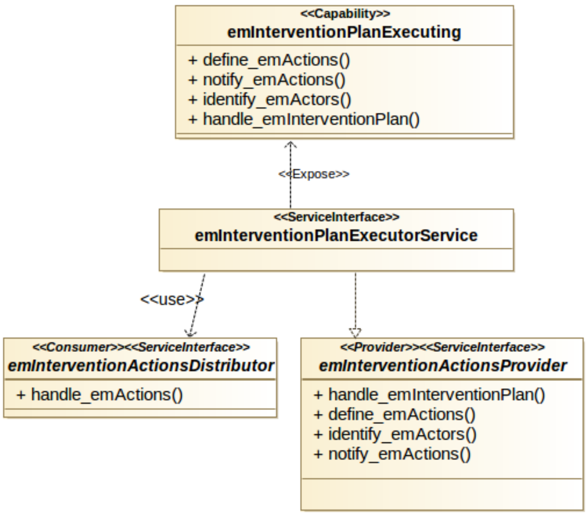
(b) Service protocol.



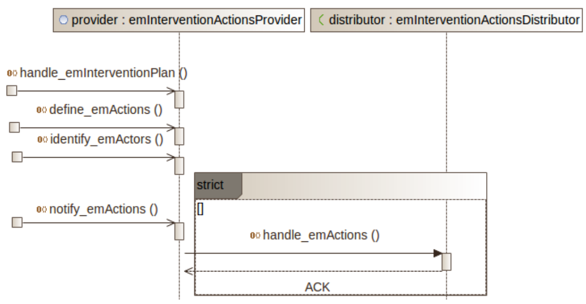
(c) Service contract.

Figure 61 – Service interface, contracts and protocol diagrams for the capability *emergencyIntervention-Planning*

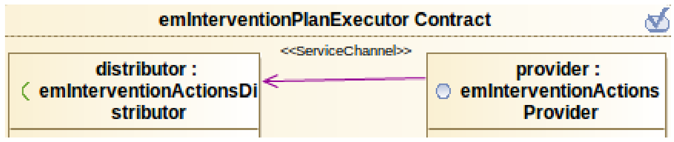
For the capability *emInterventionPlanExecuting*, the service interface *emIntervention-PlanExecutorService* was proposed, as showed in Figure 65a. The *Emergency interventions executor* uses the interface *emInterventionActionsProvider* to handle the intervention plan, define actions and responsible for executing each action, and communicate them to a consumer, i.e., the *interceptor* in the HSB, that using the interface *emInterventionActionsDistributor* receives such actions for further distribution to interested participants. Contract and protocol related to the service interface *emInterventionPlanExecutorService* are presented in Figures 62c and 62b, respectively.



(a) Service interface.



(b) Service protocol.



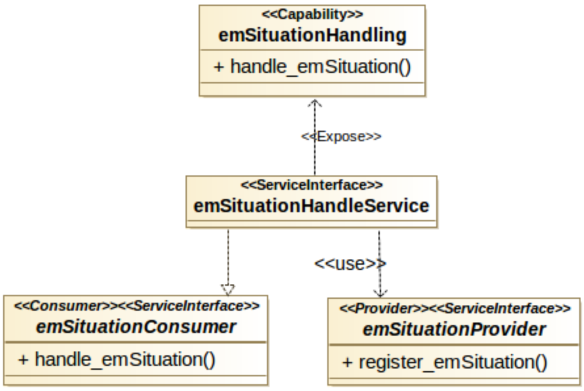
(c) Service contract.

Figure 62 – Service interface, contracts and protocol diagrams for the capability *emInterventionPlanExecuting*

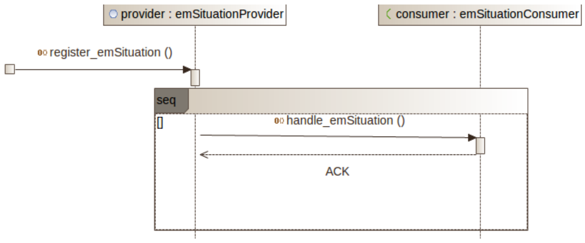
The service interface *emSituationHandleService* was propose to realize the capability *emSituationHandling*, as showed in Figure 63a. A provider, e.g., the *Quality of life monitor* or *Health condition monitor*, uses the interface *emSituationProvider* to notify an emergency situation. The consumer, i.e., *Emergency intervention planner*, uses the interface *emSituationConsumer* to handle such situations. Contract and protocol associated to the service interface *emSituationHandleService*, are presented in Figures 63c and 63b, respectively.

For realizing the capability *emergencyInterventionPlanning*, three service interfaces are proposed: (i) the *emSituationHandleService* that was aforementioned described (See Figure 63), (ii) the *emInterventionPlannerService* that is presented in Figure 64a, and (iii) the *emInterventionPlanNotificationService* that is detailed in Figure 65a.

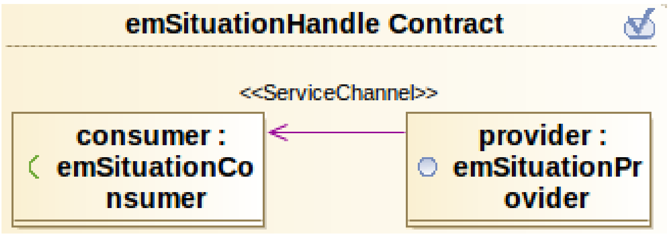
The service interface *emInterventionPlannerService* inherits operations from three services interfaces, as showed in Figure 64a: the *patientHealthRecordSendService*, *inferenceRuleSendService* and *emInterventionPlanSendService* presented in Figures 53a, 22a and 60a, correspondingly. Contract and protocol for the service interface *emInterventionPlannerService* are illustrated in Figures 53c and 53b, respectively.



(a) Service interface.

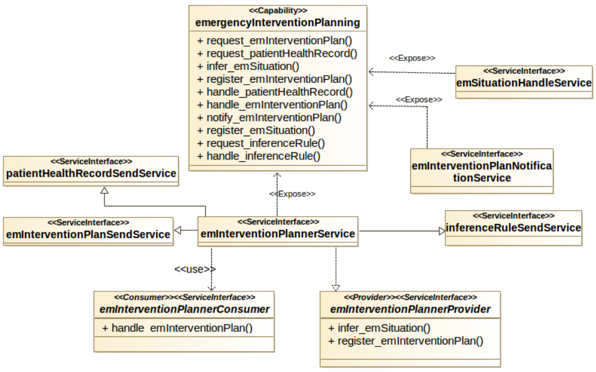


(b) Service protocol.

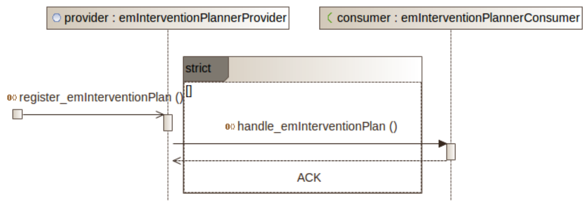


(c) Service contract.

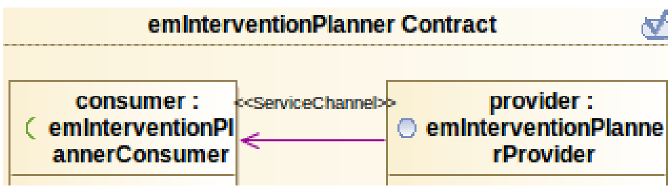
Figure 63 – Service interface, contracts and protocol diagrams for the capability *emSituationHandling*



(a) Service interface.



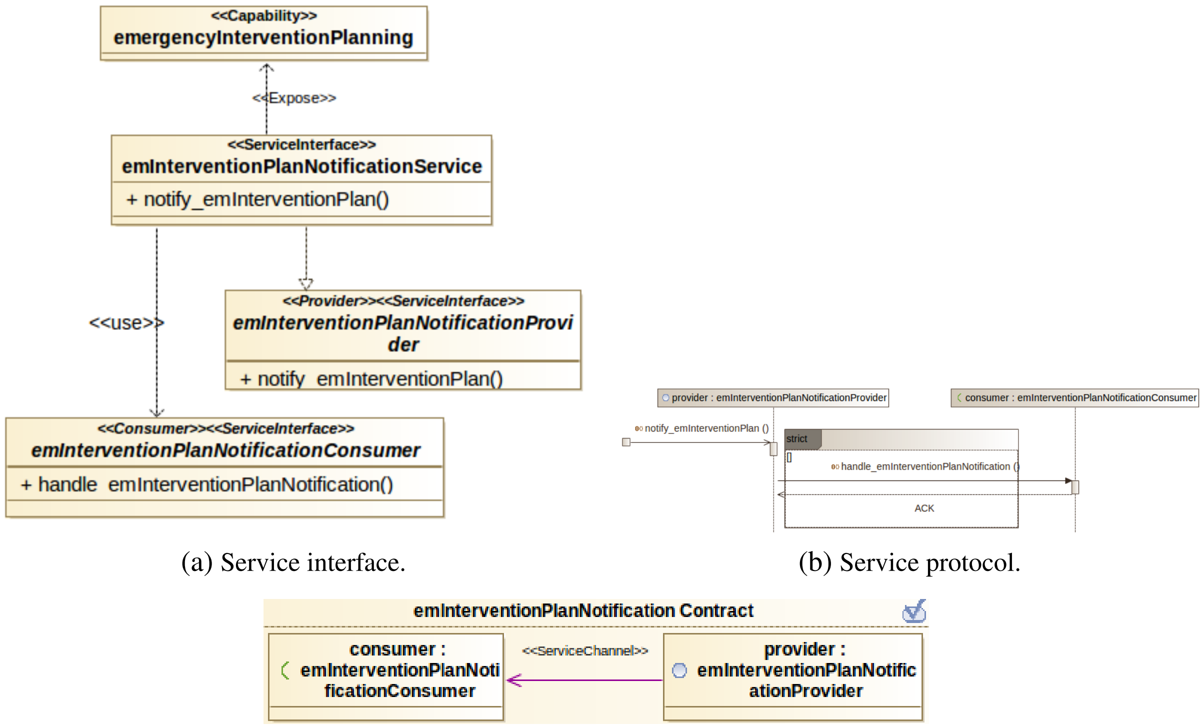
(b) Service protocol.



(c) Service contract.

Figure 64 – Service interface, contracts and protocol diagrams for the capability *emergencyIntervention-Planning*

Finally, the service interface *emInterventionPlanNotificationService* complements the capability *emergencyInterventionPlanning*, as showed in Figure 65a. A provider, i.e., *Emergency intervention planner*, uses the interface *emInterventionPlanNotificationProvider* to notifies about the selected intervention plan to resolve an emergency situation. The consumer, using the interface *emInterventionPlanNotificationConsumer*, handles such plan. Contract and protocol related to the service interface *emInterventionPlanNotificationService* are presented, respectively, in Figures 65c and 65b.



(a) Service interface.

(b) Service protocol.

(c) Service contract.

Figure 65 – Service interface, contracts and protocol diagrams for the capability *emergencyIntervention-Planning*

Services architecture diagram

The services architecture presented in Figure 66 details participants required to execute the business process defined in Figure 58. Specifically, such architecture was proposed to address two high-level missions defined in *HomecARe*:*GM1.2.1 - Emergency problems alerted* and *GM1.2.3 - Immediate interventions in emergency situations executed*. Hence, the services architecture details the internal structure of the business process service *BPSv011 - Emergency manager*. Moreover, service architecture in Figure 66 describes how participants interact through services interfaces, and which contracts rule such interactions. Service interfaces, contract and protocols for this service architecture were presented in last section.

MV6 - Formal Care Improved

This mission view aims to represent how *HomecARe* achieves the domain requirements *DR08*, which are directly related with the high-level missions of *HomecARe*: *GM1.2 - Formal care improved*. Domain requirements of *HomecARe* were detailed in Section 4.2, and high-level

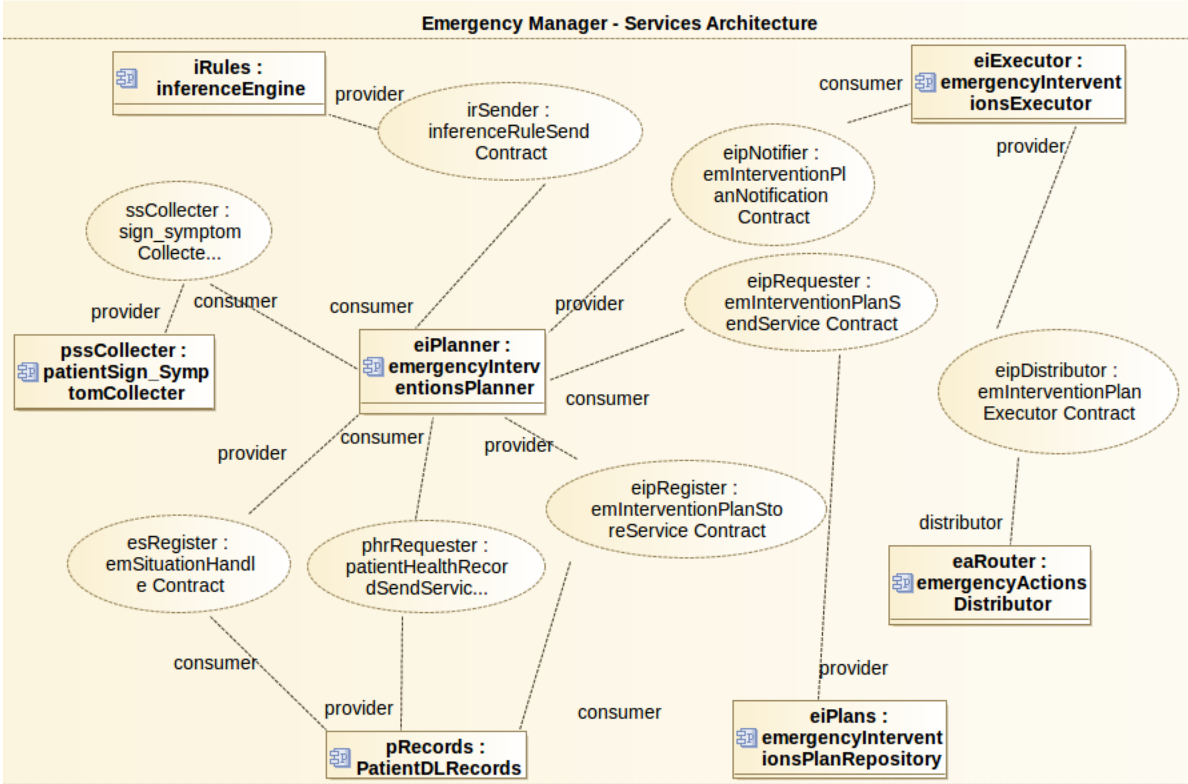


Figure 66 – Services architecture for the business process service *BPSv011 - Emergency manager*.

missions in Section 4.1. To achieve the domain requirement DR08, and hence, the corresponding high-level missions, GM1.2, the business process service *BPSv010 - Formal care manager* was proposed in *HomecARe*.

Relationships between this mission view, with significant domain requirements, high-level missions of *HomecARe*, and business process services were presented in Table 11. Diagrams used to represent this mission view, i.e., business process, capabilities, service interfaces, contracts, protocols and services architecture for BPSv010 are detailed as follows.

Business process diagram

The diagram presented in Figure 67 depicts the work-flow required to establish the formal care profile of a patient. Such profile compiles information about patient's situations regarding her/his QoL, health condition, and AoDL. To establish such profile, the *Formal care monitor* uses inference rules contained in the *Inference rule engine* and patient health records obtained from the participant *Patient DL records*. Once the profile is inferred, it is stored or notified to interested members of the home healthcare team.

Capabilities diagram

Capabilities required to perform the business process in Figure 67, are depicted in Figure 68. Such capabilities allow the realization of the business process service BPSv010.

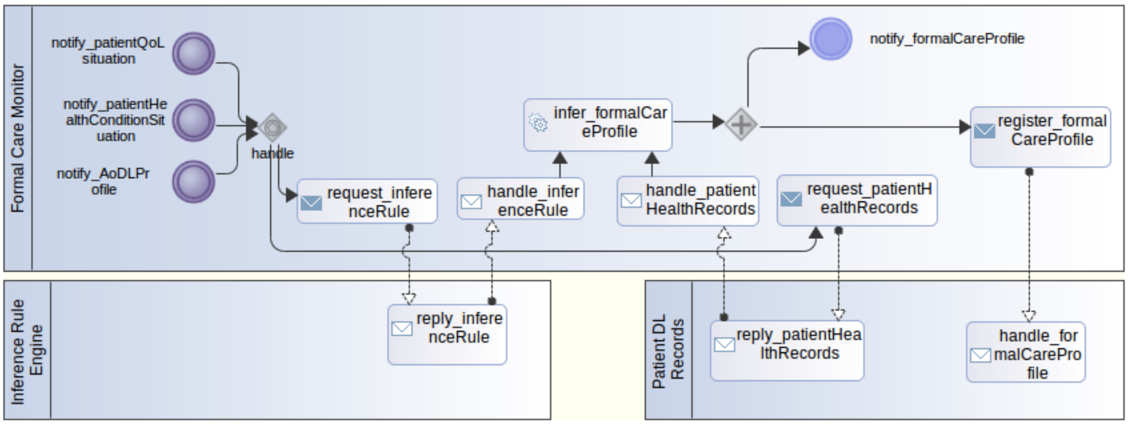


Figure 67 – Process diagram for the business process service *BPSv010 - Formal care manager*

.

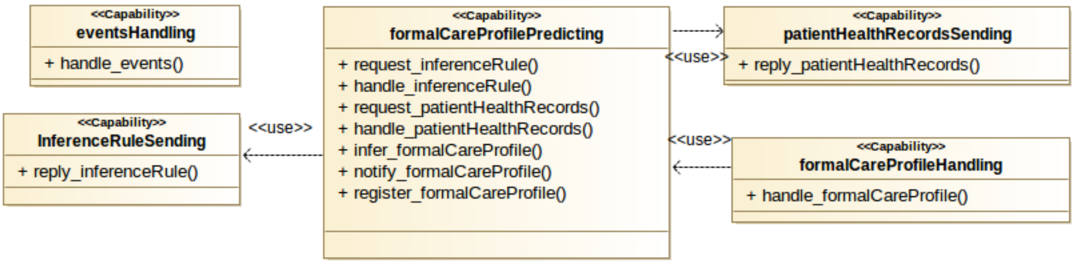
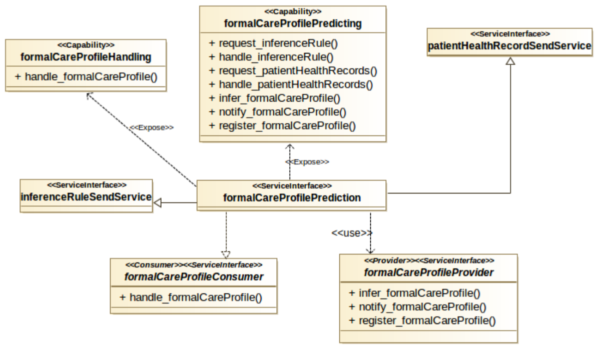


Figure 68 – Capabilities diagram for the business process service *BPSv010 - Formal care manager*

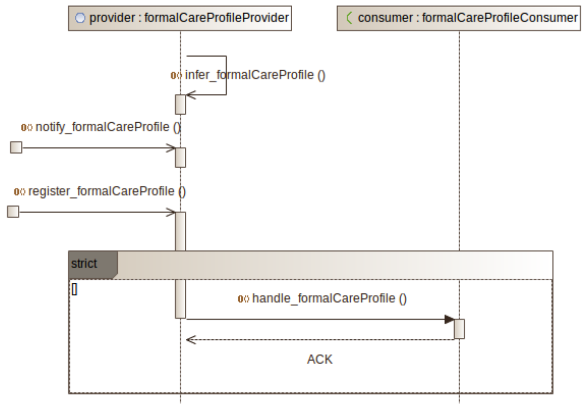
Services interfaces, contracts and protocols diagram

For each capability in the capability diagram in Figure 68, a service interface, contract and protocol were established, aiming the choreography of participants to execute the business process in Figure 67. Diagrams of service interfaces, contracts and protocols for each capability are presented as follows. It is worth mentioning that capability *inferenceRuleSending* was introduced in the MV1, specifically in Section 5.2, Figure 22. Moreover, capability *patientHealthRecordsSending* was described in the MV4, precisely in Section 5.2, Figure 53. Similarly, in the MV4, the capability *eventsHandling* was detailed, in Figure 54. Other capabilities showed in Figure 68, are reported as follows.

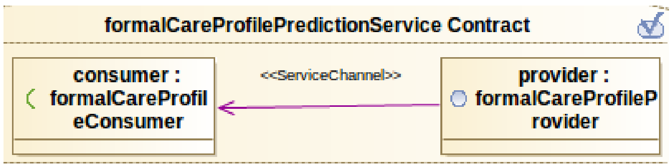
For realizing the capabilities *formalCareProfilePredicting* and *formalCareProfileHandling*, the service interface *formalCareProfilePrediciton* was proposed, as depicted in Figure 69a. Such service interface inherits operations from other two service interfaces, the *inferenceRuleSendService* and the *patientHealthRecordSendService*, described in Figures 22a and 53a, respectively. The service interface *formalCareProfilePrediciton* allows to a provider (i.e., the *Formal care monitor*), through *formalCareProfileProvider* interface, infers, notifies and registers the formal care profile of a patient. Similarly, a consumer (i.e., a service of the healthcare team), through the *formalCareProfileConsumer*, can handle such profile. Contract and protocol of the service interface *formalCareProfilePrediciton* are presented in Figures 69c and 69b, respectively.



(a) Service interface.



(b) Service protocol.



(c) Service contract.

Figure 69 – Service interface, contracts and protocol diagrams for the capability *formalCareProfilePredicting*

Services architecture diagram

The services architecture presented in Figure 70 details participants required to execute the business process defined in Figure 67. Specifically, such architecture was proposed to address the high-level missions defined in *HomecARe*: *GM1.2 - Formal care improved*. Hence, the services architecture details the internal structure of the business process service *BPSv010 - Formal care manager*. Moreover, service architecture in Figure 70 describes how participants interact through services interfaces, and which contracts rule such interactions. Service interfaces, contract and protocols for this service architecture were presented in last section.

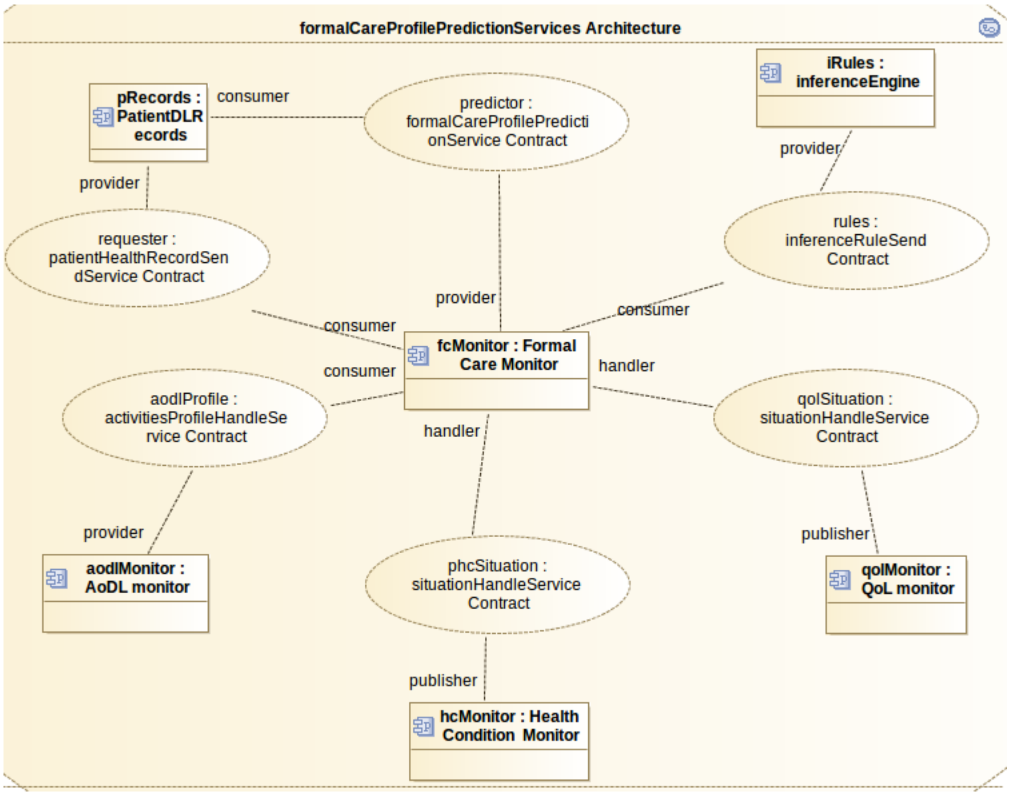


Figure 70 – Services architecture for the business process service *BPSv010 - Formal care manager*.

5.3. Quality Viewpoint

The quality viewpoint aims to describe the rationale for architectural decisions made in *HomecARe* to address quality attributes requirements regarding interoperability, adaptivity, security, and reliability, which were specified as ASRs of *HomecARe* in Section 4.2. Four quality views (QV) compose this architectural viewpoint as showed in Table 12. The *Integration and Interoperability View* (QV01) details how interoperability issues are addressed by *HomecARe* through the description of decisions made in the Health Service Bus (HSB). The *Adaptivity View* (QV02) describes the structure of the quality manager service, which is used to allow runtime adaptations in *HomecARe*. The *Security View* (QV03) details tactics used to protect patient information from unauthorized access and avoid repudiation of operations by participants in HSH systems. Finally, the *Reliability View* (QV04) presents mechanisms adopted in *HomecARe* to offer fault-tolerance, prevention of bottlenecks, and trusted operations for HSH systems. Table 12 shows details of each QV: ID, name, requirements of *HomecARe* that the QV is oriented to, and the diagrams used to represent the QV. The remainder of this section describes each QV in detail.

5.4.1 QV01 - Integration and Interoperability View

This view addresses the integration and interoperability requirements in *HomecARe*, i.e., II01, II02, and II03, specified in Section 4.2. Three diagrams are presented in this view: (i) an

Table 12 – Structure of the Quality Viewpoint of *HomecARe*

ID	Architectural View	Requirement	Diagrams	
QV01	Integration and Interoperability View. (Section 5.4.1)	II01; II02; II03	Integration diagram	Figure 71
			Broker structure diagram	Figure 72
			Interoperable structure messages	Figure 73
QV02	Adaptivity View. (Section 5.4.2)	AR07; AR08	Quality manager service diagram	Figure 74
QV03	Security View. (Section 5.4.3)	SR01; SR02	Authentication and authorization diagrams	Figures 75 and 76.
			Non-repudiation diagram	Figure 77
QV04	Reliability View. (Section 5.4.4)	RR01; RR02; RR03; RR04; RR05	Textual description of reliability tactics.	

integration diagram detailing how constituent systems services (CSSv) are incorporated as part of a HSH system, which is presented in Figure 71; (ii) a broker structure diagram presenting how are realized interoperable communications between a HSH system and external services (e.g., constituent systems or consumer services), detailed in Figure 72; and (iii) the structure of classes of interoperable messages exchanged in *HomecARe*, that can be instantiated in specific HSH systems, which is depicted in Figure 73. The three diagrams are presented as follows.

Integration Diagram

Figure 71 shows activities that must be executed to integrate, in a HSH system, a new service provided by a constituent system. A constituent system service, *cssv*, presents itself sending a *presentationMessage* to the *interceptor* in the HSB. The *presentationMessage* must contain information about: (i) capabilities offered that will be used to identify possible consumers of *cssv*; (ii) the standard for formatting messages used by the *cssv*, e.g., HL7 v2; (iii) the clinical terminology utilized by the *cssv*, e.g., SNOMED-CT or ICD-10; (iv) interfaces and contracts information; (v) transport layer protocols; and (vi) vendors, organizations, providers or devices information that allow to identify the physical entity responsible of offering the *cssv*. The *presentationMessage* must be formatted in XML (eXtensible Model Language)(BRAY *et al.*, 2008) for being understood by the *explorer*, a discovery service.

The *presentationMessage* is routed to the *explorer* located in the HSB that verifies if service interfaces and contracts of the *cssv* are in conformance to the specified in *HomecARe*. For this, the *explorer* consults service interfaces and contracts specifications in the *services-contractsReporsitory* located in the *Governance and policies layer*. Contracts and service interfaces considered in *HomecARe* are presented in the missions viewpoint in Section 5.2. When specifications of services interfaces and contracts of *cssv* do not accomplish the specifications in *HomecARe*, no integration can be done.

When capabilities offered by the *cssv* fulfil specifications of interfaces and contract in *HomecARe*, this service is registered in the *container*, that maintains updated records of all information of integrated services in the HSH system. Similarly, routing tables in the *interceptor*

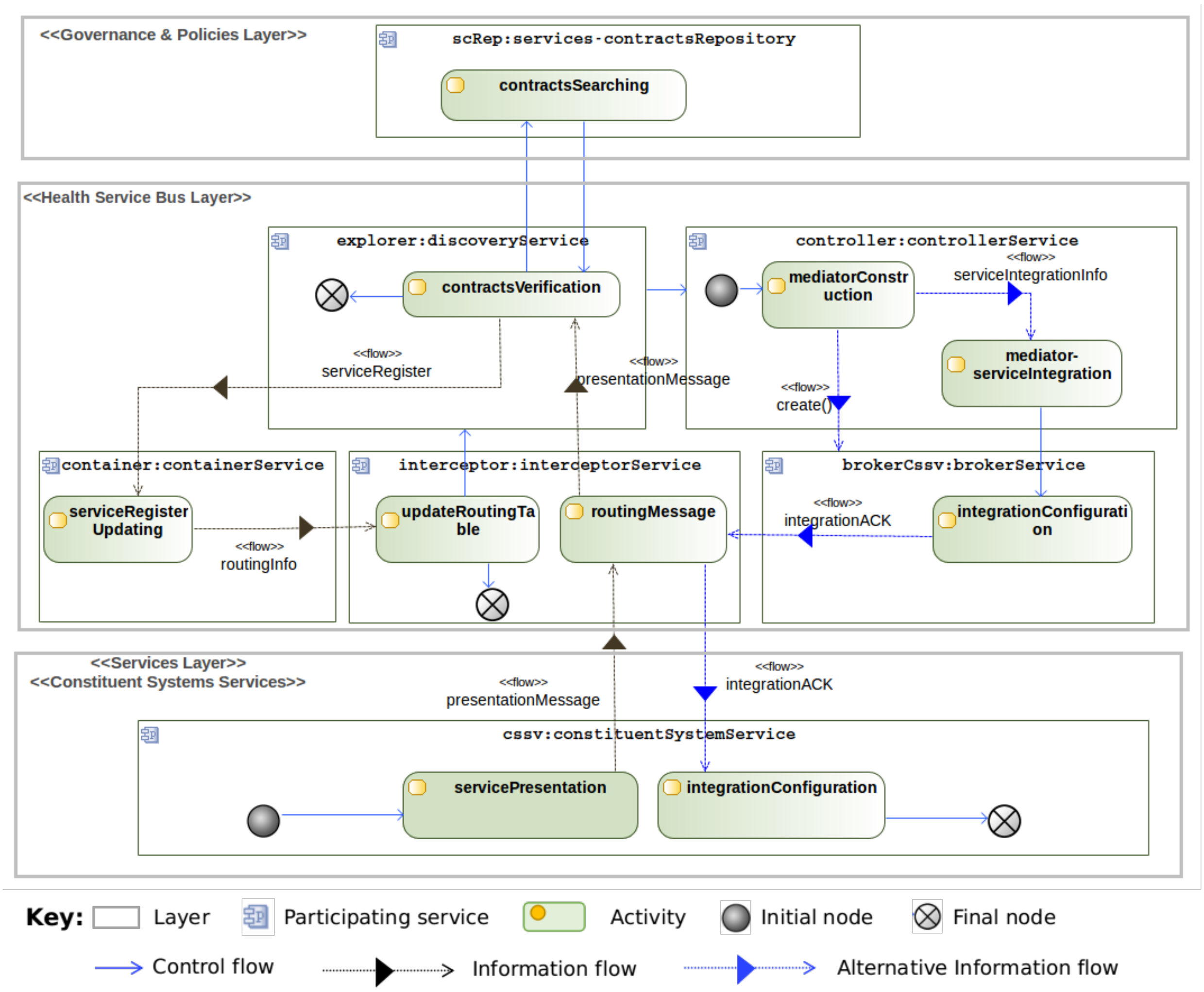


Figure 71 – Integration of new services in *HomecARe*.

must be updated including routing information for *cssv*, such as possible consumers/providers of capabilities offered by the *cssv*.

Whether standards to format its messages or clinical terminologies used by the *cssv* are different from those adopted in *HomecARe* (i.e., HL7 v2 and SNOMED-CT, respectively), a *broker* must be created by the *controller* to transform *cssv* messages to interoperable messages classes specified in *HomecARe*. An example of interoperable message in *HomecARe* is presented in Figure 73. Therefore, the *controller* constructs the *brokerCssv*, a mediator of type broker to facilitates interoperable communication between *cssv* and the HSB. Mediators classification was presented in Section 2.2. After creating the *brokerCssv*, it is configured to allow transformation of messages formats and clinical terminology used by *cssv* to those formats and terminologies used in *HomecARe*. Configuration parameters are specified in the *serviceIntegrationInfo*. Additionally, interfaces between *cssv* and dedicated *broker* are configured by the *controller*. Internal structure of a *broker* is presented in Figure 72. Finally, a message *integrationACK* is sent from the *brokerCssv* to the *cssv*, through the *interceptor*, to end the integration configuration of the *cssv* within the HSH system, allowing the *cssv* to start its interoperation with other services in the

HSH system.

Broker Structure Diagram

In its general structure, a *broker service* is composed of four services, a *collector service*, *filter service*, *translator service*, and a *wrapper service*, as showed in Figure 72. For each *cssv* operating in the HSH system, a dedicated *broker service* is allocated to it. When the *cssv* publishes a message, *cssvMessage*, instead of sending it directly to the *interceptor* in the HSB, such message is sent to the *collector service* in *brokerCssv2HSB*. The *collector* receives the *cssvMessage* and transmits it to the *decoder*, a type of *filter service*, whose purpose is to separate data corresponding to clinical terminology from those related to message format. Hence, the *decoder* must previously know the message format (e.g., any standard used for syntactical interoperability as presented in Section 3.3) used by the *cssv*, to be able of separating clinical contents from message structure. Therefore, the *decoder* sends a *decodedData* to the *interpreter*, a type of *translator service*, containing just the clinical terms comprised in the *cssvMessage*. Whether terms contained in the *decodedData* are inconsistent with clinical terminology used in *HomecARe*, i.e., SNOMED-CT, the *interpreter* translates terms in the *decodedData* to respective terms in SNOMED-CT. The *interpretedData* are sent to an *encoder*, a type of *wrapper service*, that aims to format the *interpretedData* in a message, *cssvEncodedMsg*, that follows the standard HL7 v2. The *cssvEncodedMsg* is published in the HSB being gathered by the *interceptor* for further routing to interested parts.

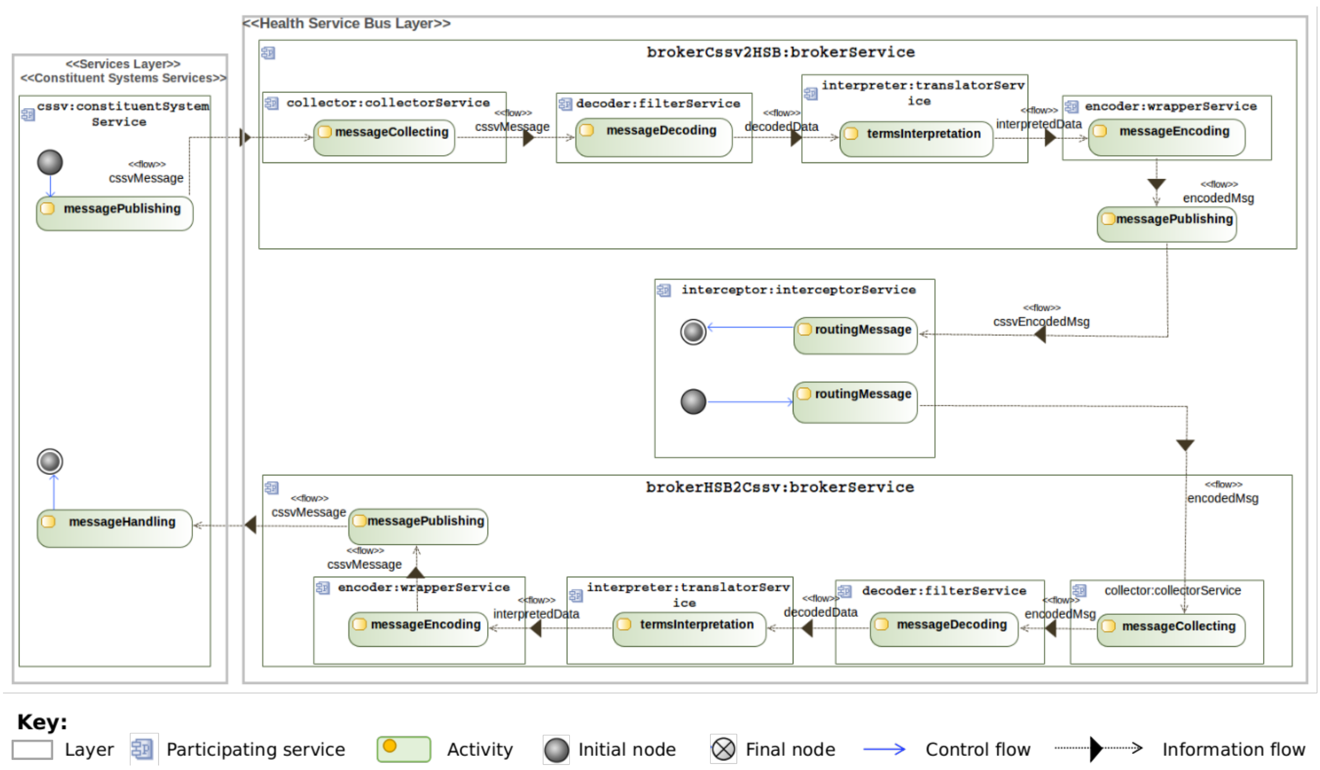


Figure 72 – Structure of a broker service in *HomecARe*.

Table 13 – ORU^R01 Message Structure (HEALTH LEVEL SEVEN INTERNATIONAL, 2015).

Segments	Description	Segments	Description
MSH	Message Header	[{ SFT }]	Software Segment
[UAC]	User Authentication Credential	{	— PATIENT_RESULT begin
[	— PATIENT begin	PID	Patient Identification
[PD1]	Additional Demographics	[{PRT}]	Participation (for Patient)
[{NTE}]	Notes and Comments	[{NK1}]	Next of Kin/Associated Parties
[{ARV}]	Access Restrictions	[{	— PATIENT_OBSERVATION begin
OBX	Observation (for Patient ID)	[{PRT}]	Participation (Observation Participation)
}]	— PATIENT_OBSERVATION end	[	— VISIT begin
PV1	Patient Visit	[PV2]	Patient Visit - Additional Info
[{PRT}]	Participation (for Patient Visit)	]	— VISIT end
]	— PATIENT end	{	— ORDER_OBSERVATION begin
[	— COMMON_ORDER begin	ORC	Order common
[{PRT}]	Participation (for Observation)	[	— ORDER DOCUMENT begin
OBX	Observation containing Document	[{PRT}]	Participation
TXA	Transcription Document Header	]	— ORDER DOCUMENT end
]	— COMMON ORDER end	OBR	Observations Request
{[NTE]}	Notes and comments	[{PRT}]	Participation (for Observation)
[{	— TIMING_QTY begin	TQ1	Timing/Quantity
[{TQ2}]	Timing/Quantity Order Sequence	}]	— TIMING_QTY end
[CTD]	Contact Data	[{	— OBSERVATION begin
OBX	Observation related to OBR	[{PRT}]	Participation (Observation Participation)
{[NTE]}	Notes and comments	}]	— OBSERVATION end
[{FT1}]	Financial Transaction	{[CTI]}	Clinical Trial Identification
[{	— SPECIMEN begin	SPM	Specimen
[{	— SPECIMEN_OBSERVATION begin	OBX	Observation (for Patient ID)
[{PRT}]	Participation (Observation Participation)	}]	— SPECIMEN_OBSERVATION end
}]	— SPECIMEN end	}	— ORDER_OBSERVATION end
}	— PATIENT_RESULT end	[DSC]	Continuation Pointer

When a message is transmitted from the HSB to a *cssv*, an inverse transformation must be done by a *broker*. In this situation, the *interceptor* sends a message, *encodedMsg*, formatted in HL7 v2, to the *collector*, that sends it to the *decoder*. The *decoder* filters HL7 structures from the *encodedMsg*, and transmits message content, *decodedData*, to the *interpreter*, who translates terms in SNOMED-CT to terminology used by the *cssv*. The *interpretedData* are transmitted to the *encoder* that wraps such data in the message format recognized by the *cssv*. Finally, the *encoder* sends a message, *cssvMessage*, that is understandable by the *cssv*.

Interoperable Messages Model

To achieve interoperable communications between services in a HSH system, messages exchanged through the HSB must be formatted following the standard HL7 v2. Specifically, in *HomecARe*, the class of message ORU^R01 defined in HL7 v2 is used. Table 13 lists all segments comprised in the ORU^R01 message structure.

Figure 73 illustrates an example of how an interoperable message in *HomecARe* could be structured. In this example, the constituent system service *CSSv001 - Physiological functions assessment* sends to a HSH system a vital sign measure of the patient. To ensure semantic interoperability of clinical terms, SNOMED-CT codes are included in the HL7 message. Thus, to communicate a vital sign, the CSSv001 must use the SNOMED-CT code *363789004* corresponding to a *General patient characteristic*.

In *HomecARe*, different classes of interoperable messages were defined, as the one showed in Figure 73, for each type of information provided or consumed by CSSv and BPSv through the HSB. In this context, depending of the exchanged information type, the ORU^R01 message must be formatted regarding the segments and structure defined in Table 13. Classes of interoperable messages in *HomecARe* can be consulted in (VICENTE; GARCÉS; NAKAGAWA, 2017c). Information types and the related SNOMED-CT codes associated to each CSSv and BPSv were presented in the conceptual view, in Section 5.1, Tables 8 and 10, respectively.

```
MSH|^~\&|^[cssv001]|[PHYSIOLOGICAL FUNCTIONS ASSESSMENT]|[HSH SYSTEM]^[HSH SYSTEM
CODE]||[DATE/TIME OF MESSAGE]||ORU^R01|[MESSAGE CONTROL ID]|P|2.8||||||||||||
PID|||||[PATIENTE FULL NAME]||||||[ADDRESS]|||||||||||||||||||||||||||||
OBR||||[61746007]^[TAKING PATIENT VITAL SIGNS]^SNM3|||[OBSERVATION DATE/TIME]||||
|||||||||||[DATE/TIME WHEN THE RESULTS WERE REPORTED]|||F|||||||||||||||||||||||
||||||
OBX||NM|[363789004]^[GENERAL PATIENT CHARACTERISTIC]^SNM3||[V007]|
[258666001]^[MEASURE UNIT]^SNM3|||||F|||[OBSERVATION DATE/TIME]|||[61746007]^[TAKING
PATIENT VITAL SIGNS]^SNM3|||||||||||||
```

Figure 73 – Structure of an interoperable message in *HomecARe*.

5.4.2 QV02 - Adaptivity View

This quality view presents solutions to address the adaptivity requirements in *HomecARe*, i.e., AR08 and AR09, which were specified in Section 4.2. The adaptivity view details decisions made to allow reconfigurations of HSH systems to ensure the continuous accomplishment of their missions, decreasing as possible, manual interventions of developers, systems managers or architects. Reconfigurations can be required when the behaviour of participating services in a HSH system do not comply with quality levels expected to achieve its missions.

The strategy used in *HomecARe* was to create a *Quality manager service*, which is located in the *Quality of Service Layer* of *HomecARe*. Such layer was presented in the *Conceptual viewpoint* in Section 5.1. A *Quality manager service* is specialized to achieve requirements for each quality attribute specified in *HomecARe*. Hence, two instances of the quality managers are part of the *Quality of Service Layer*: (i) *Reliability manager*, which ensures that reliability requirements are addressed by HSH systems, and aims the identification and correction of faults, identification of possible bottlenecks, and the provision of trusted operations on patient information; and (ii) *Security manager*, which assures the accomplishment of security requirements by participating services in the HSH system, and detects if patient information is being protected against unauthorized access and if non-repudiation mechanisms are being followed by services participating of a HSH system.

In its internal structure, as presented in Figure 74, a *Quality manager service* is constructed following the MAPE-K architectural pattern. MAPE-K was introduced in Section 2.2. Initially, an *interceptor* receives *systemData* from the services interacting in the HSH. The *inteceptor* selects data from the HSB and sends *observations* of interest to a *qualityManager*. The *interceptor* is a specialization of an *Interceptor mediator*, a type of communication mediator located in the HSB layer. Interceptor is detailed in Appendix C.

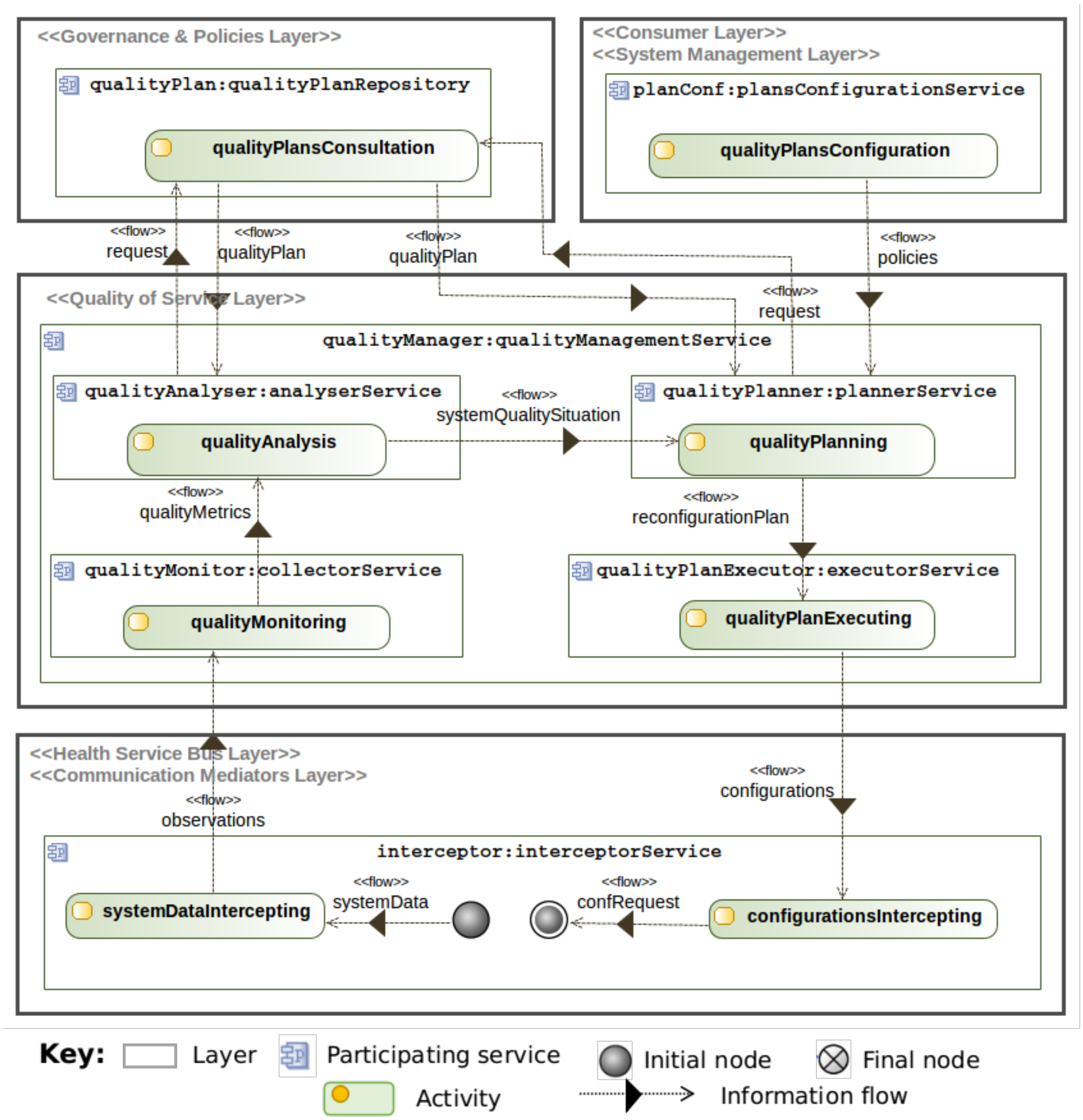


Figure 74 – Quality Manager Service Diagram in *HomecARe*.

The *qualityManager* is composed of:

- *qualityMonitor*, a specialization of a collector service (see Appendix C), which selects data from the observed entities (i.e., participating services in the HSB) and calculates quality metrics of the HSH system based on collected data; these metrics can be calculated based on the standard ISO/IEC 9126 (ISO/IEC, 2000);

- *qualityAnalyser*, a specialization of an analyser service (see Appendix C), which predicts and transfers a situation regarding the accomplishment of a quality requirement by the HSH system; for situation prediction, the analyser can consult *qualityPlans* stored in repositories located in the Governance & Policies Layer;

- *qualityPlanner*, a specialization of a planner service (see Appendix C), that, based on a quality situation sent by the *qualityAnalyser*, intends to select the best *reconfigurationPlan* to be performed by services in any layer of the HSH system architecture; for selecting this plan, the *qualityPlanner* can consult stored quality plans, or request configuration *policies* to a human manager of the HSH system through a *planConf* service, a specialization of a plans configuration service, in the Consumer Layer; and

- *qualityPlanExecutor*, a specialization of an executor service (see Appendix C), that based on the *reconfigurationPlan* sent by the *qualityPlanner*, establishes *configuration* messages, containing reconfiguration commands and target services, which are sent to the *interceptor* that is responsible to deliver each message to the specific target service.

5.4.3 QV03 - Security View

The security view describes how *HomecARe* addresses the security requirements specified in Section 4.2, i.e., SR01 and SR02. This view details mechanisms adopted in *HomecARe* to address confidentiality and integrity of patient data, non-repudiation, and authorization and authentication requirements. Such mechanisms were selected based in security tactics proposed in (BASS; CLEMENTS; KAZMAN, 2003), and are presented in the remainder of this section.

Authentication and Authorization

Figure 75 details how *HomecARe* allocates identities to constituent systems services (CSSv) when they are being integrated to the HSB. Hence, Figure 75 is an extension of the integration diagram presented in Figure 71, as part of the *Integration and interoperability view*.

Depending on assigned identities to a CSSv, such service can have rights to request, access or modify either data or other services. Hence, in *HomecARe*, identities are assigned based on roles and profiles. For instance, in the context of *Home healthcare team services*, to each member of the healthcare team a specific profile can be associated, e.g., to a physician the HSH systems can allow her / his access to respective *MD-DO services*. *Home healthcare team services* are located in the *Consumer services layer*, which was introduced in the conceptual viewpoint, Section 5.1. The allocation of identities to services in a HSH system is made by the *securityManager*, specifically, by its *qualityPlanner*. The *securityManager* is a specialization of a *qualityManager*. The structure of the *qualityManager* is given in the *Adaptivity view*, in Figure 74.

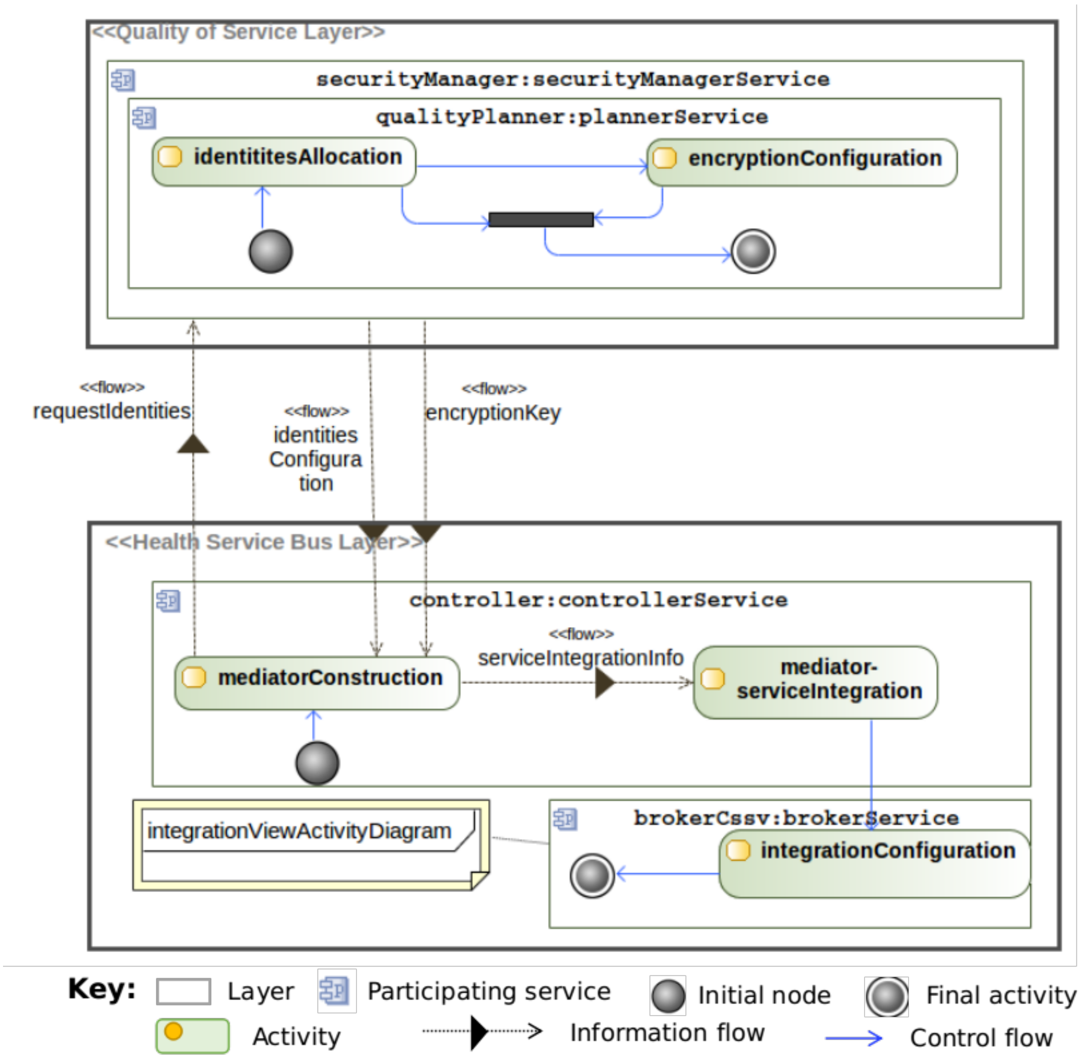


Figure 75 – Authentication diagram: assignation of service identities in *HomecARe*.

When a CSSv is publishing a message through its dedicated *broker*, the *interceptor* in the HSB authenticates permissions of the CSSv to execute actions specified in the message. For this, the *interceptor* requests permissions of the CSSv regarding the identities allocated for such service. Whether the CSSv has the rights to perform requested operations, the message is routed to related services, otherwise, the message is not routed and an *observation* of the occurrence is sent to the *securityManager* for further intrusion analysis.

Confidentiality and Integrity of Patient's Data

To address confidentiality and integrity of patient's data, end-to-end security is addressed in *HomecARe*. When a CSSv is being integrated into a HSH system, an *encryptionKey* is designated to such service's identities, as showed in Figure 75. The *encryptionKey* are constructed by the *qualityPlanner* of the *securityManager*, which is located in the *Quality of service layer*.

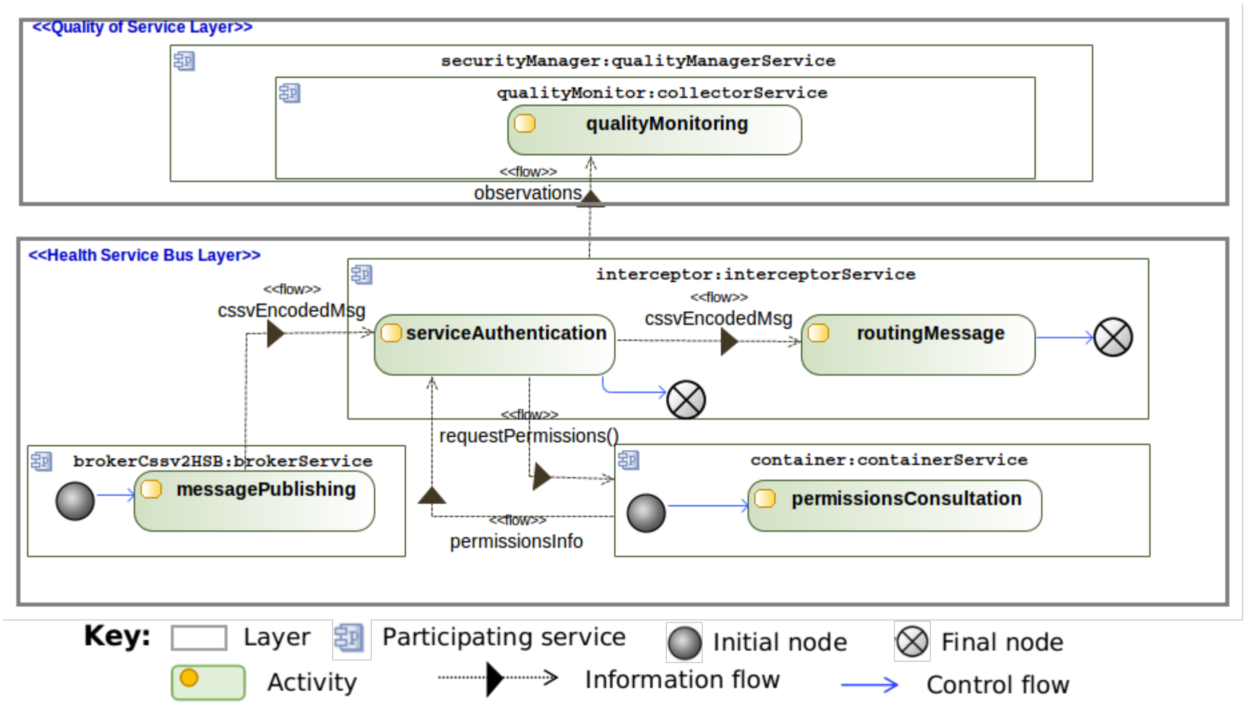


Figure 76 – Authorization diagram: verification of services permissions in *HomecARe*.

The *encryptionKey* is configured by the *brokerCssv service*, when such broker is being integrated to the CSSv in the *Integration view*, Figure 71.

Therefore, when a CSSv publishes or receives messages from the HSB through its related broker, i.e., *brokerCssv*, such broker must encrypt and decrypt patient's data following the *encryptionKey* designated to it. The encryption process is made by the *encoder service* in *brokerCssv*, whilst, the decryption process is performed by the *decoder service* in *brokerCssv*. The structure of a *broker service* was presented in Figure 72, regarding the *Integration and interoperability view*.

Regarding services outside patient's home, e.g., services in the *consumer services layer*, transport layer security protocols, i.e., SSL (Secure Sockets Layer), must be considered by such services to offer patient's data protection. Moreover, patient's data must be protected by the *Patient DL Records* repository, located in the *Information architecture layer* of *HomecARe*. For this, authentication and authorization mechanisms adopted in *HomecARe* are helpful to avoid unauthorized access to patient's data stored in the *Patient DL Records* repository.

Non-repudiation

HomecARe guarantees that the sender of a message can not later deny having sent the message, and that the recipient of a message can not deny having received the message. Hence, in *HomecARe*, all security related information is stored in the *System logs repository* located in the *Governance and policies layer*, as illustrated in Figure 77. The *interceptor*, located in the *Health service bus* records all messages sent and operations performed by services interoperating in the HSB. Registered information can be used for further auditing of services and users of the HSH system.

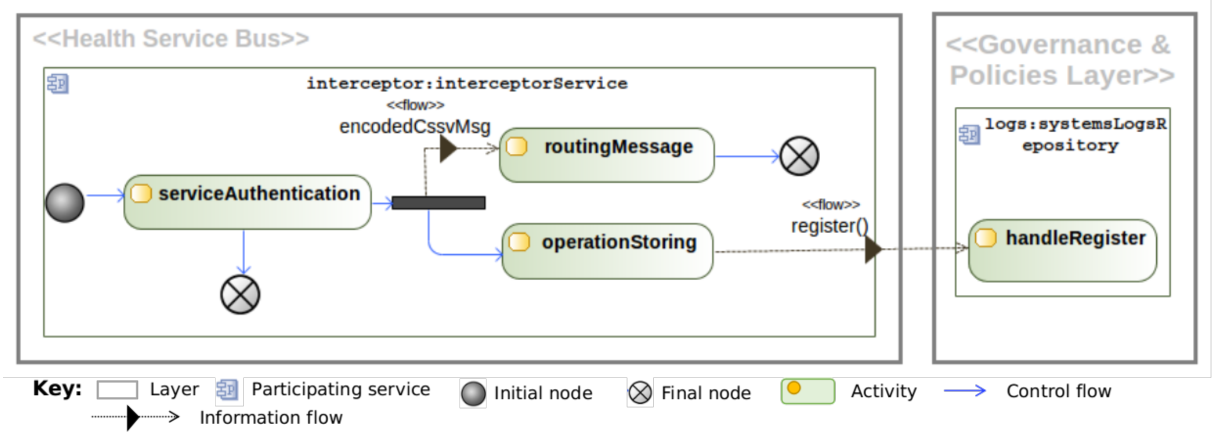


Figure 77 – Non repudiation diagram in *HomecARe*.

5.4.4 QV04 - Reliability View

This view presents tactics used in *HomecARe* to address the reliability requirements RR01, RR02, RR03, RR04, and RR05 presented in Section 4.2. This view presents approaches adopted in *HomecARe* to offer fault-tolerance, bottlenecks prevention, and trusted operations in HSH systems.

Faut-tolerance mechanisms

The *Reliability manager* is in charge of monitoring the state of the HSH system and of the participating services. The *Reliability manager* is a specialization of the *Quality manager* defined in the *Adaptivity view*, Figure 74.

The *Reliability manager* through its *qualityMonitor* detects failures or malfunctioning of services and possible congestions on networks and shared resources (e.g., repositories). The *qualityMonitor* can use tactics to detect faults, such as self-testing, ping/echo, or heartbeat, as proposed in (BASS; CLEMENTS; KAZMAN, 2003). Once a failure or malfunctioning is detected, such observation, i.e., *qualityMetric*, is sent to the *qualityAnalyser* to interpret how the HSH system reliability can be affected. A *qualitySituation* is sent from the *qualityAnalyser* to the *qualityPlanner* to select the best fault recovering mechanisms for the detected situation. When a reparation of services or parts of the HSH system are required, the *qualityPlanner* can suggest tactics, as software upgrade, degradation or reconfiguration, to be executed by the *qualityPlanExecutor*. When reintroduction of a service or parts of the HSH system are needed, the *qualityPlanner* can order the execution of some tactics, such as state resynchronization, shadow or escalating restart performed by the *qualityPlanExecutor*.

Bottlenecks prevention

To avoid bottlenecks in the HSB, dedicated brokers are allocated to each CSSv participating in the HSH system. Hence, operations with high demand of resources, such as encoding, decoding, translation, are concurrently executed, instead of being executed by a centralized entity. The broker structure was presented in the *Integration and interoperability view*, Figure 72.

Routing tables, identities, encryption keys, and other information of participating services are allocated in the *container* of the HSB, instead of being embedded into the *interceptor*. Hence, instances of *interceptor* can be created, as required, to avoid centralization of routing operations in the HSB, and each instance can have access to updated information stored in the *container*. The *interceptor* and *container* were presented in the *Integration and interoperability view*, Figure 71.

Another approach proposed in *HomecARe* to prevent bottlenecks is the creation of dedicated quality managers to ensure desired quality levels of the HSH system. Hence, instead of centralizing quality assessment to guarantee quality requirements of reliability and security in a HSH system, decentralized quality managers are proposed, i.e., *Reliability manager* and *Security manager*. The *Quality manager* was defined in the *Adaptivity view*, Figure 74.

Trusted operations

To ensure trusted operations, verifications of roles, profiles and permissions of services participating in a HSH system are performed by the *interceptor* of the HSB. Such verifications are done before routing messages sent by services. Verification of services permissions in *HomecARe* was presented in the *Security view*, Figure 76.

To guarantee trusted communication between services consumers and providers, protocols of contracts related to services interfaces define synchronous communication for exchanging critical message. Protocols, contracts, and services interfaces of participating services in *HomecARe* were detailed in the *Missions viewpoint*, Section 5.2.

5.4. Final Considerations

This chapter presented the architectural synthesis of *HomecARe*, which corresponds to the Step 3 of ProSA-RA. Three viewpoints were proposed to describe architectural solutions in *HomecARe*, namely, conceptual, missions, and quality viewpoints. Each viewpoint is composed of at least one architectural view representing how HSH systems can be configured to accomplish missions and satisfy quality attribute requirements related to interoperability, security, reliability, and adaptivity. Important decisions made in *HomecARe* were: (i) the consideration of HSH systems as a collaborative SoS; (ii) the selection of the S3 reference architecture to structure services in HSH systems; (iii) the definition of BPSvs, which are responsible to accomplish missions; (iv) the use of conversion and communication mediators to structure the HSB; (v) the adoption of brokers, and the standards SNOMED-CT and HL7 v2 by the HSB to grant semantic and syntactic interoperability; (vi) the employment of the MAPE-K pattern to structure the quality manager service and meet adaptivity requirements; (vii) the usage of tactics for authorization, authentication, data confidentiality and integrity, and non-repudiation to ensure security requirements; and (viii) the selection of strategies for fault-tolerance, bottlenecks prevention, and trusted operations, and guarantee reliability requirements in HSH systems. Architectural decisions were represented in views using BPMN, SoaML, and UML.

Architectural viewpoints, views, decisions, and domain models are evaluated in the next chapter to investigate the viability of *HomecARe* for guiding the architectural design of HSH systems.

CHAPTER

6

ARCHITECTURAL EVALUATION OF *HOMECARE*

In this thesis, *HomecARe*, a reference architecture for HSH systems was proposed. Chapters 3 to 5 presented results of conducting the three firsts steps for its development (i.e., domain analysis, architectural analysis and architectural synthesis), as proposed in the process ProSA-RA, which was presented in Figure 4, Section 2.1. The last step in the process of reference architecture engineering addresses its evaluation (NAKAGAWA; OQUENDO; MALDONADO, 2014). Hence, to assess *HomecARe*, a case study was conducted. Specifically in this thesis, an exploratory, positivist case study was defined, since it aims to search evidence to test the hypothesis that *HomecARe* is a suitable approach to address the challenges found at developing HSH systems. For this, the case study research process presented in (RUNESON; HÖST, 2008) was followed. Results of conducting such process are presented in this chapter. Specifically, Section 6.1 describes the case study design and planning, detailing objectives, hypothesis, research questions, and methods to collect data. Section 6.2 presents the collected data used to bring the required evidence to answer each research question. Section 6.3 presents the analysis and synthesis, based on collected data, to resolve the research questions, hypothesis, and objective. Discussion of the results obtained through this case study are detailed in Section 6.4. Threats of validity are discussed in Section 6.5. Finally, Section 6.6 concludes this chapter.

6.1. Case Study Design

To assess *HomecARe*, the software architecture of *DiaManT@Home* was designed as an instance of such reference architecture. *DiaManT@Home* is a HSH system that supports patients suffering with Diabetes Mellitus to manage their conditions at home (VICENTE; GARCÉS; NAKAGAWA, 2017a). *DiaManT@Home* architecture was designed following the process proposed to instantiate *HomecARe*, as presented in Appendix A. In short, such process guides the domain and architectural analysis, architectural synthesis, and the architectural assessment of a concrete HSH system architecture, through the reuse of knowledge contained in *HomecARe*. The instantiation process is based on the guidelines proposed by (HOFMEISTER *et al.*, 2005) for designing software architectures.

In this section, the case study plan is presented. Specifically, the case to be studied in this chapter is the software architecture of *DiaManT@Home* being constructed through the conduction of the instantiation process presented in Appendix A. The objective of this case study is refined in research questions (RQs) that were proposed to resolve the hypothesis stated in this thesis. Moreover, methods to collect data and bring evidence to answer the RQs are also defined in this section.

Case study objective

The main objective of this case study is to validate the viability of *HomecARe* to support the software architecture design of HSH systems capable of addressing their requirements and overcoming the challenges presented in the domain.

Research Questions (RQs)

To resolve the general objective, seven RQs were proposed. For each RQ, an hypothesis is intended to be verified or denied through the assessment of units of analysis, which are assessed using collected data. Table 14 presents the RQs and related hypothesis, units of analysis, and data to be collected during the conduction of this case study.

Table 14 – Research questions, hypothesis, units of analysis and data collected.

RQs	Hypothesis	Units of analysis	Data collected
RQ1 Can the domain knowledge, models, architectural solutions and architectural descriptions contained in *HomecARe*, improve the architectural design of HSH systems?	At using *HomecARe*, it is possible to decrease time and efforts when the software architecture of a HSH system is under designing.	Instantiation process presented in Appendix A	(a) Documents resulting of conducting the instantiation process; and (b) Analysis of time spent and people involved of conducting the process for creating the architectural design of *DiaManT@Home*.
RQ2 Can software architectures of HSH systems, designed using *HomecARe*, be considered as viable solutions?	*HomecARe* allows to design software architectures of HSH systems valid for their concrete problems.	Mapping *DiaManT@Home* architecture vs requirements	(a) Requirements document of *DiaManT@Home*; (b) Architecture models of *DiaManT@Home*; (c) Table mapping how *DiaManT@Home* architecture achieves its requirements.
RQ3 Is *HomecARe* an alternative to address interoperation issues of HSH systems?	At using *HomecARe*, an architecture of a HSH system can achieve interoperability of services provided by constituent systems.	Interoperability scenario	(a) Information from interoperability scenario template; (b) Architectural views of *HomecARe*; (c) Diagrams and models of *DiaManT@Home*.

Table 14 – *(Continuation)*

RQs	Hypothesis	Units of analysis	Data collected
RQ4 Is it possible to create software architectures of reliable HSH systems using *HomecARe*?	At using *HomecARe*, software architectures of HSH systems can address reliable operations.	Reliability scenario	Information from reliability scenario template; (b) Architectural views of *HomecARe*; (c) Diagrams and models of *DiaManT@Home*.
RQ5 Is it possible to instantiate software architectures of secure HSH systems using *HomecARe*?	At using *HomecARe*, software architectures of HSH systems can address security requirements.	Security scenario	Information from security scenario template; (b) Architectural views of *HomecARe*; (c) Diagrams and models of *DiaManT@Home*.
RQ6 Is it possible to create adaptive architectures of HSH systems using *HomecARe*?	Instances of *HomecARe* can be considered as adaptive architectures.	Adaptivity scenario	Information from adaptivity scenario template; (b) Architectural views of *HomecARe*; (c) Diagrams and models of *DiaManT@Home*.
RQ7 Does *HomecARe* present low coupling regarding patient health conditions and her/his context, e.g., country legislations, health organization policies?	*HomecARe* can be used to support architectural design of HSH systems independently from patient's political, economical, regional and health conditions.	Reusability scenarios	Information from reusability scenario template; (b) Architectural views of *HomecARe*; (c) Diagrams and models of *DiaManT@Home*.

Procedures for Data Collection

In order to obtain valid information to investigate the established units of analysis, answer the RQs, and denied or verify the pre-defined hypothesis, the following three procedures were followed.

Procedure 1 - Documenting the conduction of HomecARe instantiation process

To support the investigation of the research question RQ1, the spreadsheet available in (GARCÉS, 2017) was used. Such spreadsheet, considered as a process log, was used to document results of conducting the first three activities defined in the instantiation process in Appendix A: (i) domain analysis; (ii) architectural analysis; and (iii) architectural synthesis. Resources (time, people) required to conduct each activity were estimated. In summary, data collected and documented in such spreadsheet were:

- Requirements document containing functional and non-functional requirements of *DiaManT@Home*;
- Health information that *DiaManT@Home* must manage in its operations. Such data are defined as SNOMED CT codes;
- Services, their related health data, and functional requirements that each service is involved in. Services conforming *DiaManT@Home* are represented as instances of services defined in *HomecARe*;

- Services architectures that compose *DiaManT@Home*. Such architectures define different type of services configurations and interactions that can be realized by *DiaManT@Home*. For each services architecture, it is documented all participating services, their role (e.g., consumer, provider, subscriber or publisher), contracts used to exchange health information, and the exchanged health information (in form of SNOMED CT codes);
- Models of services architectures composing the *DiaManT@Home*. Architectures are modelled using models in SoaML defined in *HomecARe*; and
- Elements located in the information architecture, and governance and policies layers.

Procedure 2 - Documenting DiaManT@Home architecture validation

To investigate the research question RQ2, a mapping between *DiaManT@ Home* functional requirements and its architecture is made.This mapping gives evidence that all functional requirements are addressed by at least one architectural element of *DiaManT@Home*. Elements showed in Table 16 were used to document such mapping, registering the ID of the functional or non-functional requirement specified in the requirements document of *DiaManT@Home* in (VICENTE; GARCÉS; NAKAGAWA, 2017b). Following, the element (e.g., service or repository) responsible for each requirement is described, and links offering additional information of elements are given. Hence, this mapping supports the architecture evaluation activity as defined in the instantiation process of *HomecARe*.

Procedure 3 - Specifying and documenting quality scenarios

Aiming to answer the research questions RQ3, RQ4, RQ5, RQ6, and RQ7, quality scenarios specifications were proposed. Scenarios help to understand how the system behaves, and which the system's response is when an stimulus is given in determined environmental settings (BASS; CLEMENTS; KAZMAN, 2003). In this context, scenarios assist the validation of architectural decisions made to address quality attributes requirements. Therefore, this procedure is oriented to support the architectural evaluation of *DiaManT@Home* as defined in the instantiation process of *HomecARe*.

In the context of this case study, general scenarios templates as those provided in (BASS; CLEMENTS; KAZMAN, 2003; CLEMENTS; KAZMAN; KLEIN, 2002) were used to establish scenarios to assess the architecture of *DiaManT@Home* regarding interoperability, security, reliability, adaptivity, and reusability attributes.

Scenarios for quality attributes of interoperability, security, reliability, and adaptivity were considered important for this case study, since they conform the architecturally significant requirements of *HomecARe*, as presented in Section 4.2. Moreover, scenario for the quality attribute of reusability was also considered in this case study, since it represents additional challenges found in the HSH systems domain.

Therefore, for analysing how the software architecture of *DiaManT@Home* (as instance of *HomecARe*) addresses interoperability, security, reliability, adaptivity, and reusability, at least one scenario was defined for each attribute. In summary, a scenario specification is composed of eight parts as defined in (CLEMENTS; KAZMAN; KLEIN, 2002):

- *Scenario identities:* Detailing the ID number and scenario objective;
- *Attribute(s)*: Specifying the quality attribute(s) with which the scenario is concerned;
- *Environment:* Detailing relevant assumptions about the environment in which the system resides, and the relevant conditions when the scenario is carried out;
- *Stimulus:* Describing a precise statement of the quality attribute stimulus embodied by the scenario;
- *Response:* Exposing a precise statement of the designed quality attribute response. Such response should be measurable in some way to further test the quality attribute requirement;
- *Architectural decision(s):* Describing architectural decisions relevant to the scenario that affect the quality attribute requirement;
- *Reasoning:* Explaining the rationale (in a qualitative or quantitative way) behind the architectural decisions, detailing why such decisions support the achievement of quality attribute requirement; and
- *Architectural diagram:* Illustrating architectural information to support the above reasoning.

Methods for Data Analysis

Since the case study was designed as positivist, exploratory study, qualitative data analysis is used to generate the evidence for confirming or denying the established hypothesis. Hence, to answer each RQ and validate the respective hypothesis, conclusive statements were made, as proposed in (RUNESON; HÖST, 2008).

6.2. Collecting evidence

Four people participated of this case study during its conduction: (i) The software architect of *HomecARe*, in charge of verifying the correct conduction of the instantiation process of *HomecARe*, described in Appendix A, and responsible for collecting and analysing the evidence to answer the RQs; (ii) the software architect of *DiaManT@Home*, responsible for conducting and documenting the instantiation process; (iii) a system analyst responsible for supporting the requirements elicitation of *DiaManT@Home*; and (iv) a registered nurse assisting the domain analysis activity. The remainder of this section presents the information collected at conducting each procedure described in Section 6.1.

Procedure 1 - Documenting the conduction of the HomecARe instantiation process

Information collected through the conduction of the three first activities (i.e., domain analysis, architectural analysis and architectural synthesis) of the instantiation process in Appendix A are detailed as follows.

Activity 1 - Domain analysis:

In this activity, the scope of *DiaManT@Home* was established. In short, *DiaManT@Home* is a HSH system, oriented to support, at home, patients suffering of diabetes mellitus disease in the self-management of their conditions.

To identify the requirements of *DiaManT@Home*, guidelines from the Brazilian Diabetes Society (SBD, 2016), the Diabetes UK Foundation (FOUNDATION, 2016), and lectures offered by physicians and registered nurses from the University of Copenhagen & Copenhagen Business School[1]. Quality attributes requirements were obtained with the support of QM4AAL, the quality model for AAL systems (GARCÉS; OQUENDO; NAKAGAWA, 2016; GARCÉS; OQUENDO; NAKAGAWA, 2017). The requirements document for *DiaManT@Home* was refined through several iterations conducted during group meetings. In total, 73 functional requirements and 63 non-functional requirements were defined for *DiaManT@Home*. Requirements document of *DiaManT@Home* is available in (VICENTE; GARCÉS; NAKAGAWA, 2017b).

Stakeholders identified in the context of *DiaManT@Home* are the patient, family, endocrinologist, physician, nutritionist, physical educator, and nurse, which are classified as home healthcare team in *HomecARe*.

Moreover, as *DiaManT@Home* was designed in the Brazilian context, the interoperability standards that must be contemplated by the system are SNOMED CT, ICD and HL7 v2.

Resources: Three people were involved in this activity, namely a system analyst, a software engineer, and a registered nurse. In total, 368 hours were required to achieve this activity. Specifically, 320 hours were spent by the system analyst, 32 hours by the software engineer, and 16 hours by the registered nurse.

[1] Lectures were provided during the course *Business Models for Innovative Care for Older People* were used. Available in <https://www.coursera.org/learn/business-models-innovative-care/home/welcome>.

Activity 2 - Architectural analysis:

In this activity, the architecturally significant requirements (ASRs) of *DiaManT@Home* were defined. For this, non-functional requirements (NFRs), more specifically quality attributes requirements, were categorized in four priority levels: obligatory, high desired, desired, desired but not essential, and optional. Requirements categorization was made by two reviewers, the system analyst and the system engineer, who individually allocated priority levels to each of the 63 NFRs. When disagreements occurred, a consensus between both reviewers was made. As result, 52.4% (33 of 63) of NFRs were classified as obligatory to be addressed by the *DiaManT@Home* system. Moreover, 15.8% (11 of 63), 22.2% (14 of 63), 6.34% (4 of 63), and 3.17% (2 of 63) of NFRs were classified, respectively, as high desired, desired, desired but not essential, and optional. Classification of NFRs can be consulted in (VICENTE; GARCÉS; NAKAGAWA, 2017b).

The obligatory NFRs of *DiaManT@Home* were considered as types of ASRs defined in *HomecARe*, since their non-compliance by the software architecture incurs in low qualities of *DiaManT@Home*. ASRs of *HomecARe* were detailed in Section 4.2. Table 15 maps obligatory NFRs of *DiaManT@Home* and ASRs of *HomecARe*. First column lists the NFR ID of *DiaManT@Home*, as defined in the requirements document of such system (VICENTE; GARCÉS; NAKAGAWA, 2017b). Second column provides a brief description of each NFR. Third column lists the ASRs IDs of *HomecARe* where the NFR is contemplated. The last column of Table 15 describes to which quality attribute, i.e., adaptivity, security, reliability, and interoperability, considered in *HomecARe*, the NFR is associated.

Table 15 – Relationship between ASRs of *HomecARe* and NFRs of *DiaManT@Home*.

NFRs of *DiaManT@Home*		**ASRs of *HomecARe***	
ID	**Description**	**ID**	**Description**
NFR01	The system must provide adaptation in both off and run-time.	AR01 - AR08	Adaptivity
NFR02	The system must facilitate cooperation between constituents in an autonomous way.	II01	Interoperability
NFR06	The system must allow its configuration at runtime.	AR09	Adaptivity
NFR09	The system must allow easily integration of new constituents.	II01	Interoperability
NFR10	The system must provide discovery of new services and self-configuration mechanisms.	AR08, AR09	Adaptivity
NFR16	The system must recover itself from faults.	RR01, RR02	Reliability
NFR17	The system must be tolerant to constituent faults.	AR08, RR01, RR02	Adaptivity and Reliability
NFR18	The system must avoid failures propagations to constituents.	RR01, RR02	Reliability
NFR19	The system must provide errors handling.	RR01, RR02	Reliability
NFR20	The system must detect and prevent unauthorized operations.	SR01	Security
NFR21	The system must be aware of its situation, and prevent and correct internal faults and failures.	AR08	Adaptivity

Table 15 – *(Continuation)*

NFRs of *DiaManT@Home*		ASRs of *HomecARe*	
ID	**Description**	**ID**	**Description**
NFR22	The system must prevent conflicting context information from constituents.	RR01	Reliability
NFR27	The system must obtain information from heterogeneous providers.	AR01	Adaptivity
NFR30	The system must authenticate users and participating systems.	SR01, SR02	Security
NFR31	The system must offer authorization mechanisms for users and participating services.	SR01, SR02	Security
NFR32	The system must certify each constituent system individually.	SR02	Security
NFR33	Constituent systems must protect patient data, and address confidentiality and integrity of such data.	SR01	Security
NFR38	The system must offer well designed data structures to provide efficient operations over data.	None relation found	
NFR42	The system must offer reliable patient's data for diagnosis purposes.	RR05	Reliability
NFR43	The system must facilitate the communication and integration of heterogeneous constituents.	II01	Interoperability
NFR44	The system must address semantic and syntactical interoperability.	II02	Interoperability
NFR45	The system must allow integration of external systems.	II02	Interoperability
NFR51	The system must be able to track back actions on sensitive patient's information.	SR02	Security
NFR52	The system must guarantee secure operations on patient's data.	RR05, SR01, SR02	Reliability and Security
NFR53	The system must respect patient's privacy.	SR01, RR05	Reliability and Security
NFR54	The system must protect patient's data.	SR01, RR05	Reliability and Security
NFR57	The system must ensure proper executions of its operations.	RR05	Reliability
NFR58	The system must prioritize the communication of emergency related messages.	RR03, RR04	Reliability
NFR59	The system must provide reliable operations and confidentiality when patient's data are handled.	SR01	Security
NFR60	The system must offer access control mechanisms when patient's data are collected.	SR02, SR01	Security
NFR61	The system must provide trusted communication.	RR05	Reliability
NFR62	The system must verify collected patient's data are trust.	RR05	Reliability
NFR63	The system must offer trustworthy operations.	RR05	Reliability

Moreover, from the requirements document, considering functional requirements, health information was characterized following terminology offered by the standard SNOMED CT. ASRs and SNOMED CT codes were registered in the spreadsheet available in (GARCÉS, 2017). As result of the architectural analysis activity, the selection of ASRs for *DiaManT@Home*, presented in Table 15, and the characterization of health information managed in *DiaManT@Home* were made.

Resources: Two persons, one system analyst, and one software architect, participated in this activity. Classification of NFRs according with priority levels required 16 hours, i.e., 8 hours

/ person. Selection of health data and allocation of SNOMED CT codes for such data required 62 hours, i.e., 31 hours / person. In total, this activity consumed 78 hours, i.e., 39 hours / person.

Activity 3 - Architectural synthesis:

In this activity, architecture description of *DiaManT@Home* was made. Specifically, services architectures of *DiaManT@Home* were constructed based on architectural models, constructed in SoaML, that are provided in *HomecARe*.

For this, constituent systems services (CSSv) participating of *DiaManT@Home* were identified based on health information obtained in the architectural analysis (Activity 2), and that are represented as SNOMED CT codes. For each data code, an instance of a CSSv of *HomecARe* was created to provide such data. Moreover, business process services (BPSv) essential for *DiaManT@Home* were identified and instances of them were created based on required health information. Additionally, consumer services also were identified and instantiated in the context of *DiaManT@Home*. Based on requirements document, interoperation standards and healthcare plans repositories also were identified. Appendix B shows instances of *HomecARe* elements used to define the software architecture of *DiaManT@Home*. Moreover, in Appendix B responsibilities of each element of *DiaManT@Home* are detailed, linking each element to the respective functional requirement of *DiaManT@Home*.

Figure 78 shows the conceptual view of *DiaManT@Home* architecture, highlighting the instantiated elements from *HomecARe*. Descriptions of *HomecARe* services and other elements were detailed in Section 5.1. As illustrated in Figure 78, ten constituent systems services were defined for *DiaManT@Home*, being instantiated from the following six types of CSSv defined in *HomecARe* :

- *CSSv001 - physiological function assessment* instantiated into *weight assessment*, *body measures assessment* and *blood pressure assessment services*;
- *CSSv002 - cardiovascular system assessment* instantiated into the *heart beat monitor service*;
- *CSSv003 - abdomen assessment* instantiated into the *gastrointestinal assessment service*;
- *CSSv004 - nervous system assessment* instantiated into the *sleep monitor service*;
- *CSSv006 - skin ulcer assessment* instantiated into the *skin ulcer assessment service*; and
- *CSSv012 - patient environment manager* instantiated into the *domotic service*;

Regarding business processes services (BPSv), five BPSv were instantiated for *DiaManT@Home* by its software architect: (i) the *physical exercise manager* as instance of *BPSv002-*

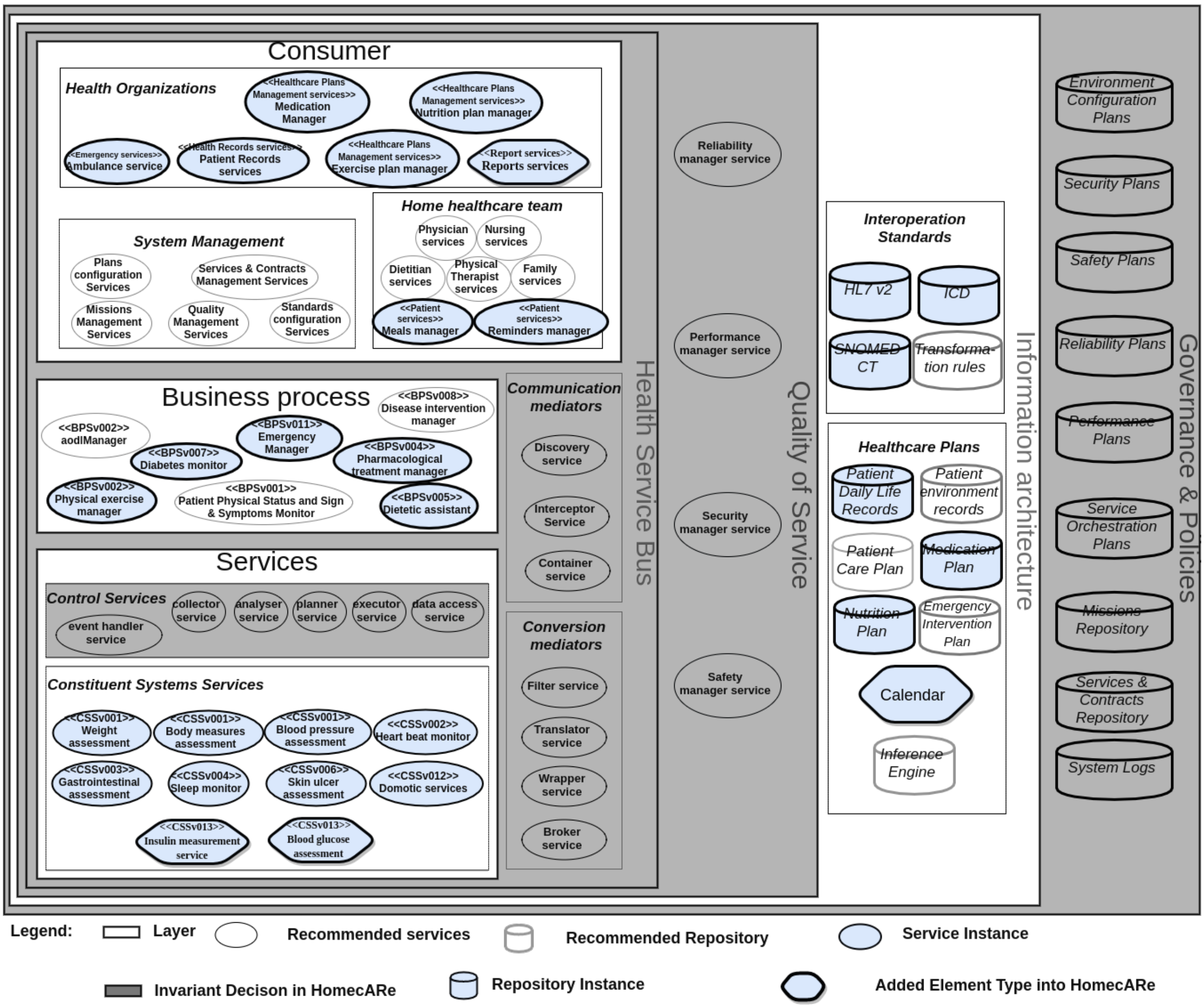


Figure 78 – Layered architecture of *DiaManT@Home* as instance of *HomecARe*

Activities of daily life manager; (ii) the *diabetes monitor* as instance of *BPSv007 - Health condition monitor*; (iii) the *dietetic assistant* as instance of *BPSv005 - Dietetic assistant*; (iv) the *pharmacological treatment manager* as instance of *BSPv004 - Pharmacological treatment manager*; and the *emergency manager* as instance of *BSPv011 - Emergency manager*.

Considering services in the consumer layer, six services related to health organizations were proposed in the architecture of *DiaManT@Home*. Three of them, i.e., *medication manager*, *nutrition plan manager* and *exercise plan manager*, are instances of the *Healthcare plans management services* proposed in *HomecARe*. Instances of *Emergency services*, i.e., ambulance services, and *Health records services*, i.e., patient records services, also were defined for *DiaManT@Home*. Concerning services for the home healthcare team, in *DiaManT@Home*, were identified two instances of *Patient services*, named, *meals manager* and *reminders manager* services.

Relating to elements in the information architecture layer, three interoperation standards were defined to be used in *DiaManT@Home*, i.e., *HL7 v2*, *ICD* and *SNOMED CT*. Moreover, important healthcare plans to be considered in *DiaManT@Home* are *medication plan* and *nutrition plan*. Furthermore, a repository representing *Patient daily life records* was also taken into consideration for such system.

Once the software architecture of *DiaManT@Home* was designed by its software architect, a reviewing meeting with the architect of *HomecARe* was performed to obtain the most feedback as possible at using *HomecARe* to create such concrete HSH system. In this perspective, during the reviewing meeting, it was identified the need for adding, into *HomecARe*, a new type of CSSv called *CSSv013 - body substance assessment*. Hence, in *DiaManT@Home*, the CSSv013 was instantiated into the *insuline measurement service* and *blood glucose assessment service*. Moreover, a new type of service for health organizations, named *Reports service*, was identified as important, hence, it was included into *HomecARe*. An additional repository, named *Calendar*, was also identified as required for *DiaManT@Home*, and since it was not contemplated initially in *HomecARe*, it was added into the reference architecture. Furthermore, the architect of *HomecARe* recommended the architect of *DiaManT@Home* to include additional services located into the business process and consumer layers. Finally, repositories and engines not contemplated initially by the architect of *DiaManT@Home*, were recommended to be included in the architecture of such system.

The remainder of services and repositories located in layers of control services, health service bus, quality of service, and governance and policies, were contemplated into *DiaManT@Home*, since these elements of *HomecARe* give support to achieve quality attributes requirements in both *DiaManT@Home* and *HomecARe*. In this perspective, such elements of *HomecARe* could offer an invariant platform for HSH systems, hence, the software architect of *DiaManT@Home* can be worried about create instances for those elements.

Services Architectures of *DiaManT@Home*

As result of the architectural synthesis, the software architecture of *DiaManT@Home* was modelled using SoaML models of *HomecARe*. Diagrams depicted services architectures composing *DiaManT@Home* are presented in Figures 79 to 83. Such services architectures were created based on the BPSvs required in *DiaManT@Home* and depicted in Figure 78.

A services architecture represents the internal structure of specific BPSvs. BPSvs use several contracts to communicate information between them. Contracts in green colour represent contracts used by a BSPv to publish an information into the HSB to be used by other BSPvs. Contracts in blue colour denote such contracts used by BPSv to recover information published by other BPSvs from the HSB.

Services architecture for the *BPSv001-1 Patient Physical Status and Sign-Symptoms Monitor* is presented in Figure 79. The *BPSv001-1* collects physical measures and sign &

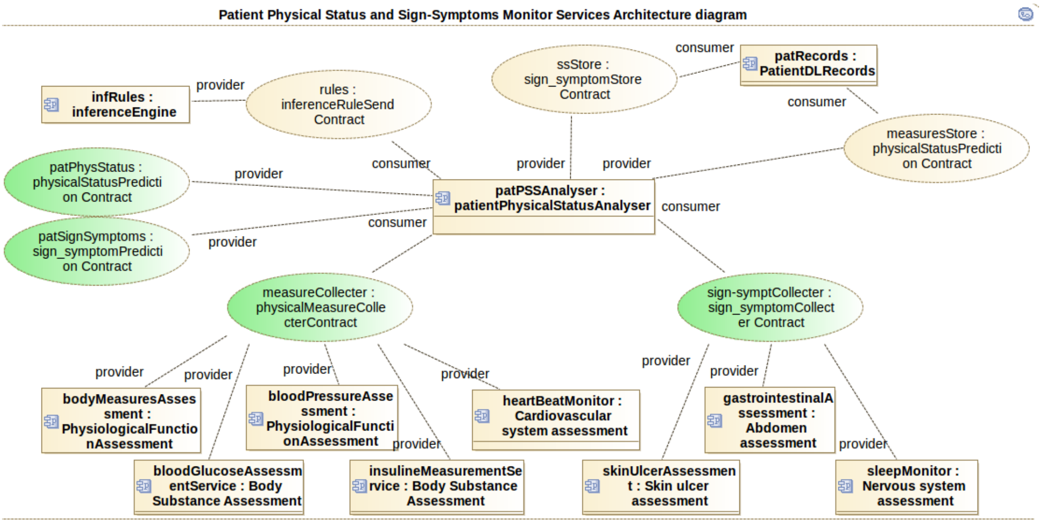


Figure 79 – Services architecture for the *BPSv001-1 Patient Physical Status and Sign-Symptoms Monitor* of *DiaManT@Home*.

symptoms from the HSB. Physical measures and sign& symptoms are published by constituent system services into the HSB. Moreover, the *BPSv001-1*, through its *patPSSAnalyser*, establishes the patient's physical status and additional patient's sign and symptoms through the use of patient records, obtained from *patRecords*, and inference rules, recovered from *infRules*. Both information,physical measures and sign & symptoms are published by the *patPSSAnalyser* into the HSB.

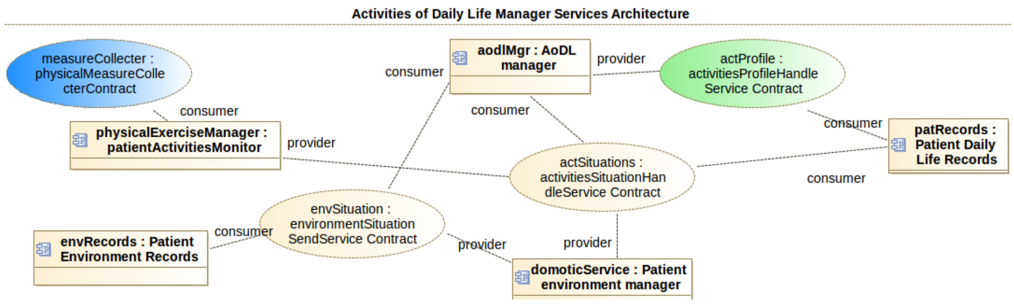


Figure 80 – Services architecture for the *BPSv002-2 AoDL Manager* of *DiaManT@Home*.

Services architecture for the *BPSv002-2 AoDL Manager* is presented in 80. The *AoDL Manager* establishes the patient's AoDL profile that contains information about the status of patient's house (e.g., informing the status of doors, windows, gas, water) and the activities that the patient performed at home (e.g., cleaning, douching). Moreover, the *AoDL Manager* uses information about the patient's exercise situation (e.g., frequency of physical exercises) provided by the *BPSv002-1 physicalExerciseManager*. The *physicalExerciseManager* collects information about patient's physical measures from the HSB to establish patient's physical exercise situation, that is used by the *AoDL Manager* to establish the patient's activities profile.

Services architecture for the *BPSv008-1 Diabetes Intervention Manager* is depicted in Figure 81. The *Diabetes Intervention Manager* establishes the patient's intervention pro-

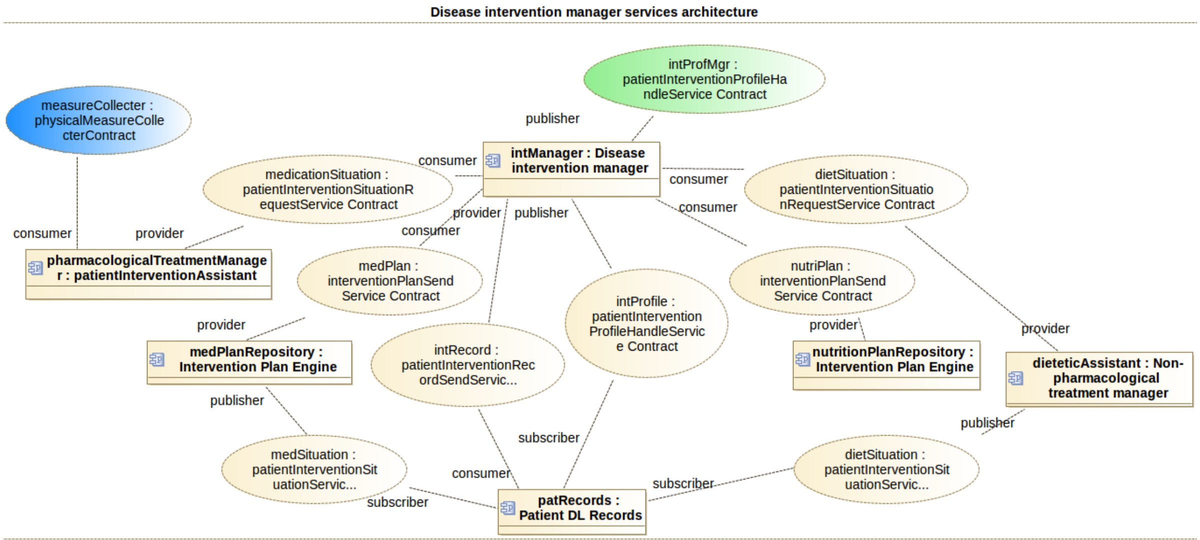


Figure 81 – Services architecture for the *BPSv008-1 Diabetes Intervention Manager* of *DiaManT@Home*.

file, containing information about the pharmacological treatment situation, which is provided by the *pharmacologicalTreatmentManager*, and diet situation, which is established by the *dieteticAssistant*. To establish the pharmacological treatment situation of the patient, the *pharmacologicalTreatmentManager* collects patient's physical measures, such as blood glucose and insulin levels from the HSB.

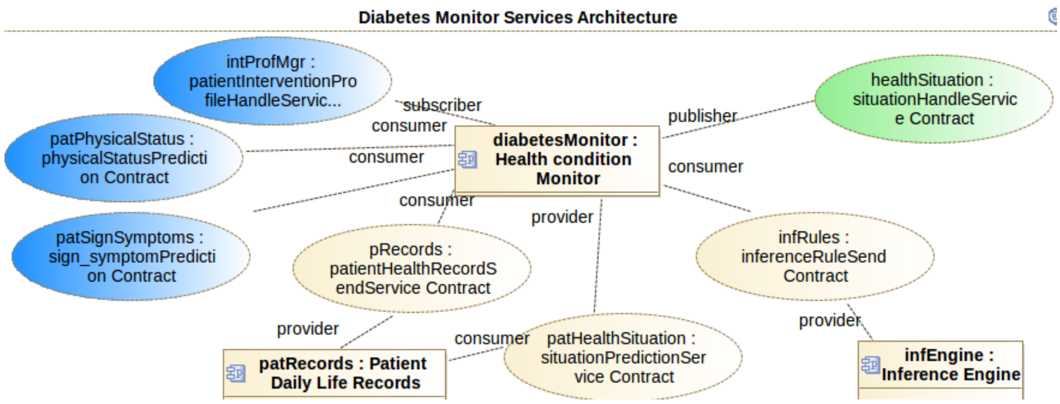


Figure 82 – Services architecture for the *BPSv007-1 Diabetes Monitor* of *DiaManT@Home*.

Services architecture for the *BPSv007-1 Diabetes Monitor* is presented in Figure 82, that is in charge of establish the patient's health situation. For this, the *Diabetes Monitor* obtains from the HSB the following information: (i) information about patient's physical status and sign & symptoms, that is provided by the *BPSv001-1 Patient Physical Status and Sign-Symptoms Monitor*, as showed in Figure 79; and (ii) information about patient's intervention profile defined by the *BPSv008-1 Diabetes Intervention Manager*, as illustrated in Figure 81. The patient's health situation is published by the *Diabetes Monitor* into the HSB.

Services architecture for the *BPSv0011-1 Emergency Manager* is depicted in Figure 83. The the *Emergency Manager* defines actions that must be executed when an emergency situation

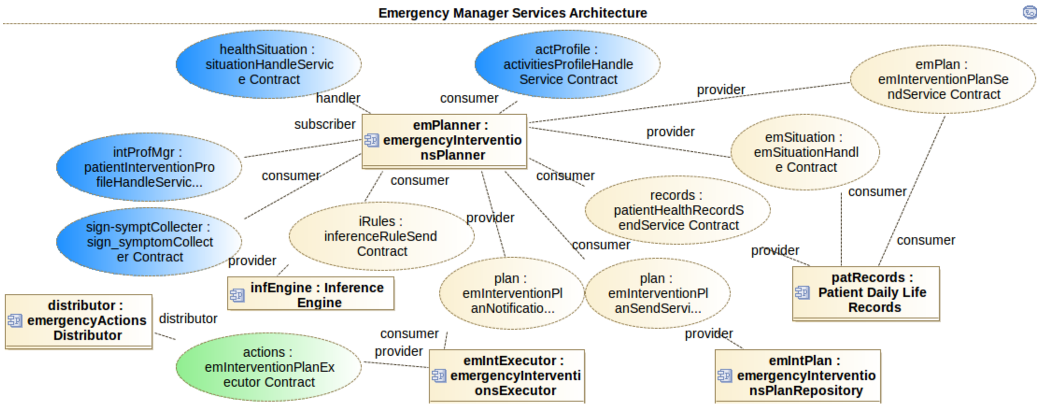


Figure 83 – Services architecture for the *BPSv0011-1 Emergency Manager* of *DiaManT@Home*.

is detected. For this, the *Emergency Manager* is composed of an emergency intervention planner, *emPlanner*, which collects from the HSB information about patient's health situation, intervention profile and signs & symptoms, and detects or predicts emergency situations regarding patient's health, e.g., patient presents symptoms of hypoglycaemia. When the emergency situation is detected, the *emPlanner* establishes an intervention plan to support the patient to obtain the adequate support by the home healthcare team or healthcare services. Moreover, the *Emergency Manager* is composed of an emergency intervention executor, *emIntExecutor*, that is in charge of interpret the intervention plan and establish which activities must be done by which entities, and thus, to send the respective notifications to interested parts.

Resources: To conduct the architectural synthesis of *DiaManT@Home* two persons were required: the software architect of *DiaManT@Home* and the architect of *HomecARe*. Most of the work made in this activity was under responsibility of the architect of *DiaManT@Home*. Such architect spent a total of 82 hours to understand and instantiate *HomecARe* into *DiaManT@Home*. The SoaML project created to design *HomecARe* was used by the architect as basis to create models for *DiaManT@Home*. The architect of *HomecARe* spent 26 hours, in this activity, to resolve doubts and in the reviewing meeting made jointly with the software architect of *DiaManT@Home*. Considering time spent by both architects, this activity demanded 108 hours to be completed.

Procedure 2 - Documenting DiaManT@Home architecture validation

This procedure is related with the architectural evaluation activity proposed in the instantiation process in Appendix A. Specifically, in this procedure the architecture of *DiaManT@Home* is validated regarding its capacity to address its functional requirements that are specified in (VICENTE; GARCÉS; NAKAGAWA, 2017b). In this context, Table 16 presents a mapping between *DiaManT@Home* requirements and its architecture. This mapping offers evidence of which functional requirements (FR) are addressed by which architectural elements (e.g, CSSv, BPSv, consumer services or repositories). Elements contained in the architecture of

DiaManT@Home are presented in the layered view in Figure 78. The purpose of such mapping is to give evidence that no functional requirement has been omitted in the architecture of *DiaManT@Home*, or in other words, to show that all functional requirements has been considered at least by one element of such architecture.

In the first column of Table 16 is documented the ID of the functional requirement as specified in the requirements document of *DiaManT@Home* in (VICENTE; GARCÉS; NAKAGAWA, 2017b). Following, the architectural element that addresses the requirement is named, and a link is offered for additional information (e.g., diagrams or textual information) about how the element participates in the software architecture of *DiaManT@Home*.

Table 16 – Mapping functional requirements and architectural description of *DiaManT@Home*.

Requirement ID	**Architectural element**	**Link**
FR01	Physical exercise manager	Service architecture in Figure 80.
FR02	Exercise plan manager	Layered architecture in Figure 78
FR03	Exercise plan manager	Layered architecture in Figure 78
FR04	Nutrition plan manager	Layered architecture in Figure 78
FR05	Dietetic assistant; Meals manager	Service architecture in Figure 81.
FR06	Nutrition plan manager; Nutrition plan repository	Layered architecture in Figure 78
FR07	Nutrition plan manager; Nutrition plan repository	Layered architecture in Figure 78
FR08	Nutrition plan manager; Nutrition plan repository	Layered architecture in Figure 78
FR09	Dietetic assistant	Service architecture in Figure 81.
FR10	Dietetic assistant	Service architecture in Figure 81.
FR11	Dietetic assistant; Nutrition plan repository	Service architecture in Figure 81.
FR12	Dietetic assistant; Nutrition plan repository	Service architecture in Figure 81.
FR13	Meals manager; Nutrition plan repository	Layered architecture in Figure 78
FR14	Meals manager	Layered architecture in Figure 78
FR15	Medication manager; Medication plan repository	Layered architecture in Figure 78
FR16	Medication manager	Layered architecture in Figure 78
FR17	Medication manager	Layered architecture in Figure 78
FR18	Medication manager; Medication plan repository	Layered architecture in Figure 78
FR19	Pharmacological treatment manager; Patient DL Records	Service architecture in Figure 81.
FR20	Diabetes monitor; Patient DL Records	Service architecture in Figure 82.
FR21	Insulin measurement service, Diabetes monitor	Services architecture in Figure 79
FR22	Diabetes monitor	Service architecture in Figure 82.
FR23	Diabetes monitor	Service architecture in Figure 82.
FR24	Blood glucose assessment service; Patient DL Records	Services architecture in Figure 79
FR25	Blood glucose assessment service	Services architecture in Figure 79
FR26	Weight assessment	Services architecture in Figure 79
FR27	Body measures assessment	Services architecture in Figure 79
FR28	Body measures assessment	Services architecture in Figure 79
FR29	Body measures assessment	Services architecture in Figure 79
FR30	Patient records services; Patient DL Records	Layered architecture in Figure 78
FR31	Patient records services; Patient DL Records	Layered architecture in Figure 78
FR32	Patient records services; Patient DL Records	Layered architecture in Figure 78
FR33	Dietetic assistant	Service architecture in Figure 81.
FR34	Domotic services	Service architecture in Figure 81.
FR35	gastrointestinal assessment	Services architecture in Figure 79

Table 16 – *(Continuation)*

Requirement ID	**Architectural element**	**Link**
FR36	Skin ulcer assessment	Services architecture in Figure 79
FR37	Blood pressure assessment	Services architecture in Figure 79
FR38	Heart beat monitor	Services architecture in Figure 79
FR39	Sleep monitor	Services architecture in Figure 79
FR40	Reminder manager; Calendar	Layered architecture in Figure 78
FR41	Reminder manager; Calendar	Layered architecture in Figure 78
FR42	Reminder manager; Calendar	Layered architecture in Figure 78
FR43	Reminder manager; Calendar	Layered architecture in Figure 78
FR44	Reminder manager; Calendar	Layered architecture in Figure 78
FR45	Reminder manager; Calendar	Layered architecture in Figure 78
FR46	Reminder manager; Calendar	Layered architecture in Figure 78
FR47	Reminder manager; Calendar	Layered architecture in Figure 78
FR48	Patient DL Records	Services architectures in Figures 80, 81, 82, 83
FR49	Physical exercise manager; Emergency manager	Service architectures in Figures 80 and 83
FR50	Physical exercise manager; Emergency manager	Service architectures in Figures 80 and 83
FR51	Physical exercise manager; Emergency manager	Service architectures in Figures 80 and 83
FR52	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR53	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR54	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR55	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR56	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR57	Dietetic assistant; Emergency manager	Service architectures in Figures 80 and 83
FR58	Pharmacological treatment manager; Emergency manager	Service architectures in Figures 80 and 83
FR59	Pharmacological treatment manager; Emergency manager	Service architectures in Figures 80 and 83
FR60	Diabetes monitor; Emergency manager	Service architectures in Figures 82 and 83.
FR61	Diabetes monitor; Emergency manager	Service architectures in Figures 82 and 83.
FR62	Diabetes monitor; Emergency manager	Service architectures in Figures 82 and 83.
FR63	Diabetes monitor; Emergency manager	Service architectures in Figures 82 and 83.
FR64	Emergency manager	Service architecture in Figure 83.
FR65	Domotic services; Emergency manager	Service architecture in Figure 83.
FR66	gastrointestinal assessment; Emergency manager	Services architectures in Figures 79 and 83
FR67	Skin ulcer assessment; Emergency manager	Services architectures in Figures 79 and 83
FR68	Blood pressure assessment; Emergency manager	Services architectures in Figures 79 and 83
FR69	Heart beat monitor; Emergency manager	Services architectures in Figures 79 and 83
FR69b	Sleep monitor; Emergency manager	Services architectures in Figures 79 and 83
FR70	Report services; Patient DL Records	Layered architecture in Figure 78
FR71	Report services; Nutrition plan repository	Layered architecture in Figure 78
FR72	Report services; Medication plan repository	Layered architecture in Figure 78
FR73	Report services; Patient DL Records	Layered architecture in Figure 78

Resources: To analyse which functional requirements are under responsibility of which architectural elements, the architect of *DiaManT@Home* spent 8 hours. To assist such activity, information about the *HomecARe* instantiation, presented in Appendix B, was used.

Procedure 3 - Specifying and documenting quality scenarios

This procedure is also oriented to support the activity of architectural evaluation of *DiaManT@Home*. Specifically, this procedure gives evidence to assess architectural decisions regarding quality attributes requirements. In this context, nine quality scenarios were used to validate the software architecture of *DiaManT@Home* regarding quality attributes of interoperability, security, reliability, adaptivity and reusability. For each quality attribute at least one scenario was proposed following the guidelines offered in (CLEMENTS; KAZMAN; KLEIN, 2002).The nine quality scenarios are presented as follows.

Scenario 1 - Interoperability scenario

- *Attribute(s)*: **Technical, Semantic and Syntactic Interoperability**
- *Environment:* The sender and destination services are previously known by the interceptor. The health vocabulary used by sender service is based on ICD, while the vocabulary used by the destination service is based on SNOMED CT. The sender service uses its own message format based on XML, whilst the destination service utilizes HL7 v2 format.
- *Stimulus:* A constituent system service (CSSv) publishes a message in the HSB.
- *Response:* The message is delivered and interpreted to interested parts.
- *Architectural decision(s):*
    - Formatting exchanged information as messages to ensure technical interoperability among services;
    - Using well defined standards for message formatting and health related terminology to address semantic and syntactic interoperability.
    - It is used a broker that provides bridging logic that carries out the message conversion between a CSSv and the HSB. The broker is connected to the HSB, that is in charge of routing the messages.
    - Broker is structured following a pipe and filter style, where each operation inside the broker, i.e., collecting, encoding, decoding, and translating, is considered a filter.
    - A dedicated broker collects information from a CSSv, decodes the XML-based message, filters the message segments, selects the terms in ICD vocabulary from the segments, traduces terms in ICD vocabulary into terms in SNOMED CT, encodes terms into HL7-based messages using segments types, and publishes the encoded message into the HSB for further routing to destination services. Decoding and encoding operations are made using transformation rules between vocabularies and message formats.

- Vocabularies, message formats, and transformation rules are stored in repositories to ensure persistence and facilitate reusability, configurations and modifications of interoperation standards.

- *Reasoning:*
  - **Benefits:** (i) High maintainability of interoperation standards and transformation rules; (ii) Low coupling between CSSv and HSB, facilitating modifications of CSSv and decreasing impacts on the HSB; (iii) High semantic and syntactic interoperation between CSSv; (iv) Reusability of vocabularies, message formats, and transformation rules.
  - **Liabilities:** (i) Significant performance overhead can be imposed by brokers; (ii) interoperation standards must be well structured to allow transformations; (iii) transformation rules must be created when a CSSv does not follow a specific interoperability standard, i.e., to map segments of the XML-based messages to segments of HL7-based messages.

- *Architectural diagram:* Broker diagram in Figure 72, corresponding to the Integration and interoperability view of *HomecARe* presented in Section 5.3.

Scenario 2 - Interoperability scenario

- *Attribute(s)*: **Business Process Interoperability**

- *Environment: DiaManT@Home* addressing the missions: *GM1.1.B - Remote physical examination performed*, and *GM1.1.1.A - Sign and symptoms monitored remotely.*

- *Stimulus:* Physical measurements, e.g., blood glucose, hearth beats or insulin levels, are published into the HSB of *DiaManT@Home*.

- *Response:* The business process service *BPSv001-1 Patient Physical Status and Sign-Symptoms Monitor* establishes and communicates the patient's physical status, *patPhysStatus*, and sign and symptoms, *patSignsSymptoms*.

- *Architectural decision(s):* To design BPSv001-1 in *DiaManT@Home*, the following decisions were instantiated from *HomecARe*:
  - A business process model, which follows clinical guidelines, defines the participants, activities and information that must be coordinated to establish the patient's physical status and sign and symptoms.
  - Based on the business process model, capabilities, services interfaces, contracts and protocols were established to certify the well execution of the business process.

- The patient's physical status, *patPhysStatus*, and sign and symptoms, *patSignsSymptoms* communicated by the BPSv001-1, are delivered by the HSB to other BPSvs which have under their responsibility the well execution of additional business processes that *DiaManT@Home* must execute, e.g., *BPSv007-1 Diabetes condition monitor* .

- *Reasoning:*

  - **Benefits:** (i) The use of SOA approach allows to achieve business interoperability, since it is possible to choreograph and orchestrate services using the contracts and protocols of their interfaces; (ii) low coupling of services offering the required capabilities to perform the business processes, due that any service providing such capability is allowed to participate of the system; (iii) high flexibility and maintainability, since services can be replaced or modified in runtime without affecting the overall behaviour of the HSH system.
  - **Liabilities:** (i) security can be affected, since no all services participating of the HSH system are previously known, meaning that before being allowed to initiate their operations in the system, mechanisms to ensure the reliability of their operations must be executed.

- *Architectural diagrams:* Business process, capabilities, services interfaces, contracts and protocols diagrams, presented in the mission viewpoint *MV1- Physical Status and Signs & Symptoms Monitored Remotely*, in Section 5.2.

Scenario 3 - Interoperability scenario

- *Attribute(s)*: **Adaptivity, Integration**
- *Environment:* Normal operation of *DiaManT@Home*.
- *Stimulus:* Physician updates the patient care plan, requiring to incorporate extra services to monitor new conditions of the patient.
- *Response:* One of the following: (i) New services discovered and integrated, and reconfigurations of the HSB done; (ii) No service discovered, or (ii) No service integrated.
- *Architectural decision(s):*

  - The discovery service is responsible for discovering available services to be integrated into the HSH system.
  - The discovery service performs service verifications regarding its identities, message format, vocabulary type, services interfaces, capabilities, contracts and qualities.

- After correct verification, the service is integrated, otherwise, no integration is done.
- A broker is allocated, by the controller, to offer interoperable communication between the new service and the HSB.

- *Reasoning:*

  - **Benefits:** Automatic discovering and integration of new services, reducing manual configuration of brokers and interfaces.
  - **Liabilities:** (i) If none service offering desired capabilities is found, functionalities required to support new patient's conditions can not be offered; (ii) Only services using message formats and vocabularies stored in repositories in the interoperability standards layer can be integrated, otherwise, the new service must offer transformation rules to be executed by the broker.

- *Architectural diagram:* Diagram of integration of new services in *HomecARe*, showed in Figure 71, corresponding to the integration and interoperability view presented in Section 5.3.

Scenario 4 - Interoperability scenario

- *Attribute(s)*: **Integration and Interoperability**
- *Environment:* Normal operation of the HSH system
- *Stimulus:* A consumer layer service desires to consult patient records from the HSH system.
- *Response:* One of the following: (i) service verified and patient records sent to the service; (ii) service can not be verified and no records sent; or (iii) service interface is customized and records sent.
- *Architectural decision(s):*

  - The consumer service sends a presentation message containing its credentials and interfaces specifications. Service verification is made by the discovery service located into the HSB. Whether credentials and interfaces are in conformity, the health record service is integrated, and if required, a dedicated broker is allocated to serve as a bridge between such service and other participating elements in the HSH system.
  - When verification of the consumer service can not be done, integration of such service to the HSH system is denied;

  - When services interfaces of the consumer service are not in compliance with those interfaces specified in the governance and policies layer, the HSH system manager can support the creation of new services interfaces to allow the consumer service the access to the requested information.

- *Reasoning:*

  - **Benefits:** Allows the evolution of HSH system, since new demands, e.g., information requests, made by consumers can be handled.
  - **Liabilities:** Reusability of interfaces can be affected, due to new interfaces will be considered as fitted interfaces that are dedicated for responding only to the consumer service.

- *Architectural diagram:* Integration diagram in Figure 71,and Broker diagram in Figure 72. Both diagrams correspond to the Integration and interoperability view of *HomecARe* presented in Section 5.3.

Scenario 5 - Reliability scenario

- *Attribute(s)*: **Reliability**
- *Environment:* Normal operation of *DiaManT@Home*.
- *Stimulus:* An emergency situation was detected by a BPSv.
- *Response:* Emergency message is routed by the HSB to the emergency manager in less than 1 second. The emergency plan is defined by the emergency manager and distributed to interested parts in less than 30 seconds.
- *Architectural decision(s):* It is used the publish / subscriber pattern to communicate emergency messages in the HSB. The emergency manager is subscribed to any message containing emergency situations information. A message for specific emergency situation is identified by the container within the HSB and it is routed to the emergency manager. Emergency messages formats are in conformity with the standard HL7 v2, containing SNOMED CT codes, communicating the finding of a risk, e.g., the SNOMED CT code 302866003 is used to communicate an event of hypoglycaemia. Different types of emergency codes in *DiaManT@Home* are detailed in Appendix B. According to the emergency situation, an emergency plan is proposed by the emergency planner that is part of the emergency manager. Emergency situations can be related with abnormal patient's health status, problems in the environment, or issues related with medication intake. Emergency plans for all possible emergency situations are stored in the emergency intervention plans repository. The emergency plan is processed by the emergency intervention executor that

establishes specific message types to be send to the respective emergency services, e.g., ambulance, hospitals, or members of the healthcare team.

- *Reasoning:*
    - **Benefits:** (i) performance improvement at communicating only emergency messages to the emergency manager, avoiding thus, unnecessary pre-processing of messages by the emergency manager; (ii) At using standardized message format, the emergency manager can easily identify the type of emergency situation, and hence, to establish the adequate emergency plan; (iii) modifiability improvement due to low coupling between publishers of emergency messages and the emergency manager; (iv) dynamic scalability improvement, due that new types of emergency situations can be detected and managed, since emergency plans can be added to the repository, and the emergency manager can be subscribed to new topics (or SNOMED CT codes).
    - **Liabilities:** (i) Increasing of design time, since emergency plans must agree with clinical guidelines, that in turns, depend from country legislations. Hence, for each emergency situation, e.g., patient presenting symptoms related with hypoglycaemia, specific actions to be performed by specific stakeholders must be defined in a way that can be computed by the emergency manager; (ii) decreasing of performance, since to ensure reliable emergency situations, for each emergency message published in the HSB, publisher's credentials must be checked to authorize the delivery of such message to the emergency manager.
- *Architectural diagram:* Services architecture for the *BPSv011-1 Emergency Manager* of *DiaManT@Home* in Figure 83 presented in Section 6.2.

Scenario 6 - Security scenarios

- *Attribute(s)*: **Security**;
- *Environment:* Normal operation of *DiaManT@Home*.
- *Stimulus:* Unauthorized attempt is made to access, modify or delete patient's data stored in the patients records repository. The attempt of access is made by a service that was neither previously verified nor integrated in the HSB.
- *Response:* Access to patient's information is denied and the attack is recorded to ensure that responsible can not repudiate its participation in the attack.
- *Architectural decision(s):* When a service is being integrated into the HSB, identities are allocated to such service. Identities contain profiles and encryption keys specific for such service and are registered in the container of the HSB. A service can only interact

with data repositories through the HSB, and can realize operations according with its identities registered in the container. In this context, if a non integrated service wants access, modify or delete any information of any repository, the interceptor does not authorize its operations and records the attack into the systems logs, for auditing purposes. Moreover, the interceptor sends the observation to the security manager, who is in charge to notify the attack to the respective authorities.

- *Reasoning:*
  - **Benefits:** (i) Improvement of detection of unauthorized access to patient's data; (ii) Provision of mechanisms for auditing operations and entities involved in such operations, and thus, to avoid possible denial of their participation.
  - **Liabilities:** Negative impact on performance, since for each message published in the HSB, publisher's credentials must be checked to authorize the publication of their messages.
- *Architectural diagram:* Three diagrams, corresponding to the security view of *HomecARe*, are associated to this scenario: (i) Assignation of service identities in *HomecARe*, presented in Figure 75; (ii) Verification of services permissions in *HomecARe*, depicted in Figure 76; and (iii) Non repudiation in *HomecARe*, showed in Figure 77. Moreover, the quality manager diagram is also related to this scenario. The quality manager is a generalization of the security manager, and it is showed in Figure 74, presented in the adaptivity view of *HomecARe*, in Section 5.3.

Scenario 7 - Security scenarios

- *Attribute(s)*: **Security, Authorization and Non-repudiation**;
- *Environment:* A message containing confidential patient's information is published in the HSB.
- *Stimulus:* An integrated service in the HSB attempts to obtain patient's information contained in a message for which such service have not required permissions.
- *Response:* Messages are only delivered to previously identified and authorized services. The attack is recorded to ensure that responsible can not repudiate its participation in the attack.
- *Architectural decision(s):* For each message published in the HSB by a service, service's credentials are authenticated by the interceptor based on service's profile registered in the container. When the service is authorized to realize operations contained in the message, such message is routed to interested parts, otherwise, the message is not routed. The interceptor registers the operation in the system logs repository, for auditing purposes.

- *Reasoning:*
  - **Benefits:** (i) Improvement of detection of unauthorized access to patient's data; (ii) Provision of mechanisms for auditing operations and entities involved in such operations, and thus, to avoid possible denial of their participation.
  - **Liabilities:** Negative impact on performance, since for each emergency message published in the HSB, publisher's credentials must be checked to authorize the publication of their messages.
- *Architectural diagram:* Three diagrams, corresponding to the security view of *HomecARe*, are associated to this scenario: (i) Assignation of service identities in *HomecARe*, presented in Figure 75; (ii) Verification of services permissions in *HomecARe*, depicted in Figure 76; and (iii) Non repudiation in *HomecARe*, showed in Figure 77.

Scenario 8 - Adaptivity scenario

- *Attribute(s)*: **Reliability and Adaptivity**
- *Environment:* Normal operation of *DiaManT@Home*.
- *Stimulus:* A fault in a service was detected by the reliability manager.
- *Response:* (i) To log the fault; (ii) to follow the quality plan; (iii) recover from the fault.
- *Architectural decision(s):* It was used the MAPE-K pattern to design the reliability manager, i.e., an instance of the quality manager in *HomecARe*. The reliability manager realizes a reflection about the current status of services that are interacting through the HSB. The interceptor in the HSB sends observations about services availability to the collector of the reliability manager. Based on those observations an analyser establishes the possibility of a fault, and communicates the situation to the planner that establishes a recovery plan to be executed by services and the HSB, e.g., removing, changing or restarting the service.
- *Reasoning:*
  - **Benefits:** (i) Offering an holistic view about the HSH system status, regarding the availability of the participating services; (ii) Improvement of fault preventions, since faults can be predicted before its occurrence.
  - **Liabilities:** (i) Negative impact on performance, since the establishment of current HSH system situation requires information processing demanding additional time; (ii) The successful fault detection, prevention or recovery, depends from the quality of the observations and prediction models used to establish the system's situations and recovery plans.

- *Architectural diagram:* Quality manager service in *HomecARe*, presented in Figure 74 comprised in the adaptivity view in Section 5.3.

Scenario 9 - Reusability scenario

- *Attribute(s)*: **Reusability**
- *Environment:* Initial configuration of *DiaManT@Home*.
- *Stimulus: DiaManT@Home* must be executed in another country different to the one it was originally designed .
- *Response: DiaManT@Home* is configured considering constraints imposed by country legislations. Specifically, information contained in repositories from the governance & policies layer are configured, as well as, the interoperation standards located in the information architecture layer.
- *Architectural decision(s):*
  - To create managers for configuring interoperation standards, and quality plans. Both managers, named plans configuration services and standards configuration, are located in the sub-layer system management contained in the consumer services layer of *DiaManT@Home*.
  - Plans and standards are stored in repositories.
- *Reasoning:*
  - **Benefits:** (i) High maintenance of elements in the governance and policies layer of *DiaManT@Home* ; (ii) Improve reuse of services located in other layers of *DiaManT@Home* ; (iii) Ensuring persistence of plans and standards.
  - **Liabilities:** (i) Reliability of configured plans and standards depend of mechanisms used by the plans configuration services and standards configuration services, and are out of scope of the reliability levels offered by the HSB; (ii) Mechanisms used to interpret plans and standards, e.g., transformation rules, must be defined by the system's manager to allow brokers and quality manager to interpret such plans and standards.
- *Architectural diagram:* Layered architecture of *DiaManT@Home* showed in Figure 78 in Section 6.2.

Resources: To establish quality scenarios, both software architects, of *DiaManT@Home* and *HomecARe*, were involved. In total were spent 28 hours, being spent 18 hours by the architect of *DiaManT@Home* and 10 hours by the architect of *HomecARe*. The spreadsheet in (GARCÉS, 2017) was used to register information about scenarios.

6.3. Analysis of collected data

In this section conclusions are derived based on information collected in Section 6.2. For each research question, conclusive statements are proposed offering evidence to support or refute the related hypothesis.

RQ1 - HomecARe improves the architectural design of HSH systems

To answer *RQ1 - Can the domain knowledge, models, architectural solutions, and architectural descriptions contained in HomecARe improve the architectural design of HSH systems?*, time and people required to conduct each activity of the instantiation process (in Appendix A) was registered. For this, at the end of each procedure in Section 6.2, information about time spent and people involved to design the software architecture of *DiaManT@Home* was offered.

To support the hypothesis that *"At using HomecARe, it is possible to decrease time and efforts when the software architecture of a HSH system is under designing"*, information about time and amount of people involved at architecting *DiaManT@Home* was compared with similar information reported by other two HSH systems projects, i.e., Dem@care[2] and HearthCycle[3]. Dem@care is a HSH system conceived to support patients with dementia condition at home. Dem@care was a 45 months (5,400 hours) research project funded by the European Union (Grant: FP7-288199) and developed by a consortium of seven European companies, three universities and one hospital. To create Dem@care, €10,626,393.oo, were invested. Meanwhile, HearthCycle is oriented to assist patients diagnosed with hearth failure. Such system was also funded by the European Union (Grant: FP7-216695) and was developed by a consortium of fourteen private organizations, eight universities, and one hospital. To develop HearthCycle, 66 months (7,920 hours) were required, and €21,962,747.oo were invested.

Available technical reports, work packages, papers, and articles of Dem@care and HearthCycle were under analysis to obtain information about time and people required to their development. Such information is compared with resources invested in *DiaManT@Home*. Figures 84 and 85 summarize information about time and people required to create the three HSH systems. In such figures, all HSH systems are compared regarding five activities: (i) domain and architectural analysis, (ii) architectural synthesis, (iii) architectural evaluation, (iv) implementation, and (v) testing. However, for the purpose of answering the RQ1, only the activities of *domain and architectural analysis*, *architectural synthesis*, and *architectural evaluation* are under consideration, since neither implementation nor testing activities have been conducted yet in the context of *DiaManT@Home*.

As showed in Figure 84, for the first activity of *domain and architectural analysis*, two

[2] http://www.demcare.eu/
[3] http://www.heartcycle.eu/

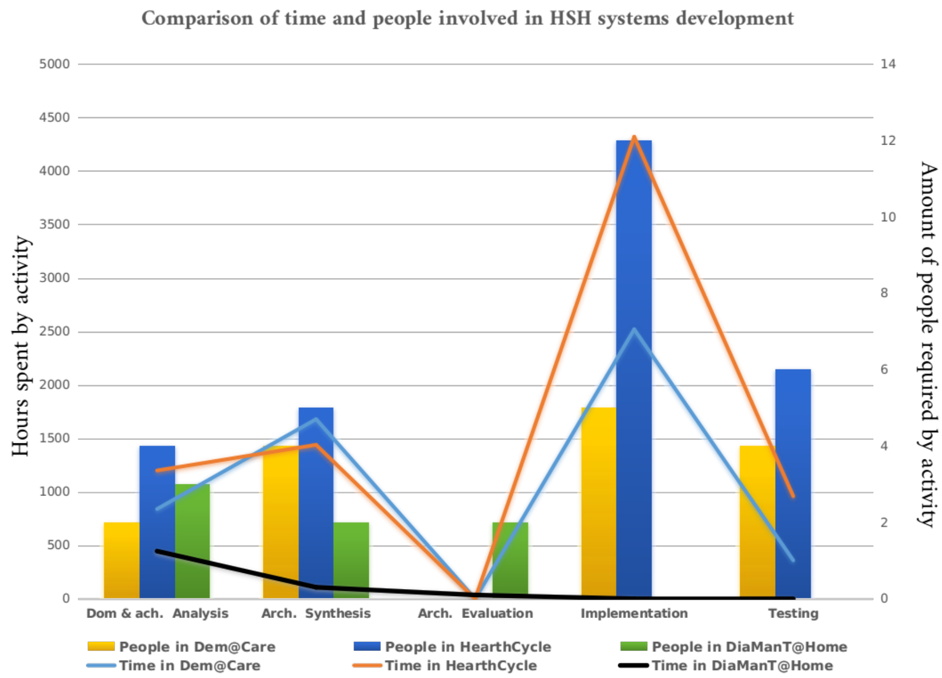


Figure 84 – Comparison of time and people involved in *DiaManT@Home* with other HSH systems

people were involved and 840 hours were spent in Dem@care. For the same activity, in HeartCycle, four people were required and 1,200 hours were necessary. Similarly, in *DiaManT@Home*, three people were involved and 446 hours were required.

To realize the second activity of *architectural synthesis*, in Dem@care were required 4 people and 1,680 hours. In HearthCycle, 5 people and 1440 hours were necessary. Finally, in *DiaManT@Home*, 2 people and 108 hours were needed.

The activity of *architectural evaluation* was only considered by *DiaManT@Home* requiring 2 people and 36 hours to validate the architecture of such system. The fact that only *DiaManT@Home* validated its architecture can be explained because both Dem@care and HearthCycle were implemented and tested in subsequent activities, and maybe, for such systems it was not a priority to previously validate their design decisions. Another explanation can be that they conducted such validation, but they did not report their results.

In total, as showed in Figure 85, to establish the software architecture, Dem@care required 2,520 hours, and HearthCycle spent 2,640 hours, whilst in *DiaManT@Home*, 582 hours were needed to define and also validate its architecture. Thus, approximately, *DiaManT@Home* spent only 23% of the time invested in related systems.

Considering this evidence, it is possible to affirm that, at using *HomecARe*, less time was needed to establish and validate the software architecture of *DiaManT@Home*, comparing with other HSH systems that did not use a reference architecture to support such activities. However, additional instantiations of *HomecARe*, in similar circumstances as presented in *DiaManT@Home*, must be performed to offer more evidences to support this hypothesis, due to

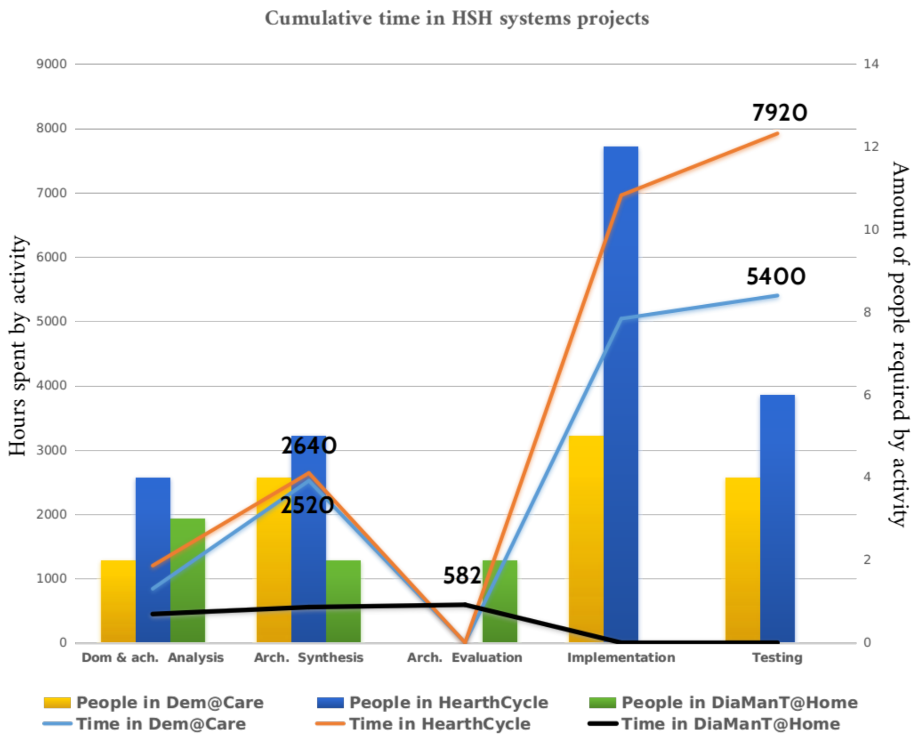


Figure 85 – Cumulative time spent in *DiaManT@Home* and in other HSH systems

both systems used to compare *DiaManT@Home* present different scopes, which influences time and people required in each project, since the domain or context of such systems can be more complex to be analysed and the selection of architectural solutions can be more challenged than in *DiaManT@Home*.

RQ2 - HomecARe allows to architect viable HSH systems

This section aims to bring evidence to answer *RQ2 - Can software architectures of HSH systems designed using HomecARe be considered viable solutions?* To consider *DiaManT@Home* as viable solution for its concrete problem, the accomplishment of their functional requirements is assessed. Hence, *DiaManT@Home* can be considered viable if, through its software architecture, it is possible to accomplish its requirements. For this, a mapping between *DiaManT@Home* requirements and architecture elements was done, as presented in Section 6.2, Table 15.

As detailed in Table 15, functional requirements of *DiaManT@Home* are considered by at least one of the architectural elements defined in its software architecture. Or in other words, each functional requirement is under the responsibility of at least one architectural element. With this evidence, it is possible to determine that *DiaManT@Home* offers a viable solution to achieve the functionalities expected for such system. However, it is not possible to fully support the hypothesis accompanying RQ2: *HomecARe allows to design software architectures of HSH systems valid for their concrete problems*. As documented in activity 3, related to the architectural synthesis of *DiaManT@Home*, in Section 6.2, *HomecARe* did not offer all types of abstract elements that *DiaManT@Home* required to address its functional requirements. Three new elements were needed in *DiaManT@Home* that *HomecARe* did not provide: (i) The *CSSv013 - Body substance assessment service* located in the constituent system services layer; (ii) the *Report service* placed in the home healthcare team services layer; and (iii) the *Calendar* repository located in the healthcare plans layer. Therefore, it is required to add such elements into *HomecARe* to affirm the hypothesis proposed for this RQ2.

RQ3 - HomecARe improves interoperability of HSH systems

To resolve *RQ3 - Is HomecARe an alternative to address interoperation issues of HSH systems?*, four interoperability scenarios, i.e., Scenarios 1 to 4, in Section 6.2, were analysed. Based on evidence obtained from such scenarios, the following hypothesis is intended to be supported: *At using HomecARe, an architecture of a HSH system can achieve interoperability of services provided by constituent systems.*

In HSH systems, interoperability is related with the capacity of participating services to exchange and correctly use the exchanged information. Moreover, interoperating services in a HSH system must be able to perform processes defined in the clinical guidelines to properly offer home care to patient's with chronic conditions.

To address interoperability, architectural decisions made and identified analysing the four scenarios include: (i) the use of messages to achieve technical interoperability between participating services of the HSH system; (ii) to achieve semantic and syntactic interoperability between constituent systems services (CSSv) and the HSB, each CSSv is connected to the HSB through a dynamically customized broker responsible for message encoding / decoding, and health information translation; (iii) the use of semantic approaches to represent semantic and syntactic interoperability standards and support the broker's activities; and (iv) the establishment of SOA approach to address business processes interoperability.

The evidence obtained allows to argue that architectural decisions made in *HomecARe* and instantiated in *DiaManT@Home* present high potential to address interoperability in HSH systems. This is, it is possible to ensure technical, semantic, syntactic, and business process interoperability among services participating of the HSH system. For this, services aiming to participate of the HSH system (e.g., constituent systems services, business process services, or consumer services) must have well-defined interfaces, messages formats, and terminology. This fact can reduce the diversity of services that can be considered adequate to be part of the HSH system.

RQ4 - HomecARe allows to create reliable HSH systems

Scenarios 5 to 8 in Section 6.2, were analysed to obtain the evidence to answer the research question *RQ4 - It is possible to create reliable software architectures of HSH systems using HomecARe ?* To estimate if operations in *DiaManT@Home* are reliable, as consequence of using *HomecARe* for its construction, the following evidence is considered:

1. Reliability of *DiaManT@Home* depends on the availability level of services provided by constituent systems (i.e., CSSv) to offer the required capabilities, and how reliable is the information offered by such services. In *DiaManT@Home*, the *reliability manager service*, located in the *quality of service layer*, is responsible to measure availability level of all participating services, as evidenced in Scenario 8. Moreover, quality of information provided by CSSvs is ensured through authentication and authorization mechanisms, as presented in Scenarios 6 and 7.

2. Reliability of *DiaManT@Home* relies on how reliable is the exchange of information between its entities. In *DiaManT@Home*, patient's information is exchanged using ACK messages to acknowledge that a message containing patient's information, or other sensitive data, was correctly received. This is specified in *HomecARe*, through interaction protocols that participating services must implement to conform contracts of services interfaces. Protocols of each service interface in *HomecARe* were presented in the mission viewpoint in Section 5.2.

3. Reliability of *DiaManT@Home* is influenced by its ability to satisfactorily deal with emergency situations. As stated in Scenario 5, emergency situations are detected and communicated by the HSB to the emergency manager, through the notification of specific messages containing information about the type of risk found. The emergency manager establishes an intervention plan to overcome the situation, and defines which emergency actions must be performed by which entities.

4. Reliability of *DiaManT@Home* is affected by its capacity to detect, prevent, and react to faults in its operations. As presented in Scenario 8, in *DiaManT@Home*, the *reliability manager service* is responsible to offer fault detection, prevention, and recovering mechanisms.

Considering this evidence, it is possible to support the hypothesis related with RQ4: *at using HomecARe, software architectures of HSH systems address reliable operations*. However, to fully confirm this hypothesis, additional decisions must be considered, specially, at selecting the type of transport protocols to be used in service interfaces, which was not specified in the current version of *HomecARe*.

RQ5 - HomecARe permits to design secure HSH systems

Scenarios 6 and 7 in Section 6.2 were analysed to obtain grounds for resolving *RQ5 - It is possible to instantiate secure software architectures of HSH systems using HomecARe?*

As instance of *HomecARe*, *DiaManT@Home* adopts architectural decisions made for designing the HSB. Therefore, mechanisms adopted by the HSB to address authentication, authorization, non-repudiation and confidentiality of patient's information (through using of encryption keys), are also presented in *DiaManT@Home*. Hence, based on evidence provided in security scenarios, i.e., Scenarios 6 and 7, it is possible to support the hypothesis that *at using HomecARe, software architectures of a HSH system can address security requirements.*

However, security mechanisms instantiated from *HomecARe* are only concerned with operations made at using the HSB. Thus, architects of HSH systems must ensure that all participating services in the system being only interconnected through the HSB, avoiding thus, the establishment of connection links outside such infrastructure.

RQ6 - HomecARe ensures the construction of adaptive architectures of HSH systems

Information presented in Scenarios 3 and 8 in Section 6.2 give insights to answer *RQ6 - It is possible to create adaptive architectures of HSH systems using HomecARe?*

In Scenario 3, it was evidenced the capacity of *DiaManT@Home* to add new services without the need of human intervention to configure them. This is achievable due to capabilities of the HSB for the discovering, integration, and reconfiguration services. Additionally, it is possible to observe that as the patient's condition change, the HSH system is also adapted to support the current patient's health situation. Moreover, in Scenario 8, it was presented how *DiaManT@Home*through the *reliability manager*, detects, predicts, and reacts to possible faults, and thus, to ensure a desired level of functionality based on quality plans defined in the governance and policies layer.

Since such characteristics were instantiated in *DiaManT@Home* from decisions made in *HomecARe*, it is possible to partially support the hypothesis that *instances of HomecARe can be considered as adaptive architectures.* Although, to fully sustain this hypothesis, changes in the software architecture of *DiaManT@Home* must be assessed at runtime to measure real adaptivity capabilities of such architecture.

RQ7 - HomecARe allows to reuse software architectures of HSH systems

Scenario 9 in Section 6.2 exhibits information to solve *RQ7 - Does HomecARe present low coupling regarding patient health conditions and her/his context, e.g., country legislations,* *health organization policies?* To answer RQ7, the following evidence is used: (i) standards, guidelines, plans, and policies are represented as well-defined data structures contained in repositories, allowing their persistence and modifiability; (ii) connections with such repositories are made using well established interfaces defined in *HomecARe*; (iii) the execution of business processes or parts of them are offered by independent services that can not be affected by the type of information stored in repositories; and (iv) managers services, located at the consumer services layer, facilitate the configuration of standards, guidelines, plans, and policies as required by country legislations.

In this perspective, logic contained in services originally defined in *DiaManT@Home* (e.g., CSSv and BPSv) can be reused in other contexts, supporting thus the hypothesis that *HomecARe can be used to support architectural design of HSH systems independently of patient's political, economical, regional, and health conditions.* However, it is important to consider that standards, guidelines, plans, and policies defined by country legislations must be established in such a way that can be interpreted by participating services in the HSH system. Therefore, it is required that information is well structured, e.g., using ontologies or other type of semantic approaches.

6.4. Discussion of Results

In this section, a synthesis of findings obtained from conducting the case study is presented. Specifically, it is provided an updated version of *HomecARe*, which contains new elements identified during the study conduction. Moreover, some liabilities and trade-off found in *HomecARe* are also presented.

Updated version of HomecARe

As result of conducting the case study, some additional elements were identified as important to be included in *HomecARe*: two services (the body substance assessment service (with ID CSSv013) and the reports service), and the calendar repository. The *body substance assessment service* is responsible for measuring chemical and physical compositions of substances from patient's body, such as blood, breath or water, and to publish such measures into the HSB. The *reports service* gathers information of interest about patient's conditions and treatment evolution and makes it available to members of the home healthcare team. Finally, the *calendar repository* contains patient's schedule detailing all activities required to fulfil her/his treatments. These elements are depicted in Figure 86 with a star icon.

Similarly, possible mandatory elements in *HomecARe* were identified and are presented in Figure 86 using an icon of a closed padlock. However, more instances of *HomecARe* must be designed to verify the mandatory elements for HSH systems. The remaining elements defined in *HomecARe* are considered optional and their selection to conform or not the software architecture of a HSH system depends of system's specificities, which are defined in the two first activities of the instantiation process of *HomecARe*.

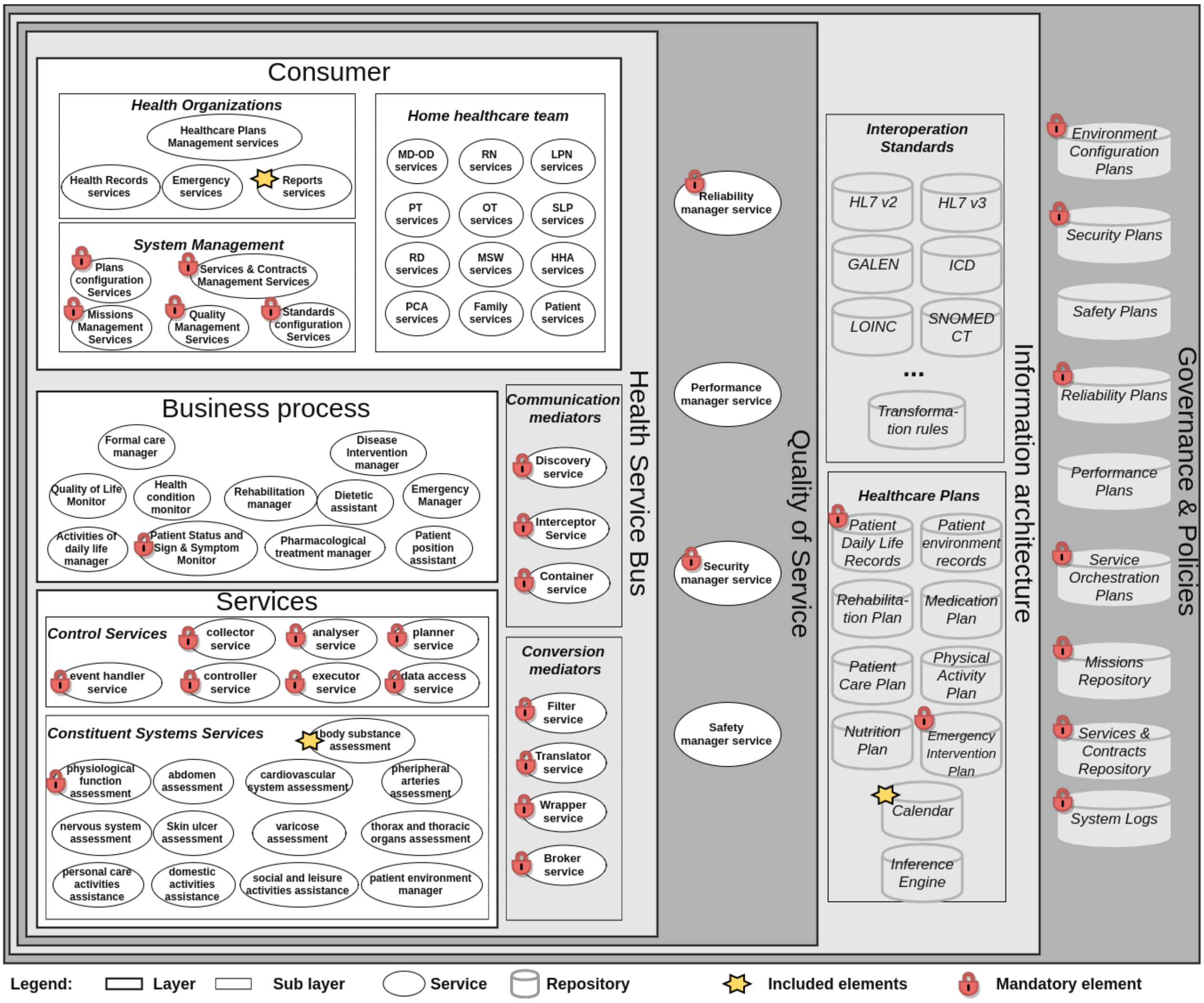


Figure 86 – Updated version of *HomecARe* after validation.

Liabilities of using HomecARe to develop HSH systems

The instantiation process of *HomecARe*, as presented in Appendix A, allowed a systematic design of *DiaManT@Home* software architecture. However, models conforming *HomecARe*, e.g., missions, BPMN, SoaML, or UML, were manually related and instantiated by the architect, which is an error prone task. In this perspective, automatic techniques to assist the instantiation, verification, and validation of models used to represent software architectures of HSH systems would improve the architecting process of those systems using *HomecARe*.

An approach for continuous updating *HomecARe* must be defined, since new elements or possible modifications in its services specifications or governance methods can appear. The lack of such approach might lead to inconsistencies between versions of *HomecARe* and its instantiated architectures. Additionally, this approach could be used to ensure the sustainability, evolution, and maturity of *HomecARe* and its instances over time.

The current version of *HomecARe* does not support code generation of its instantiated architectures. Hence, it is not possible to ensure that a HSH system implementation is conform to *HomecARe*. This fact limits the use of *HomecARe* during the whole life cycle of HSH systems. Therefore, coding verification against architectural models is key to allow that *HomecARe* can be considered a standard for the development of HSH systems.

Throughout the conduction of the instantiation process, it was not possible to measure, in a reliable way, time and effort required for understanding how to use *HomecARe*. The learning curve of *HomecARe* can affect its viability in future projects, since the architect must be familiar with concepts from different domains, such as chronic conditions, AAL, HSH, e-Health, SoS, BPMN, and SOA, and be aware about constraints over service interfaces, contracts, and protocols established in *HomecARe*.

In a similar perspective, concerning time and effort to conduct the instantiation process, and architect *DiaManT@Home*, no direct correlation between amount of people involved and time required in each activity was defined. Neither *Dem@care* nor *HeartCycle* offered enough evidence to calculate amount of time spent by each person in each activity, thus, absolute time measurements were used to compare time spent in both systems with the time required in *DiaManT@Home*. Therefore, evidence used to resolve RQ1 allows to analyse, in an independent way, both time and people required in the three HSH systems. In this context, the fact that *Dem@care* and *HeartCycle* required more efforts to establish their architectures could be more associated to the project complexity than the fact of these systems being supported by a reference architecture. Therefore, more investigations must be made to fully support the hypothesis that less efforts are required in the construction of architectures of HSH systems using *HomecARe*.

Another conclusion made about the use of *HomecARe*, is that reference architecture does not specify, in depth, how the scope (i.e., patient's conditions to be supported) of a HSH system must be defined. Hence, the first activity of the instantiation process requires an important effort by the architect of the HSH system to obtain an adequate domain knowledge (e.g., understanding of conditions and related procedures for which the system is being constructed) from clinical guidelines, interviews or studying other information sources. Thus, open and well defined domain knowledge supported by a broad community of home healthcare team members can facilitate this activity for software architects of HSH systems.

Finally, HSH systems instantiated from *HomecARe* are patient-oriented, therefore, monitoring multiple patients in the same environment (as in the case of nursing homes) is not supported yet. In this way, additional decisions must be made to allow the execution of multiple instances of different HSH systems that assist different patients in the self-management of their specific medical conditions.

HomecARe and Performance

Requirements regarding interoperability, reliability, security, and adaptivity were considered as the ASRs for *HomecARe*. Those requirements were identified in the phase of architectural analysis of *HomecARe*, as detailed in Section 4.2. To satisfy such requirements, important decisions were adopted during the architectural design of *HomecARe*, as presented in the quality scenarios specification described in Section 6.2; however, some of these decisions can have a negative impact on performance.

Through the conduction of the case study presented in this chapter, it was possible to determine that selected strategies to address interoperability (e.g., brokers to transform messages content and formats, or overhead of message headers interpretation by the interceptor of the HSB) can have a negative impact on performance of the HSB, and on the HSH system. Specifically, it is possible to have low performance when messages are communicated among constituent systems services and other entities participating of the HSH system. Similarly, the use of authorization, authentication, and encrypting mechanisms to address security in HSH systems adversely affects their performance, since for each published message in the HSB, publisher's credentials must be verified and patient's information must be encrypted and decrypted, increasing the processing and delivery time of messages. Moreover, the establishment of an holistic and updated view of the HSH system for reliability purposes negatively impacts its performance, since it is required to process a huge amount of information, communicated by constituent systems services, to identify possible faults or emergency situations. In this perspective, additional decisions must be adopted in *HomecARe* to grant low processing time when HSH systems are achieving their missions without affecting interoperability, reliability, security, and adaptivity.

6.5. Threats to Validity

To ensure the trustworthiness of results obtained at conducting the case study presented in this chapter, four aspects of validity were considered, as proposed in (RUNESON *et al.*, 2012), namely, construct, internal, external, and reliability of the study. For each threat to these viability aspects, one or more approaches to mitigate its impact in results analysis were proposed, and are presented as follows.

Construct validity: This validity aspect reflects in which measure, de facto, the units of analysis and data used to measure such units, contribute to answer the research question for which they were defined (RUNESON *et al.*, 2012). To avoid threats to the construct validity, the guidelines proposed in (RUNESON; HÖST, 2008; RUNESON *et al.*, 2012) were followed to support the planning, conduction, analysis, and reporting of this case study. Moreover, the case study planning was peer reviewed to ensure a correct execution of the study. Hence, general objective, research questions, units of analysis, and collected data were reviewed by two software engineering researchers before the case study conduction. However, more data could be collected to support triangulation of evidence and improve results obtained in this case study. Replication of this case study in other HSH systems can also offer additional evidence to support validation of *HomecARe*.

Internal validity: This validity aspect considers whether all causal relations that affect the results analysis of this case were defined and handled. The following factors that could prejudice the objective of this case study were identified: (i) the learning curve of *HomecARe*, which had no impact on results, since the architect of *DiaManT@Home* previously knew the reference architecture before conducting the study; (ii) the difficulty to understand the instantiation process, which was avoided, since this process was based on a well-known software architecture design approach (HOFMEISTER *et al.*, 2005); (iii) the comprehension of architectural solutions established in *HomecARe*, which was prevented, since this reference architecture was described using viewpoints and views (as established in (ISO/IEC/IEEE, 2011)), and models constructed employing BPMN, SoaML, and UML; and (iv) the problems at defining the HSH system scope, which were resolved through the use of missions models of *HomecARe*, and requirements documentation of *DiaManT@Home*.

External validity: This validity aspect is concerned with the generalization of results obtained at conducting this case study, i.e., it is possible to obtain similar results in other HSH systems at using *HomecARe*. Therefore, guidelines to instantiate *HomecARe* were proposed based on well-established architectural design process (HOFMEISTER *et al.*, 2005). Such guidelines, presented in Appendix A, could be used to instantiate other HSH systems. Documents supporting activities in such process are also provided in (GARCÉS, 2017) and could be used in other HSH systems projects to document results of each activity. However, it is possible that during the establishment of other HSH systems, modifications in the instantiation process, and possibly in *HomecARe*, could be required depending on the specificities of systems under design.

Reliability of the study: This validity aspect is concerned with the dependence level of the analysis of this case study results with the specific researchers (RUNESON *et al.*, 2012). To improve the reliability of results presented in this case study, the guidelines proposed in (RUNESON *et al.*, 2012) were followed. Hence, the study was designed and planned, defining its objective, research questions, hypothesis, units of analysis, and methods to collect data, as presented in Section 6.1. Data were collected following the planned methods, correctly coded to avoid misunderstandings, and documented in a spreadsheet, which was created to support the analysis of this study. Collected data are available in (GARCÉS, 2017) to be consulted by other researchers that desire to replicate the results obtained in this study. Finally, qualitative analysis of these data were used and reported in Section 6.3.

6.6. Final Considerations

In this chapter, results of evaluating *HomecARe* were presented. Evaluation was made through the conduction of a case study that was designed, planned, conducted, and reported following the guidelines proposed in (RUNESON *et al.*, 2012). The objective of this study was to validate the viability of *HomecARe* to support the software architecture design of HSH systems. Therefore, the software architecture of *DiaManT@Home*, was designed following the instantiation process of *HomecARe*. Optimistic results were obtained, since the domain and architectural knowledge offered in this reference architecture was reused and less time was required to design the software architecture of *DiaManT@Home*, comparing with the spent in other HSH systems, as reported in Section 6.2. Moreover, architectural solutions offered in *HomecARe* allow the accomplishment of HSH systems requirements of interoperability, security, adaptivity, and reliability, as evidenced in the quality scenarios presented in Section 6.2. Section 6.4 presented an updated version of *HomecARe* containing new elements important for other HSH systems, a description of liabilities evidenced after conducting this case study, and the necessity of include tactics to improve performance of HSH systems in future versions of *HomecARe*. Threats to the four validity aspects found in case studies (RUNESON *et al.*, 2012), namely, construct, internal, external, and reliability of the study, were identified and, mitigated, to the extend possible, to ensure the trustworthiness of results obtained in the case study presented in this chapter.

CHAPTER

7

CONCLUSIONS

HSH systems have been conceived as an alternative to offer healthcare services at home to patients suffering from one or more chronic conditions, improving thus, their quality of life and autonomy and, at the same time, decreasing costs for governmental entities. Nowadays, given the high rates of population ageing at world wide, HSH systems are an important alternative to support elders in their activities of daily living. In this perspective, a great amount of HSH systems have been developed over the world, aiming to assist patients in the self-management of chronic diseases, such as dementia, Alzheimer's, stroke, heart failure, Parkinson's and chronic obstructive pulmonary diseases. HSH systems have been proposed by different consortia from different countries, and are composed of a variety of technologies including, sensor networks, medical devices, e-Health systems, robotic systems, smart homes systems, and ambient intelligence solutions. Despite the great diversity of available HSH systems in market, they are proprietary and monolithic solutions, presenting high coupling between their entities, and are considered a high expensive alternative for patients, their relatives, and health care providers. Most of these issues are presented due to the lack of interoperability between HSH systems, and the absence of approaches promoting the reuse of domain and technical knowledge for creating such systems in a standardized way.

The aim of this thesis is to contribute to the development of interoperable, reliable, secure and adaptive HSH systems, offering an alternative to solve current problems presented in the domain. Specifically, in this thesis is proposed *HomecARe*, a reference architecture that, constructed from the perspective of SoS, offers domain and architectural knowledge that can be reused to create quality HSH systems.

This chapter presents important contributions to different topics that have been made as result from the development of *HomecARe*. Specifically, contributions made during the conduction of this thesis are revisited in Section 7.1. Limitations and future works are detailed in Section 7.2. Finally, possible extensions are described in Section 7.3.

7.1. Revisiting the Thesis Contributions

The main contributions of this thesis are framed in topics of AAL, e-Health, software quality, SoS, reference architecture, and software architecture, and are summarized as follows.

Quality in the AAL domain: A systematic mapping investigating how quality have been treated in AAL systems (GARCÉS *et al.*, 2017a) was conducted following the guidelines proposed in (KITCHENHAM; CHARTERS, 2007). As results, the most important quality attributes considered in developed AAL systems were identified and reported (GARCÉS *et al.*, 2017a). Based on the knowledge obtained at performing this mapping, a quality model for AAL systems, named QM4AAL, was proposed (GARCÉS; OQUENDO; NAKAGAWA, 2016). QM4AAL relates stakeholders, adaptive properties, consolidated definitions and quality attribute requirements. In total, QM4AAL provides 175 types of quality attributes requirements that can be used as a basis to establish requirements for AAL systems. In this thesis, QM4AAL was used to identify architectural significant requirements for both *HomecARe* and *DiaManT@Home*, demonstrating thus, its viability to support AAL systems development. QM4AAL was presented in Section 3.4.

Process for domain analysis of reference architectures for SoS: ProSA-RA (NAKAGAWA *et al.*, 2014) is an approach used to engineer reference architectures and was used as a basis to establish *HomecARe*. However, ProSA-RA does not consider important issues when the domain analysis activity for SoS (i.e., HSH systems) is being performed. For this, a process to support the activity of domain analysis for establishing reference architectures for SoS was proposed (GARCÉS; NAKAGAWA, 2017). This process is based on goal oriented requirement engineering approaches (LAMSWEERDE, 2001), and mission-models oriented techniques proposed for SoS requirements elicitation (SILVA; BATISTA; OQUENDO, 2015). This process was followed to establish domain models of *HomecARe*, i.e., missions, responsibilities, capabilities, data entities and emergent behaviours models. Domain models were further used in the architectural analysis activity to identify capabilities required from constituent systems services, and to establish services interfaces, contracts and protocols. Moreover, domain models of *HomecARe* were used by the architect of *DiaManT@Home* to understand HSH domain, and thus, to orient the selection of capabilities and services required in its architecture. In this context, at using the proposed process for the domain analysis of *HomecARe*, it was given evidence about its applicability in reference architectures for SoS. This process and results of its application in *HomecARe* were presented in Chapter 4.

Taxonomy of mediators: In SoS, mediators can be seen as first class entities that offer reusable structure and functionality for purposes of communication, coordination, cooperation and collaboration among constituent systems and other SoS entities. To the best of our knowledge, no investigation exists detailing which and how mediators must be considered in SoS architectures. In this context, a taxonomy of mediators was proposed in this thesis. Specifically, twelve mediators in three categories were proposed to provide capabilities of communication, conversion and control to SoS software architectures. In *HomecARe*, those mediators were used as primary structures of the HSB to address interoperability and integration of participating services of HSH systems. It is expected that mediators in this taxonomy can be reused as primary elements to facilitate interactions between heterogeneous and independent constituent systems in other type of SoS. The mediators taxonomy was introduced in Section 2.2 and is detailed in Appendix C.

Health Service Bus: In e-Health systems it is possible to find several proposals using the Enterprise Service Bus (ESB) to mediate communications between services, such as presented in (RYAN; EKLUND, 2010; SIDDIQUI, 2010; CRICHTON *et al.*, 2013; MERIDOU *et al.*, 2015; ZEINALI; ASOSHEH; SETAREH, 2016). However, those solutions have been proposed for specific systems and technologies and do not consider implicit characteristics of dynamic architectures as the ones presented in SoS. In *HomecARe*, the HSB offers the infrastructure required to mediate, communicate, coordinate, and convert messages sent by participating services of HSH systems. Moreover, the HSB was conceived to support the achievement of interoperability, security, reliability and adaptivity requirements of HSH systems. The HSB proposed in this thesis can be used as a reference for the establishment of HSB in other e-Health systems, since it offers an abstract, low coupling, flexible, dynamic, scalable and evolving solution to mediate interoperable services following health information standards. The HSB was detailed in Chapter 5.

Missions viewpoint: Different architectural viewpoints (e.g., logical, runtime, data, and physical viewpoints) can be used to document, from different perspectives, architectural decisions in reference architectures as those reported in (GUESSI; BUENO; NAKAGAWA, 2011). Viewpoints are structured in views that, using different models or diagrams represent solutions for specific stakeholders concerns. Examples of architectural views used to represent reference architectures are: conceptual, module, component & connector, deployment, enterprise, computational and information views. Considering that HSH systems share characteristics of SoS, it was proposed the mission viewpoint to represent how architectural decisions made in *HomecARe* allow the accomplishment of HSH systems missions. Thus, for each mission defined in *HomecARe*, an architectural configuration was defined. Each configuration is represented in an architectural view composed by business process, capabilities, services interfaces, services contracts, services protocols and participants diagrams. Therefore, the mission viewpoint describes all possible architectural configurations that HSH systems can have to address their missions. As the mission viewpoint was proposed based on the standard 42010 (ISO/IEC/IEEE, 2011) for architectural description, such viewpoint can be proposed as an alternative to describe architectures of SoS in other domains. The mission viewpoint was presented in Chapter 5.

***HomecARe*:** A reference architecture for guiding the architectural design of HSH systems from a perspective of SoS was proposed. *HomecARe* provides domain knowledge, domain models, a quality model, architectural solutions represented as viewpoints and views to allow the design of HSH systems architectures. To support the use of *HomecARe* in concrete HSH projects, a SoaML project containing all SOA models of *HomecARe* is offered to facilitate the selection of participating services, services interfaces, contracts and protocols that will compose the architecture of the concrete system. Additionally, instantiation guidelines also assist the application of *HomecARe* in HSH projects. The viability of *HomecARe* for supporting the architectural design of HSH systems was assessed through a case study, which demonstrated positive results. In this context, *HomecARe* can be considered a systematic approach to orient the architecture design of systems for assisting patients suffering of chronic conditions at the management of their diseases at home. Chapters 3 to 5 presented in detail all knowledge contained in *HomecARe*.

***DiaManT@Home*:** The software architecture of a HSH system was designed as instance of *HomecARe*. *DiaManT@Home* aims to assist patients suffering from diabetes mellitus disease in the self-management of their condition within their homes. As instance of *HomecARe*, *DiaManT@Home* inherits architectural decisions, demonstrating thus, its capability to address security, interoperability, reliability and adaptivity. It is expected that *DiaManT@Home* can be used by real patients in real conditions. Chapter 6 details the software architecture of *DiaManT@Home*.

7.2. Limitations and Future Works

In this section, limitations of this thesis are presented as well as, some approaches that can be used to tackle them in the future. Notice that in Chapter 6, several liabilities of using *HomecARe* to create HSH systems were presented. Therefore, herein, general limitations identified during the conduction of each phase of the process followed to construct and validate *HomecARe* are described, and possible solutions to overcome limitations are detailed as follows.

- Despite *HomecARe* offers solutions to address interoperability issues of HSH systems (i.e., technical, semantic, syntactic, and business interoperability), some challenges must still be considered in this reference architecture. Interoperability standards, proposed for the health domain and used in *HomecARe*, such as SNOMED CT, ICD, LOINC or HL7, need to be formally represented to facilitate their conversion by the *broker* defined in the HSB. For this, ontologies or other semantic technologies can be used, allowing the mapping and conversion between standards.

- Moreover, strategies used in *HomecARe* to address business interoperability are based on the formalization of work flows defined in clinical guidelines for treating chronic conditions. However, such guidelines change regarding country legislations and are principally established as text books. In this context, it is required the formal modelling of clinical guidelines to allow the establishment of business processes that HSH systems must follow. Hence, *HomecARe* must evolve to offer means to systematically structure clinical guidelines, and automatically transform such structures in business process models, which must be used to define services choreographies or orchestration of architectures configurations of HSH systems.

- One of the strategies adopted to address security requirements was the definition of user's profiles, roles, and permissions over system's resources and operations. Based on stakeholders classification in *HomecARe*, it is possible to define in a generic way roles for types of users. However, activities performed by each stakeholder can vary depending of constraints imposed in clinical guidelines of each country. Therefore, means to support the identification and configuration of user's profiles, roles and permissions from clinical guidelines, must be included in *HomecARe* . An alternative approach is the formal representation of clinical guidelines through the use of ontologies, allowing the selection of activities allowed to be performed by specific types of stakeholders.

- Domain models of *HomecARe* were defined based on a systematic process to identify, model and assess domain models of SoS in reference architectures (GARCÉS; NAKAGAWA, 2017). The main purpose was to describe the most domain knowledge as possible for HSH systems. However, as SoS evolve over time (and also do HSH systems), new missions will arise and current models presented in *HomecARe* will require modifications, triggering possible changes in architectural models. In this perspective, automatic approaches to support evolutionary development of *HomecARe* and of its instantiated architectures are required. Moreover, the process to construct reference architectures for SoS must be refined to include their evolution over time.

- During the establishment of *HomecARe*, mapping of information contained in domain models into architectural models was made manually, increasing design time. Models transformation techniques can support the automatic configuration of services architectures of HSH systems based on domain information, decreasing thus required time for this activity.

- In *HomecARe*, health information types to be managed by the HSB were defined following the standard SNOMED CT. Additionally, services interfaces providing specific capabilities to communicate health information were established. However, architectural models do not represent which types of information are managed by each interface. In this perspective, investigations about how to represent domain data (formatted following the standard SNOMED CT) into services interfaces, protocols, and contracts for each capability involved in a services architecture configuration must be conducted.

- Finally, the viability of *HomecARe* was assessed using evidence from conducting the instantiation process in a single HSH system, i.e., *DiaManT@Home*. Despite obtained results were mostly positive, replications of case study must be conducted in different types of HSH systems.

7.3. Possible Extensions

Many opportunities of research emerged during the development of this thesis. They represent perspectives of future research that can contribute to the areas of e-Health, HSH systems, and software engineering of Systems-of-Systems. Some of them are described as follows.

HSB as Infrastructure for e-Health SoS: The Health Service Bus proposed in *HomecARe* can be extended as infrastructure for other types of e-Health systems besides HSH systems. This is possible since the HSB was designed as a generic solution to achieve interoperation and integration of healthcare services, offering mediation, routing, transportation, and protocol transformation. Instances of HSB, which support heterogeneous and independent e-Health systems, can be interconnected to offer more complex solutions as proposed in (SIDDIQUI, 2010). In this perspective, aggregation of HSBs can be used as an alternative to promote the interoperation of e-Health SoS.

Pattern Language for SoS: Architectural decisions made in *HomecARe* to achieve interoperability, reliability, security, and adaptivity can be investigated to obtain evidence about their viability to solve similar problems in other types of SoS. Such investigation will allow the consolidation of a pattern language that serves as backbone to architect further SoS in different domains.

***HomecARe* as PLA:** Since *HomecARe* contains the domain and architectural knowledge required to design architectures of HSH systems, it can be used as basis to formalize products line architectures for HSH systems, being each product (a HSH system) oriented to specific patient's conditions. Henceforth, variability points in *HomecARe* must be formally represented. For this, approaches from areas of software variability management, product line architecture, and software product line can be used.

Development Framework for HSH Systems: The architectural knowledge contained in *HomecARe* can be concreted as a development framework or software environment for building and deploying HSH systems. Hence, formal representations, e.g., model-driven architecture techniques, ADLs or constraint languages, are required to allow the verification of HSH systems code regarding architectural models of *HomecARe*.

BIBLIOGRAPHY

AALIANCE (Ed.). **Ambient Assisted Living Roadmap**. Amsterdam, The Netherlands: IOS Press, 2010. 136 p. (Ambient Intelligence and Smart Environments, v. 6). Citations on pages 1, 2, 3, and 27.

AMIRAT, A.; OUSSALAH, M. First-class connectors to support systematic construction of hierarchical software architecture. **The Journal of Object Technology**, v. 8, n. 7, p. 107–130, 2009. Citations on pages 23 and 203.

AMIRIBESHELI, M.; BENMANSOUR, A.; BOUCHACHIA, A. A review of smart homes in healthcare. **Journal of Ambient Intelligence and Humanized Computing**, v. 6, n. 4, p. 495–517, 2015. Citations on pages 37 and 55.

ANGELOV, S.; GREFEN, P.; GREEFHORST, D. A classification of software reference architectures: Analyzing their success and effectiveness. In: **Working IEEE/IFIP Conference on Software Architecture (WICSA)**. Cambridge, United kingdom: IEEE, 2009. p. 141–150. Citation on page 10.

____. A framework for analysis and design of software reference architectures. **Information and Software Technology**, Elsevier B.V., v. 54, n. 4, p. 417–431, Apr. 2012. ISSN 09505849. Citations on pages 13 and 14.

ANGELOV, S.; TRIENEKENS, J.; KUSTERS, R. Software Reference Architectures - Exploring Their Usage and Design in Practice. In: DRIRA, K. (Ed.). **7th European Conference on Software Architecture (ECSA)**. Montpellier, France: Springer Berlin Heidelberg, 2013. p. 17–24. Citation on page 5.

ANGELOV, S.; TRIENEKENS, J. J. M.; GREFEN, P. Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case. In: MORRISON, R.; BALASUBRAMANIAM, D.; FALKNER, K. (Ed.). **Software Architecture**. Paphos, Cyprus: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 5292). p. 225–240. Citation on page 13.

ANTONINO, P. O.; SCHNEIDER, D.; HOFMANN, C.; NAKAGAWA, E. Y. Evaluation of AAL platforms according to architecture-based quality attributes. In: KEYSON, D.; MAHER, M.; STREITZ, N.; CHEOK, A.; AUGUSTO, J.; WICHERT, R.; ENGLEBIENNE, G.; AGHAJAN, H.; KROSE, B. (Ed.). **Ambient Intelligence**. Amsterdam, The Netherlands: Springer Berlin Heidelberg, 2011. (Lecture Notes in Computer Science, 7040 LNCS), p. 264–274. Citation on page 3.

ARMITAGE, J. W. **Process Guide for the DSSA Process Life Cycle**. Pennsylvania, 1993. 62 p. Citation on page 12.

ARSANJANI, A.; ZHANG, L. J.; ELLIS, M.; ALLAM, A.; CHANNABASAVAIAH, K. S3: A service-oriented reference architecture. **IT Professional**, v. 9, n. 3, p. 10–17, May 2007. ISSN 1520-9202. Citations on pages 61, 62, 68, 69, and 70.

AVGERIOU, P.; ZDUN, U. Architectural patterns revisited – a pattern language. In: **In 10th European Conference on Pattern Languages of Programs**. Irsee, Germany: EuroPlop, 2005. p. 1–39. Citations on pages 19 and 21.

AVILA, K.; SANMARTIN, P.; JABBA, D.; JIMENO, M. Applications based on service-oriented architecture (soa) in the field of home healthcare. **Sensors**, v. 17, n. 1703, p. 1–16, 2017. Citation on page 37.

BAL, M.; SHEN, W.; HAO, Q.; XUE, H. Collaborative smart home technologies for senior independent living: A review. In: **15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)**. Lausanne, Switzerland.: IEEE, 2011. p. 481–488. ISBN 9781457703874. Citation on page 4.

BANDARA, N. **Connected Health Reference Architecture**. 2017. Available: <https://wso2.com/whitepapers/connected-health-reference-architecture/>. Citations on pages 25, 27, and 28.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. 2. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. 528 p. Citations on pages 10, 11, 13, 119, 122, and 128.

BAYER, J.; FORSTER, T.; GANESAN, D.; JOHN, I.; KNODEL, J.; KOLB, R.; MUTHIG, D. **Definition of Reference Architectures based on Existing Systems**. Kaiserslautern, Germany, 2004. Available: <http://publica.fraunhofer.de/dokumente/N-21572.html>. Citations on pages 13 and 14.

BEALE, D.; BONOMETTI, J. **Chapter 2: Systems Engineering (SE) – The Systems Design Process.** 2006. Available: <http://www.eng.auburn.edu/~dbeale/ESMDCourse/Chapter2.htm>. Citation on page 16.

BENNACEUR, A.; ISSARNY, V. Automated synthesis of mediators to support component interoperability. **IEEE Transactions on Software Engineering**, v. 41, n. 3, p. 221–240, March 2015. ISSN 0098-5589. Citations on pages 23 and 203.

BENNETT, J.; ROKAS, O.; CHEN, L. Healthcare in the smart home: A study of past, present and future. **Sustainability**, v. 9, n. 840, p. 2–23, 2017. Citations on pages 34, 35, and 38.

BENSON, T.; GRIEVE, G. **Principles of Health Interoperability SNOMED CT, HL7 and FHIR.** Third edition. London: Springer-Verlag London, 2016. (Health Information Technology Standards). Citations on pages 38, 39, and 40.

BERGER, M.; FUCHS, F.; PIRKER, M. Ambient intelligence –from personal assistance to intelligent megacities. In: **Proceedings of the 2007 Conference on Advances in Ambient Intelligence**. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 21–35. ISBN 978-1-58603-800-7. Available: <http://dl.acm.org/citation.cfm?id=1563773.1563776>. Citation on page 24.

BORG, Z. **A reference architecture for marine systems**. Phd Thesis (Master's Thesis.) — University of Kaiserslautern and Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany, 2011. Citation on page 14.

BOSCH, J.; RAN, A. Evolution of Software Product Families. **LNCS**, v. 1951, n. Software Architectures for Product Families, p. 168–183, 2000. Citation on page 12.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C.; MALER, E.; YERGEAU, F. E. **Extensible Markup Language (XML) 1.0 (Fifth Edition).** Online, 2008. Available: <https://www.w3.org/TR/REC-xml/>. Citation on page 113.

BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M. **Pattern-oriented Software Architecture: A System of Patterns**. New York, NY, USA: John Wiley and Sons, Inc., 1996. Citation on page 12.

CAMARINHA-MATOS, L.; FERRADA, F.; OLIVEIRA, A.; ROSAS, J. **BRAID project. D6.22. Strategic research agenda for ICT and Ageing.** Europe, 2011. Citations on pages 35 and 56.

CAMARINHA-MATOS, L.; FERRADA, F.; OLIVEIRA, A.; ROSAS, J.; MONTEIRO, J. Care services provision in ambient assisted living. **IRBM**, v. 35, n. 6, p. 286 – 298, 2014. ISSN 1959-0318. Healthcom 2013. Citations on pages 25, 27, and 28.

CAMPANA, F.; MORENO, A.; RIAÑO, D.; VARGA, L. Z. K4care: Knowledge-based home-care e-services for an ageing europe. In: ____. **Agent Technology and e-Health**. Basel: Birkhäuser Basel, 2008. p. 95–115. ISBN 978-3-7643-8547-7. Available: <https://doi.org/10.1007/978-3-7643-8547-7_6>. Citations on pages 40, 47, and 48.

CANADA, P. H. A. of. **Disease Prevention and Control Guidelines**. 2017. Online. Available: <https://www.canada.ca/en/public-health/services/reports-publications/disease-prevention-control-guidelines.html>. Citations on pages 40 and 48.

Centers for medicare and medicaid services. **Chronic conditions among Medicare beneficiaries**. Baltimore,MD., 2012. 30 p. Citations on pages 17, 1, and 2.

CHOPRA, A. Business process interoperability:extended abstract. In: PARSONS, P. P. M. (Ed.). **7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)**. Estoril, Portugal: International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, 2008. p. 1730–1731. Citation on page 40.

CLEMENTS, P.; KAZMAN, R.; KLEIN, M. **Evaluating software architectures. Method and case studies.** Second edition. Boston, MA, USA: Addison Wesley., 2002. (SEI Series in Software Engineering.). Citations on pages 128, 129, and 141.

CLOUTIER, R.; MULLER, G.; VERMA, D.; NILCHIANI, R.; HOLE, E.; BONE, M. The Concept of Reference Architectures. **Systems Engineering**, John Wiley and Sons Ltd., Chichester, UK, v. 13, n. 1, p. 14–27, 2010. ISSN 1098-1241. Citations on pages 12 and 13.

CRICHTON, R.; MOODLEY, D.; PILLAY, A.; GAKUBA, R.; SEEBREGTS, C. J. An architecture and reference implementation of an open health information mediator: Enabling interoperability in the rwandan health information exchange. In: ____. **Foundations of Health Information Engineering and Systems: Second International Symposium, FHIES 2012, Paris, France, August 27-28, 2012. Revised Selected Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 87–104. ISBN 978-3-642-39088-3. Available: <https://doi.org/10.1007/978-3-642-39088-3_6>. Citation on page 165.

CUESTA, C.; ROMAY, M. P. Elements of self-adaptive systems – a decentralized architectural perspective. In: ____. **Self-Organizing Architectures: First International Workshop, SOAR 2009, Cambridge, UK, September 14, 2009, Revised Selected and Invited Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 1–20. ISBN 978-3-642-14412-7. Available: <https://doi.org/10.1007/978-3-642-14412-7_1>. Citation on page 21.

CUESTA, C. E.; FUENTE, P. de la; BARRIO-SOLáRZANO, M. Dynamic coordination architecture through the use of reflection. In: **Proceedings of the 2001 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2001. (SAC '01), p. 134–140. ISBN 1-58113-287-5. Available: <http://doi.acm.org/10.1145/372202.372298>. Citation on page 21.

CUESTA, C. E.; NAVARRO, E.; DEWAYNE, E.; RODA, C. Evolution styles: using architectural knowledge as an evolution driver. **Journal of Software Evolution: Evolution and Process.**, v. 25, p. 957–980, 2013. Citation on page 20.

DAHMANN, J.; REBOVICH, G.; LANE, J. A. Systems engineering for capabilities. **Cross Talk. The Journal of Defense Software Engineering.**, v. 21, n. 11, p. 4–9, 2008. Citation on page 17.

DENTI, E. Novel pervasive scenarios for home management: the butlers architecture. **SpringerPlus**, v. 3, n. 1, p. 52, Jan 2014. Citation on page 25.

DOBRICA, L.; NIEMELA, E. An approach to reference architecture design for different domains of embedded systems. In: **2008 International Conference on Software Engineering Research and Practice (SERP)**. Las Vegas, NV, United States: CSREA Press, 2008. p. 287–293. Citations on pages 13 and 14.

DUARTE, L. **Establishment of a reference architecture for digital television applications**. Phd Thesis (Master's Thesis.) — University of São Paulo, São Carlos, Brazil, 2012. Citation on page 14.

DUCROU, A. **Complete Interoperability in Healthcare: Technical, semantic and process Interoperability through ontology mapping and distributed enterprise Integration Techniques.** Phd Thesis (PhD Thesis) — Faculty of Informatics. University of Wollongong., 2009. Citation on page 38.

ECCHER, C.; PURIN, B.; PISANELLI, D.; BATTAGLIA, M.; APOLLONI, I.; FORTI, S. Ontologies supporting continuity of care: The case of heart failure. **Computers in Biology and Medicine**, v. 36, n. 2006, p. 789–801, 2006. Citations on pages 39, 47, and 48.

ESTEFAN, J.; LASKEY, K.; BROWN, P.; MCCABE, F.; THORNTON, D. **Reference architecture foundation for service oriented architecture version 1.0**. Online., 2012. 1–118 p. Available: <https://www.oasis-open.org/committees/download.php/42644/GOLDSTANDARD-2011-06-16.pdf>. Citations on pages 10 and 13.

FEITOSA, D. **Simus - reference architecture for service multirobotics systems**. Phd Thesis (Master's Thesis.) — University of São Paulo, São Carlos, Brazil, 2013. Citation on page 14.

FERNANDEZ-MONTES, A.; ORTEGA, J.; ALVAREZ, J.; GONZALEZ-ABRIL, L. Smart environment software reference architecture. In: **INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on**. Seoul, South Korea: IEEE, 2009. p. 397–403. Citation on page 24.

FOUNDATION, D. U. **Diabetes UK. Know Diabetes, Fight Diabetes.** Wells Lawrence House, 126 Back Church Lane, London, UK., 2016. Online. Available: <https://www.diabetes.org.uk/>. Citation on page 130.

GAIKWAD, R.; WARREN, J. The role of home-based information and communications technology interventions in chronic disease management: a systematic literature review. **Health Informatics Journal**, v. 15, n. 2, p. 122–146, 2009. Citations on pages 35, 36, and 37.

GALSTER, M.; AVGERIOU, P. Empirically-grounded reference architectures. In: **Joint ACM SIGSOFT conference and symposium on Quality of Software Architectures and Architecting Critical Systems – (QoSA/ISARCS)**. Boulder, Colorado, USA: ACM Press, 2011. p. 153–157. ISBN 9781450307246. Citations on pages 13 and 14.

GARCÉS, L. **Log file to document results from the instantiation process of HomecARe reference architecture.** 2017. Online. Available: <https://goo.gl/Vnc3mj>. Citations on pages 127, 132, 149, 160, and 185.

GARCÉS, L.; AMPATZOGLOU, A.; AVGERIOU, P.; NAKAGAWA, E. Y. A comparative analysis of reference architectures for healthcare in the ambient assisted living domain. In: **2015 IEEE 28th International Symposium on Computer-Based Medical Systems**. Sao Carlos, SP, Brazil: IEEE, 2015. p. 270–275. ISSN 1063-7125. Citations on pages 5 and 24.

____. Quality attributes and quality models for ambient assisted living software systems: A systematic mapping. **Information and Software Technology**, v. 82, n. Supplement C, p. 121 – 138, 2017. ISSN 0950-5849. Available: <http://www.sciencedirect.com/science/article/pii/S0950584916302932>. Citations on pages 40 and 164.

GARCÉS, L.; AMPATZOGLOU, A.; OQUENDO, F.; NAKAGAWA, E. **Reference architectures and reference models for ambient assisted living systems: a systematic literature review.** Sao Carlos, SP, Brazil, 2017. Available: <http://repositorio.icmc.usp.br/handle/RIICMC/6564>. Citations on pages 5, 6, 24, and 29.

GARCÉS, L.; GRACIANO-NETO, V.; NAKAGAWA, E. Architectural solutions for self-adaptive systems. **Software Practice and Experience (to be submitted)**, p. 1–25, 2017. Citations on pages 17, 19, 20, and 22.

GARCÉS, L.; NAKAGAWA, E. Y. A process to establish, model and validate missions of systems-of-systems in reference architectures. In: **Proceedings of the Symposium on Applied Computing**. New York, NY, USA: ACM, 2017. (SAC '17), p. 1765–1772. ISBN 978-1-4503-4486-9. Available: <http://doi.acm.org/10.1145/3019612.3019799>. Citations on pages 17, 45, 46, 47, 52, 53, 56, 59, 164, and 167.

GARCÉS, L.; OQUENDO, F.; NAKAGAWA, E. **QM4AAL: A quality model for AAL systems.** Brazil., 2017. Available: <http://repositorio.icmc.usp.br/handle/RIICMC/6567>. Citations on pages 43 and 130.

GARCÉS, L.; OQUENDO, F.; NAKAGAWA, E. Y. A quality model for aal software systems. In: **2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)**. Belfast, Nothern Ireland, UK: IEEE, 2016. p. 175–180. Citations on pages 17, 41, 130, 164, and 186.

____. **A reference architecture for healthcare supportive home systems: missions establishment and validation.** Sao Carlos, SP, Brazil, 2017. Available: <http://repositorio.icmc.usp.br/handle/RIICMC/6566>. Citations on pages 17, 47, 52, 53, 54, 55, and 56.

____. **Towards a Taxonomy of Software Mediators for Systems-of-Systems**. 2018. 1–8 p. Citations on pages 23 and 203.

GARCÍA-MAGARIÑO, I.; GÓMEZ-SANZ, J. J. An ontological and agent-oriented modeling approach for the specification of intelligent ambient assisted living systems for parkinson patients. In: ____. **Hybrid Artificial Intelligent Systems: 8th International Conference, HAIS 2013,**

Salamanca, Spain, September 11-13, 2013. Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 11–20. ISBN 978-3-642-40846-5. Available: <https://doi.org/10.1007/978-3-642-40846-5_2>. Citations on pages 39, 47, and 48.

GARLAN, D.; BACHMANN, F.; IVERS, J.; STAFFORD, J.; BASS, L.; CLEMENTS, P.; MERSON, P. **Documenting Software Architectures: Views and Beyond**. Washington, DC, USA: Addison-Wesley Professional, 2010. Citation on page 12.

GARLAN, D.; BARNES, J. M.; SCHMERL, B.; CELIKU, O. Evolution styles: Foundations and tool support for software architecture evolution. In: **2009 Joint Working IEEE/IFIP Conference on Software Architecture European Conference on Software Architecture**. Cambridge, UK: IEEE, 2009. p. 131–140. Citation on page 20.

GARLAN, D.; PERRY, D. Software Architecture: Practice, Potential, and Pitfalls. In: **16th Conference on Software Engineering (ICSE)**. Sorrento, Italy: IEEE, 1994. p. 363–364. Citation on page 13.

GARLAND, J.; ANTHONY, R. **Large-Scale Software Architecture. A practical guide using UML**. 1. ed. England: John Wiley and Sons Ltd., 2003. 281 p. Citation on page 9.

GORTON, I. **Essential Software Architecture**. Second edition. Berlin: Springer-Verlag Berlin Heidelberg, 2011. Citations on pages 192, 193, and 194.

GUESSI, M.; BUENO, L.; NAKAGAWA, E. Y. Representation of reference architectures: A systematic review. In: **23nd International Conference on Software Engineering and Knowledge Engineering (SEKE)**. Miami Beach, United States: SEKE, 2011. p. 782–785. Citations on pages 15 and 165.

HANKE, S.; MAYER, C.; HOEFTBERGER, O. universAAL an open and consolidated AAL platform. **Ambient Assisted Living**, Springer, Berlin, Heidelberg, p. 127–140, 2011. Citation on page 25.

HARRISON, N. B.; AVGERIOU, P.; ZDUN, U. Using patterns to capture architectural decisions. **IEEE Software**, v. 24, n. 4, p. 38–45, 2007. Citation on page 17.

HAS. **Guides maladies chroniques et ALD**. 2014. Online. Available: <https://www.has-sante.fr//portail/jcms/c_1241701/fr/prise-en-charge-des-maladies-chroniques>. Citations on pages 40 and 48.

HAYES-ROTH, B.; PFLEGER, K.; LALANDA, P.; MORIGNOT, P.; BALABANOVIC, M. A Domain-Specific Software Architecture for Adaptive Intelligent Systems. **IEEE Transactions on Software Engineering**, v. 21, n. 4, p. 288–301, 1995. Citation on page 12.

HEALTH LEVEL SEVEN INTERNATIONAL. **Health Level Seven Standard Version 2.8.2 - An Application Protocol for Electronic Data Exchange in Healthcare Environments**. 2015. Citations on pages 23 and 116.

HEESCH, V. van; ELORANTA, V. P.; AVGERIOU, P.; KOSKIMIES, K.; HARRISON, N. Decision-centric architecture reviews. **IEEE Software**, v. 31, n. 1, p. 69–76, Jan 2014. ISSN 0740-7459. Citation on page 192.

HIETALA, H.; IKONEN, V.; KORHONEN, I.; LAHTEENMAKI, J.; MAKSIMAINEN, A.; PAKARINEN, V.; PARKKA, J.; SARANUMMI, N. **Feelgood - Ecosystem of PHR based products and services**. P.O.Box 1300, 33101 Tampere, Finland, 2009. Available: <http://feelgood.vtt.fi/FeelGood_finalreport.pdf>. Citation on page 24.

HOFMEISTER, C.; KRUCHTEN, P.; NORD R.L. OBBIK, H.; RAN, A.; AMERICA, P. Generalizing a model of software architecture design from five industrial apapproach. In: **Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture (WICSA5)**. Pittsburgh, PA, USA, USA: IEEE, 2005. Citations on pages 126, 160, 185, and 192.

HUCH, M. **D2: Identification and characterisation of the main stakeholder groups for "ICT for Ageing" solutions.** Online, 2010. Available: <http://www.supras.biz/pdf/Huch-M2010_StakeholderAnalysis-ICT-Aging.pdf.> Citations on pages 32 and 34.

IBM. **An architectural blueprint for autonomic computing. Third Edition.** Online, 2006. Citations on pages 21 and 22.

IDABC. **European interoperability framework for pan-European e-government services (EIF 1.0)**. Luxembourg: European Communities, 2004. Online. Available: <http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529>. Citation on page 39.

IEEE. **IEEE standard computer dictionary: a compilation of IEEE standard computer glossaries.** New York, 1990. Citation on page 38.

INGRAM, C.; PAYNE, R.; FITZGERALD, J. Architectural modeling patterns for systems of systems. In: **25th Annual INCOSE International Symposium (IS2015)**. Seattle: Wiley Online, 2015. p. 1–16. Citations on pages 17, 18, 19, and 20.

ISO/IEC. **International Standard. ISO/IEC FDIS 9126-1. Information technology - software product quality. Part 1: Quality model.** 2000. Citation on page 119.

____. **ISO/IEC 25010 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.** 2011. Available: <http://www.iso.org/iso/catalogue\detail.htm?csnumber=35733>. Citations on pages 41 and 42.

ISO/IEC/IEEE. **ISO/IEC/IEEE-42010 - Systems and software Engineering –Architecture Description.** 2011. Available: <http://www.iso-architecture.org/ieee-1471/ads/>. Citations on pages 13, 160, and 165.

ISSARNY, V.; BENNACEUR, A. Composing distributed systems: Overcoming the interoperability challenge. In: **Formal Methods for Components and Objects**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7866). p. 168–196. Citations on pages 23 and 203.

JOSUTTIS, N. **Soa in Practice: The Art of Distributed System Design**. Beijing: O'Reilly Media, Inc., 2007. ISBN 0596529554. Citations on pages 18 and 19.

KAMEAS, A.; CALEMIS, I. Pervasive systems in health care. In: NAKASHIMA, H.; AGHAJAN, H.; AUGUSTO, J. (Ed.). **Handbook of Ambient Intelligence and Smart Environments**. Springer US, 2010. p. 315–346. ISBN 978-0-387-93807-3. Available: <http://dx.doi.org/10.1007/978-0-387-93808-0_12>. Citation on page 24.

KEHAGIAS, D.; TZOVARAS, D.; MAVRIDOU, E.; KALOGIROU, K.; BECKER, M. Implementing an open reference architecture based on web service mining for the integration of distributed applications and multi-agent systems. In: CAO, L.; BAZZAN, A.; GORODETSKY, V.; MITKAS, P.; WEISS, G.; YU, P. (Ed.). **Agents and Data Mining Interaction**. Berlin Heidelberg: Springer Berlin Heidelberg, 2010, (Lecture Notes in Computer Science, v. 5980). p. 162–177. ISBN 978-3-642-15419-5. Citations on pages 5 and 25.

KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001.** UK, 2007. Citation on page 164.

KITSIOU, S.; PARÉ, G.; JAANA, M. Systematic reviews and meta-analyses of home telemonitoring interventions for patients with chronic diseases: A critical assessment of their methodological quality. **Journal of Medical Internet Research**, v. 15, n. 11, p. e253, 2013. Available: <http://www.jmir.org/2013/11/e253/>. Citations on pages 36 and 37.

KIWELEKAR, A. W.; JOSHI, R. K. Identifying architectural connectors through formal concept analysis of communication primitives. In: ____. **Software Architecture: 4th European Conference, ECSA 2010, Copenhagen, Denmark, August 23-26, 2010. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 515–518. ISBN 978-3-642-15114-9. Available: <https://doi.org/10.1007/978-3-642-15114-9_53>. Citations on pages 23 and 203.

KRUCHTEN, P. **The Rational Unified Process: An Introduction**. 2. ed. Boston, MA, USA: The Addison- Wesley Object Technology Series, Addison-Wesley, 2000. Citations on pages 9 and 15.

____. **The Rational Unified Process: An Introduction**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. Citation on page 12.

KUBICEK, H.; CIMANDER, R.; SCHOLL, H. J. Layers of interoperability. In: ____. **Organizational Interoperability in E-Government: Lessons from 77 European Good-Practice Cases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. chap. 7, p. 85–96. ISBN 978-3-642-22502-4. Available: <https://doi.org/10.1007/978-3-642-22502-4_7>. Citations on pages 38, 39, and 40.

KURSCHL, W.; MITSCH, S.; SCHOENBOECK, J. An engineering toolbox to build situation aware ambient assisted living systems. In: **Broadband Communications, Information Technology Biomedical Applications, 2008 Third International Conference on**. Gauteng, South Africa: IEEE, 2008. p. 110–116. Citation on page 24.

LAMSWEERDE, A. van. Goal-oriented requirement engineering: A guided tour. In: **Proceedings of the 5th IEEE International Symposium on Requirements Engineering.** Toronto, Ontario, Canada: IEEE, 2001. p. 249–263. Citations on pages 46 and 164.

LAU, K. K.; VELASCO, P.; ZHENG, W. Exogenous connectors for software components. In: ____. **Component-Based Software Engineering: 8th International Symposium, CBSE 2005, St. Louis, MO, USA, May 14-15, 2005. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 90–106. ISBN 978-3-540-32049-4. Available: <https://doi.org/10.1007/11424529_7>. Citations on pages 23 and 203.

LI, X.; FAN, Y.; WANG, J.; WANG, L.; JIANG, F. A pattern-based approach to development of service mediators for protocol mediation. In: **Seventh Working IEEE/IFIP Conference on**

Software Architecture (WICSA 2008). Vancouver, BC: IEEE, 2008. p. 137–146. Citations on pages 23 and 203.

LIU, J.-S.; WANG, B.; LIAO, H.; HUANG, C.; SHIH, C.; KUO, T.; PANG, A. Reference architecture of intelligent appliances for the elderly. In: **Systems Engineering, 2005. ICSEng 2005. 18th International Conference on**. Las Vegas, NV, USA, USA: IEEE, 2005. p. 447–455. Citation on page 24.

LOPES, A.; WERMELINGER, M.; FIADEIRO, J. L. Higher-order architectural connectors. **In ACM Transactions on Software Engineering and Methodology**, v. 12, n. 1, p. 64–104, 2003. Citations on pages 23 and 203.

LOSAVIO, F.; ORDAZ, O.; ESTELLER, V. Quality-based bottom-up design of reference architecture applied to healthcare integrated information systems. In: **2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)**. Athens, Greece: IEEE, 2015. p. 70–75. ISSN 2151-1349. Citations on pages 25 and 28.

MACKENZIE, M.; LASKEY, K.; MCCABE, F.; BROWN, P.; METZ, R.; HAMILTON, B. E. **OASIS: Reference Model for Service Oriented Architecture 1.0**. Online, 2006. 1–31 p. Citation on page 11.

MAEDER, A.; WILLIAMS, P. Precision healthcare through informatics. In: ____. Xiamen, China: 2017 International Medical Informatics Assiciation (IMIA) and IOS Press, 2017. (Studies in Health Technology and Informatics, 245), chap. Health Smart Homes: New Challenges, p. 166–169. Citations on pages 3, 4, and 5.

MAES, P. Concepts and experiments in computational reflection. In: **Conference Proceedings on Object-oriented Programming Systems, Languages and Applications**. New York, NY, USA: ACM, 1987. (OOPSLA '87), p. 147–155. ISBN 0-89791-247-0. Available: <http://doi.acm.org/10.1145/38765.38821>. Citation on page 21.

MAHEU, M.; WHITTEN, P.; ALLEN, A. **E-Health, Telehealth, and Telemedicine: A Guide to Startup and Success.** San Francisco: Jossey-Bass, 2001. 400 p. Citation on page 27.

MAIER, M. W. Architecting principles for system-of-systems. **Systems Engineering**, v. 1, p. 267–284, 1999. Citations on pages 16 and 17.

MAJUMDER, S.; AGHAYI, E.; NOFERESTI, M.; MEMARZADEH-TEHRAN, H.; MONDAL, T.; PANG, Z.; DEEN, M. Smart homes for elderly healthcare-recent advances and research challenges. **Sensors (Basel).**, v. 17, n. 11, 2496, p. 2–32, 2017. Citations on pages 3, 4, and 5.

MARTÍNEZ-FERNÁNDEZ, S.; AYALA, C.; FRANCH, X.; MARQUES, H. M. Benefits and Drawbacks of Reference Architectures. In: DRIRA, K. (Ed.). **7th European Conference on Software Architecture (ECSA).** Montpellier, France: Springer Berlin Heidelberg, 2013. p. 307–310. Citation on page 5.

MCLAIN, K.; O'HARA-LESLIE, E.; WADE, A. **Foundations for Assisting in Home Care**. Open Suny Textbooks, 2016. Online. Available: <https://milnepublishing.geneseo.edu/home-health-aide>. Citations on pages 31, 32, 33, and 34.

MEDICARE. **What's home health care?** 2017. Available: <https://www.medicare.gov/what-medicare-covers/home-health-care/home-health-care-what-is-it-what-to-expect.html>. Citation on page 31.

MEHTA, N. R.; MEDVIDOVIC, N.; PHADKE, S. Towards a taxonomy of software connectors. In: **22nd international conference on Software engineering**. Limerick, Ireland: ACM, 2000. p. 178 – 187. Citations on pages 23 and 203.

MEMON, M.; WAGNER, S. R.; PEDERSEN, C. F.; BEEVI, F. H. A.; HANSEN, F. O. Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. **Sensors (Basel, Switzerland)**, v. 14, n. 3, p. 4312–4341, Jan. 2014. ISSN 1424-8220. Citation on page 3.

MERIDOU, D.; KAPSALIS, A.; KASNESIS, P.; PATRIKAKIS, C.; VENIERIS, I.; KAKLAMANI, D.-T. An event-driven health service bus. In: **Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare**. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015. (MOBIHEALTH'15), p. 267–271. ISBN 978-1-63190-088-4. Available: <http://dx.doi.org/10.4108/eai.14-10-2015.2261684>. Citation on page 165.

MINSALUD. **Guías de prácticas clínicas. Manejo de enfermedades crónicas no transmisibles.** 2009. Online. Available: <http://eps.coomeva.com.co/publicaciones.php?id=44404>. Citations on pages 40 and 48.

MINSAUDE. **ESTRATEGIAS PARA O CUIDADO DA PESSOA COM DOENÇA CRONICA**. Ii. 160p. Brasilia: Departamento de Atenção Básica. – Brasília : Ministério da Saúde., 2013. Citations on pages 40 and 48.

MITTAL, S.; RAINEY, L. Harnessing emergence: The control and design of emergent behavior in system of systems engineering. In: **Proceedings of the Conference on Summer Computer Simulation (SCSC)**. Chicago, Illinois: Society for Computer Simulation International, 2015. p. 1–10. Citations on pages 17, 54, and 55.

MULLER, G. **A Reference Architecture Primer**. Online, 2012. 1–20 p. Available: <http://www.gaudisite.nl/info/ReferenceArchitecturePrimer.info.html>. Citation on page 10.

MULLER, G.; LAAR, P. Right sizing reference architectures - how to provide specific guidance with limited information. In: **18th Annual International Symposium of the International Council on Systems Engineering (INCOSE)**. Utrecht, Netherlands: Wiley Online Library, 2008. p. 1–8. Citation on page 13.

MULLER, H. A.; KIENLE, H. M.; STEGE, U. Autonomic computing now you see it, now you don't. In: LUCIA, A.; FERRUCI, F. (Ed.). **Software Engineering**. Berlin: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Sciences, v. 5413). p. 32–54. Citations on pages 21 and 22.

NAKAGAWA, E.; OLIVEIRA, A. P.; BECKER, M. Reference Architecture and Product Line Architecture: A Subtle But Critical Difference. In: CRNKOVIC, I.; GRUHN, V.; BOOK, M. (Ed.). **Software Architecture**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6903). p. 207–211. ISBN 978-3-642-23797-3. Citation on page 12.

NAKAGAWA, E. Y.; AO, A. S. S.; FERRARI, F.; MALDONADO, J. C. Towards a Reference Architecture for Software Testing Tools. In: **International Conference on Software Engineering and Knowledge Engineering (SEKE)**. Boston, USA: SEKE, 2007. p. 1–6. Citation on page 14.

NAKAGAWA, E. Y.; BARBOSA, E. F.; MALDONADO, J. C. Exploring ontologies to support the establishment of reference architectures: An example on software testing. In: **Joint Working IEEE/IFIP Conference on Software Architecture, European Conference on Software Architecture (WICSA/ECSA).** Cambridge, U.K.: IEEE, 2009. p. 249–252. Citation on page 11.

NAKAGAWA, E. Y.; FERRARI, F. C.; SASAKI, M.; MALDONADO, J. C. An aspect-oriented reference architecture for Software Engineering Environments. **Journal of Systems and Software**, Elsevier Inc., v. 84, n. 10, p. 1670–1684, Oct. 2011. ISSN 01641212. Citation on page 14.

NAKAGAWA, E. Y.; GUESSI, M.; MALDONADO, J. C.; FEITOSA, D.; OQUENDO, F. Consolidating a Process for the Design, Representation, and Evaluation of Reference Architectures. In: **10th Working IEEE/IFIP Conference on Software Architecture (WICSA)**. Sydney, Australia: IEEE/IFIP, 2014. p. 143–152. Citations on pages 17, 7, 14, and 164.

NAKAGAWA, E. Y.; OQUENDO, F.; BECKER, M. RAModel: A Reference Model for Reference Architectures. In: **2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture (WICSA/ECSA)**. Helsinki, Finland: IEEE, 2012. p. 297–301. ISBN 978-0-7695-4827-2. Citations on pages 13 and 15.

NAKAGAWA, E. Y.; OQUENDO, F.; MALDONADO, J. C. Reference Architectures. In: OUSSALAH, M. O. (Ed.). **Software Architecture: Principles, Techniques, and Tools**. 1st. ed. London: Wiley, 2014. chap. 2, p. 55—-82. Citations on pages 5, 10, 11, 13, and 125.

National Council on Aging. **Center for Healthy Aging.** 2014. Online. <http://www.ncoa.org/improve-health/center-for-healthy-aging/chronic-disease/>. Accessed in May 5th, 2014. Citation on page 1.

NAVARRO, R. F.; RODRÍGUEZ, M. D.; FAVELA, J. Intervention tailoring in aal systems for elders with dementia using ontologies. In: ____. **Ambient Assisted Living and Home Care: 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 338–345. ISBN 978-3-642-35395-6. Available: <https://doi.org/10.1007/978-3-642-35395-6_46>. Citations on pages 39, 47, and 48.

NIELSEN, C.; LARSEN, P. G.; FITZGERALD, J.; WOODCOCK J;. PELESKA, J. System of systems engineering: Basic concepts, model-based techniques, and research directions. **ACM Computing Surveys**, v. 48, n. 2, p. 1–41, 2015. Citation on page 16.

NITZSCHE, T.; THIELE, S.; A, H.; A., W. Communication architecture for aal. supporting patient care by health care providers in aal-enhanced living quarters. **Methods of Information in Medicine**, v. 53, n. 3, p. 167–172, 2014. Citations on pages 25 and 28.

NOURY, N.; VIRONE, G.; BARRALON, P.; YE, J.; RIALLE, V.; DEMONGEOT, J. New Trends in Health Smart Homes. In: **5th International Workshop on Enterprise Networking and Computing in Healthcare Industry. (Healthcom)**. Santa Monica, CA, USA.: IEEE, 2003. p. 118–127. Citations on pages 3 and 34.

OLIVEIRA, L. B. R.; NAKAGAWA, E. Y. A service-oriented reference architecture for software testing tools. In: **5th European Conference on Software Architecture (ECSA)**. Essen, Germany: Springer-Verlag, 2011. p. 405–421. ISBN 978-3-642-23797-3. Citation on page 14.

O'MALLEY, A.; RESCHOVSKY, J. Referral and consultation communication between primary care and specialist physicians: Finding common ground. **Archives of Internal Medicine**, v. 171, n. 1, p. 56–65, 2011. Available: <+http://dx.doi.org/10.1001/archinternmed.2010.480>. Citation on page 40.

OPENEHR. **openEHR Architecture Overview**. 2015. Citation on page 25.

OQUENDO, F. Formally describing the software architecture of systems-of-systems with sosadl. In: **11th System of Systems Engineering Conference (SoSE)**. Kongsberg: IEEE, 2016. p. 1–6. Citation on page 22.

OUSD(AT&L), DoD. **Systems and Software Engineering. Systems Engineering Guide for Systems of Systems.** Online, 2008. Available: <https://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>. Citation on page 17.

PARÉ, G.; CAFAZZO, J.; KASTNER, P.; SETO, E. Clinical effects of home telemonitoring in the context of diabetes, asthma, heart failure and hypertension: A systematic review. **Journal of Medical Internet Research.**, v. 12, n. 2, p. 1–20, 2010. Citations on pages 35, 36, and 37.

PIEPER, M.; ANTONA, M.; CORTÉS, U. Ambient Assited Living. **Ercim News**, n. 87, p. 18–19, 2011. Citation on page 3.

QUEENSLAND-GOVERMENT. **Chronic Conditions Manual:Prevention and Management of Chronic Conditions in Australia.** Queensland, AU., 2015. Citations on pages 40 and 48.

REED, P. **Reference Architecture: The best of best practices**. 2002. Online. <http://www.ibm.com/developerworks/rational/library/2774.html>. Accessed in May 14th, 2014. Citation on page 9.

REEDER, B.; MEYER, E.; LAZAR, A.; CHAUDHURI, S.; THOMPSON, H.; DEMIRIS, G. Framing the evidence for health smart homes and home-based consumer health technologies as a public health intervention for independent aging: A systematic review. **International Journal of Medical Informatics**, v. 82, n. 2013, p. 565–579, 2013. Citations on pages 35, 36, and 37.

RIALLE, V.; DUCHENE, F.; NOURY, N.; BAJOLLE, L.; DEMONGEOT, J. Health "Smart" Home: Information Technology for Patients at Home. **Telemedicine Journal and e-Health.**, v. 8, n. 4, p. 395–409, 2002. Citations on pages 3 and 34.

ROMAY, M. P.; CUESTA, C. E.; FERNÁNDEZ-SANZ, L. O self-adaptation in systems-of-systems. In: **Proceedings of the 1st International Workshop on Software Engineering for Systems-of-Systems**. Montpellier, France: IEEE, 2013. p. 29–34. Citations on pages 17 and 20.

ROSEN, M.; AMBLER, S.; HAZRA, T.; ULRICH, W.; WATSON, J. Enterprise Architecture Trend. **Enterprise Architecture**, v. 10, n. 1, 2007. Citation on page 10.

ROTHENHAUS, K. J.; MICHAEL, J. B.; T., S. M. Architectural patterns and auto-fusion process for automated multisensor fusion in soa system-of-systems. **IEEE SYSTEMS JOURNAL**, v. 3, n. 3, p. 304–316, 2009. Citations on pages 17 and 20.

RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. **Empirical Software Engineering**, v. 14, n. 2, p. 131, Dec 2008. ISSN 1573-7616. Available: <https://doi.org/10.1007/s10664-008-9102-8>. Citations on pages 125, 129, and 159.

RUNESON, P.; HOST, M.; RAINER, A.; REGNELL, B. **Case Study Research in Software Engineering: Guidelines and Examples**. First. USA: Wiley Publishing, 2012. Citations on pages 159, 160, and 161.

RYAN, A.; EKLUND, P. The health service bus: an architecture and case study in achieving interoperability in healthcare. **Studies in Health Technology and Informatics**, v. 160, n. Pt2, p. 922–926, 2010. Citation on page 165.

SALEHIE, M.; TAHVILDARI, L. Self-adaptive software: Landscape and research challenges. **ACM Trans. Auton. Adapt. Syst.**, ACM, New York, NY, USA, v. 4, n. 2, p. 14:1–14:42, May 2009. ISSN 1556-4665. Available: <http://doi.acm.org/10.1145/1516533.1516538>. Citation on page 42.

SAMARIN, A. **Smart Homes as Systems-of-Systems, a Reference Architecture**. Online: [s.n.], 2016. Available: <https://www.linkedin.com/pulse/smart-home-system-of-systems-reference-architecture-alexander-samarin/>. Citations on pages 25 and 28.

SBD. **Diretrizes da Sociedade Brasileira de Diabetes: 2015-2016**. São Paulo: AC Farmacêutica., 2016. Onile. Retrieved November 2, 2016. Available: <http://www.diabetes.org.br/sbdonline/images/docs/DIRETRIZES-SBD-2015-2016.pdf>. Citation on page 130.

SIDDIQUI, B. **Integrating healthcare services, Part 1: Using an Enterprise Service Bus for healthcare. Configure a Java Business Integration server to make diverse applications interconnect and interoperate.** Online, 2010. Available: <https://www.ibm.com/developerworks/library/j-hsb1/index.html>. Citations on pages 165 and 168.

SILVA, E.; BATISTA, T.; OQUENDO, F. A mission-oriented approach for designing system-of-systems. In: **10th System of Systems Engineering Conference (SoSE)**. San Antonio, TX, USA: IEEE, 2015. p. 346–351. Citations on pages 46 and 164.

SIMONET, M.; MESSAI, R.; DIALLO, G.; SIMONET, A. Ontologies in the health field. In: BERKA, P.; RAUCH, J.; ZIGHED, D. A. (Ed.). **Data Mining and Medical Knowledge Management: Cases and Applications**. Bordeaux: Medical Information Science Reference, 2009. p. 37–56. Citations on pages 39, 47, and 48.

Software Engineering Institute. **Glossary of Software Architecture**. 2014. Online. <http://www.sei.cmu.edu/architecture/start/glossary/index.cfm>. Accessed in June 2nd, 2014. Citation on page 13.

SPALAZZESE, R.; INVERARDI, P.; ISSARNY, V. Towards a formalization of mediating connectors for on the fly interoperability. In: **Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture**. Cambridge: IEEE, 2009. p. 345–348. Citations on pages 23 and 203.

STRUCKMANN, V.; DYAKOVA, M.; GABRIELLA, M.; RISSANEN, S.; GINNEKEN, E. V. ICARE4EU: Improving care for people with multiple chronic conditions in Europe. **Eurohealth International**, v. 19, n. 3, p. 29–31, 2013. Citations on pages 1 and 2.

TAO, H.; MCROY, S. Caring for and keeping the elderly in their homes. **Chinese Nursing Research**, v. 2, n. 2, p. 31 – 34, 2015. ISSN 2095-7718. Available: <http://www.sciencedirect.com/science/article/pii/S2095771815300013>. Citations on pages 32 and 33.

TAZARI, M.-R.; FURFARI, F.; RAMOS, J.-P. L.; FERRO, E. The PERSONA Service Platform for AAL Spaces. **Handbook of Ambient Intelligence and Smart Environments**, Springer US, Boston, MA, p. 1171–1199, 2010. Citation on page 5.

TAZARI, S.; VALERO, A. F.; DOMMARCO, R.; RAMOS, J. P. L.; FURFARI, F. **PERSONA PERceptive Spaces prOmoting iNdependent Aging. Final Reference Architecture model for AAL and recommendations for future activities on the Open AAL Platform.** Germany, 2010. Available: <http://www.aal-persona.org>. Citation on page 25.

TOMSON, T.; PREDEN, J. Simulating system of systems using mace. In: **2013 UKSim 15th International Conference on Computer Modelling and Simulation**. Cambridge, UK: IEEE, 2013. p. 155–160. Citations on pages 23 and 203.

TOUMPANIARIS, P.; ILIOPOULOU, D.; LAZAKIDOU, A.; KATEVAS, N.; KOUTSOURIS, D. A survey for chronic diseases management and the related sensors in the ambient assisted living environment. **International Journal of Health Research and Innovation**, v. 2, n. 1, p. 65–84, 2014. Citation on page 37.

TRACZ, W. DSSA (Domain-Specific Software Architecture) Pedagogical Example. **SIGSOFT Software Engineering Notes**, v. 20, n. 3, p. 49—-62, 1995. Citation on page 12.

TRIENEKENS, J.; ANGELOV, S.; GREFEN, P.; KUSTERS, R. Quality of software reference architectures. In: POWELL, P.; NUNES, M.; ISAIAS, P. (Ed.). **IADIS International Conference Information Systems (IS)**. 11-13 March, Avila, Spain: Publications TUe, 2011. p. 145–151. ISBN 9789728939472. Citation on page 13.

TUOMAINEN, M.; MIKKANEN, J. Reference architecture of application services for personal wellbeing information management. In: MOEN, A. e. a. (Ed.). **Studies in Health Technology and Informatics**. Online: IOS Press Ebooks., 2011. User Centred Networked Health Care, n. 169, p. 98–102. Available: <http://ebooks.iospress.nl/publication/14134>. Citation on page 25.

United Nations. **World Population Prospects. The 2017 Revision. Key Findings and Advance Tables.** New York, 2017. Citation on page 1.

U.S. Department of Health and Human Services. **HHS Initiative on Multiple Chronic Conditions**. 2014. Online. <http://www.hhs.gov/ash/initiatives/mcc/>. Accessed in June 10th, 2014. Citation on page 1.

VEER, H. van der; WILES, A. **European Telecommunications Standards Institute: achieving technical interoperability – the ETSI approach.** France, 2006. Citation on page 39.

VICENTE, I.; GARCÉS, L.; NAKAGAWA, E. Y. Establishment of a software architecture for health care supportive home system to assist patients with diabetes mellitus. In: **25o Simpósisio Internacional de Iniciação Científica da USP - SIICUSP**. Sao Carlos, SP, Brazil: ICMC - USP, 2017. p. 1–1. Citation on page 125.

____. **Establishment of a Software Architecture for Health Care Supportive Home System to Assist Patients with Diabetes Mellitus: Functional and Non-Functional Requirement.** Sao Carlos, SP, Brazil, 2017. Citations on pages 128, 130, 131, 138, and 139.

____. **Guidelines for interoperability in e-Health reference architectures**. Sao Carlos, SP, Brazil, 2017. Citations on pages 117 and 190.

WARTENA, F.; MUSKENS, J.; SCHMITT, L.; PETKOVIC, M. Continua: The reference architecture of a personal telehealth ecosystem. In: **12th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)**. Lyon, France: IEEE, 2010. ISBN 978-1-4244-6374-9. Citations on pages 17, 2, 3, 5, 25, and 35.

WEYNS, D.; AHMAD, T. Claims and evidence for architecture-based self-adaptation: A systematic literature review. In: **7th European conference on Software Architecture**. Berlin Heidelberg: Springer Berlin Heidelberg, 2013. p. 249–265. Citation on page 22.

WEYNS, D.; SCHMERL, B.; GRASSI, V.; MALEK, S.; MIRANDOLA, R.; PREHOFER, C.; WUTTKE, J.; ANDERSSON, J.; GIESE, H.; GOSCHKA, K. On patterns for decentralized control in self-adaptive systems. In: LEMOS, R. de; GIESE, H.; MULLER, H.; SHAW, M. (Ed.). **Software Engineering for Self-Adaptive Systems II**. Berlin Heidelberg: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Sciences, v. 7475). p. 76–107. Citation on page 22.

WHO. **Health Information Systems**. Online, 2008. Available: <http://www.who.int/healthinfo/statistics/toolkit_hss/EN_PDF_Toolkit_HSS_InformationSystems.pdf>. Citation on page 27.

____. **ICD 10 - International Statistical Classification of Diseases and Related Health Problems 10th Revision**. Online, 2016. Available: <http://apps.who.int/classifications/icd10/browse/2016/en>. Citation on page 39.

ZEINALI, N.; ASOSHEH, A.; SETAREH, S. The conceptual model to solve the problem of interoperability in health information systems. In: **2016 8th International Symposium on Telecommunications (IST)**. Tehran, Iran: IEEE, 2016. p. 684–689. Citation on page 165.

ZIMMERMAN, O. Architectural decisions as reusable design assets. **IEEE Software**, v. 28, n. 1, p. 64–69, 2011. Citation on page 17.

APPENDIX

A

GUIDELINES TO USE *HOMECARE*

This appendix presents some guidelines that orient the systematic use of the knowledge contained in *HomecARe* , such as those presented in the top of Figure 87,that is structured in this thesis, when the software architecture of a specific HSH system is intended to be developed.

As proposed by Hofmeister et al. (HOFMEISTER *et al.*, 2005), the architecture design of a software system is principally composed of three activities: (i) *Architectural analysis*, that aims the definition of problems, expressed as architecturally significant requirements (ASRs), that the architecture must solve based on concerns (i.e., quality attributes requirements, mandated standards or business goals) and context (i.e., functionalities desired by stakeholders); (ii) *Architectural synthesis*, that intends to establish architectural solutions to solve the ASRs, detailing the rational behind the selection of proposed solutions; and (iii) *Architectural evaluation*, whose purpose is to validate or invalidate if proposed architectural solutions are correct to achieve the ASRs. Figure 87 details how *HomecARe* supports all activities of architectural design of a HSH system. To assist the architect in the conduction of the process presented in Figure 87, a spreadsheet was made and can be consulted in (GARCÉS, 2017)[1].

It is important to highlight that the process in Figure 87 is iterative, that is, all knowledge and models contained in *HomecARe* could be updated as such reference architecture is being used to create HSH systems. For instance, new requirements of quality attributes identified when a HSH system is analysed can be included in the quality model contained in *HomecARe* . Similarly, activities depend of new knowledge obtained during the architectural design, thus, outputs of each activity can change over architectural design iterations. Each activity is detailed as follows.

[1] Spreadsheet supporting activities of this process are available in <https://goo.gl/Vnc3mj>

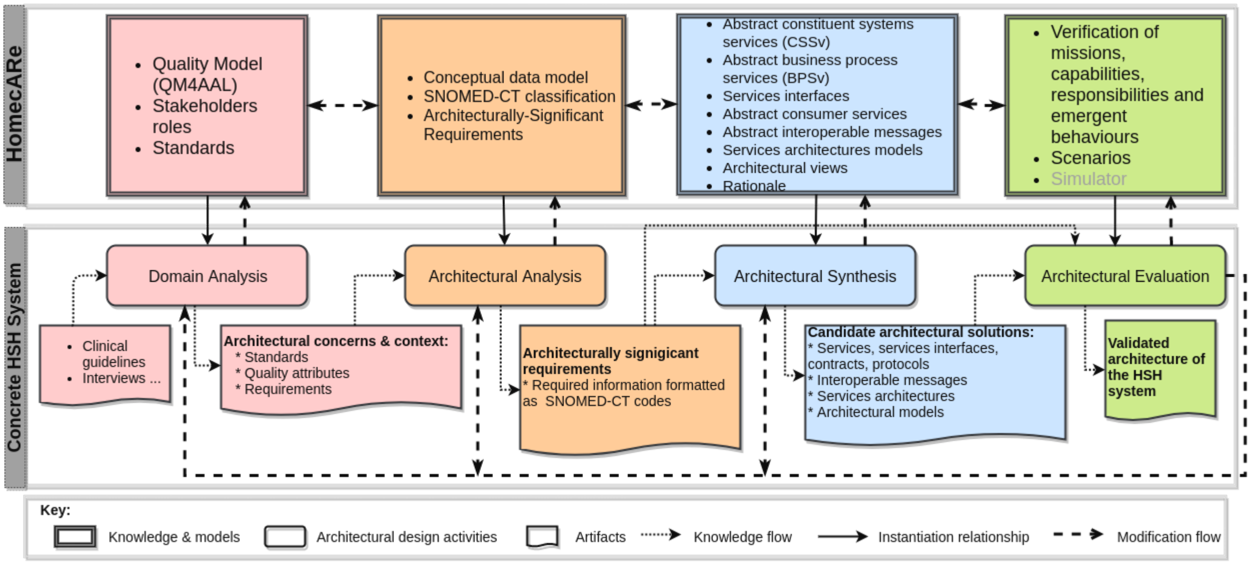


Figure 87 – Architectural design of a HSH system using *HomecARe* .

Activity 1 - Domain analysis

As a preliminary activity a domain analysis is required before starting an architectural design, in order to identify context and concerns required to conduct the architectural analysis. Hence, firstly, it is important to define which kind of health condition (e.g., chronic disease, disability, or co-morbidities) the HSH systems intends to support at home. When an initial definition about the HSH system scope is made, (e.g., the HSH system is oriented to assist cardiovascular disease patients in their treatments at home), information about how to manage such conditions must be identified, perhaps using clinical guidelines or expert knowledge obtained through interviews. Clinical guidelines are an important source of information, since they offer work-flow that must be achieved by the home healthcare team to support the condition, and are specific of country legislations.

To guide the domain analysis where the HSH system is framed in, *HomecARe* offers important domain knowledge (in Chapter 3). Therefore, for the specific HSH system, the architect can use stakeholders definition (in Section 3.1), interoperability standards in healthcare domain (in Section 3.3), and quality attributes requirements defined in the quality model (QM4AAL in (GARCÉS; OQUENDO; NAKAGAWA, 2016)), and adapt such domain knowledge according to clinical guidelines and specific stakeholders interests.

As result of this phase, concerns and context specific for the HSH system under design are identified and described as system's requirements, i.e., functional and non-functional requirements.

Activity 2 - Architectural analysis

This activity aims to delineate the problem that the architecture must solve. *HomecARe* contains important knowledge that can be used by the architect to understand better the problem and to define, with less effort, the architecturally significant requirements of her/his HSH system.

Initially, from requirements identified in last phase, types of information that must be managed by the HSH system must be characterized, using the SNOMED-CT classification. To facilitate the selection of which information types must be considered in the HSH system, the architect can make use of Table 17 jointly with a SNOMED-CT data base, such as the *SNOMED International SNOMED CT Browser* [2].

Specifically, Table 17 summarizes which parent codes of SNOMED CT must be considered in HSH systems. Such codes were defined during the construction of *HomecARe* . Based on requirements defined in last phase, the architect or analyst can specify more concretely, which children codes of SNOMED CT are required to the HSH system. To consult all children for each parent codes, the *SNOMED CT Browser* can be used.

Table 17 – SNOMED CT codes defined in *HomecARe*

Description	**Parent codes in SNOMED-CT ID**	**Children codes list**
Measures on patient vital signs, such as, body temperature (BT), blood pressure (BP), pulse or heart rate (HR), and respiration rate (RR).	363789004 - General characteristic of patient (observable entity).	
Measures on patient cardiovascular functions, such as, heart sound, cardiac flow or systole function.	70337006 - Cardiovascular function (observable entity).	
Information on abdominal findings such as, abdominal aorta, mass or rigidity.	609624008 - Finding of abdomen; 249273002 - Finding of urinary tract.	
Measures on patient involuntary movements, such as, excessive blinking, tremors or spasms.	267078001 - Involuntary movement finding.	
Measures on patient varicose or skin ulcer.	271652003 - Varicose vein finding.	
Measures on patient skin ulcer, such as, pressure ulcer stage, surface area of ulcer.	439744001 - Ulcer observable (observable entity).	
Measures on findings in systemic arterial, such as, arterial bruit, carotid bruit or pulse.	301139003 - Systemic arterial finding; 54718008 - Peripheral pulse, function (observable entity)	
Measures regarding abnormal breathing, lung capacity or respiration difficulty.	106048009 - Respiratory finding.	
Information regarding patient activities in social context, such as, shopping, reading, using telephone.	300574001 - Community living activity (observable entity).	
Information on personal care activities, such as, dressing, personal hygiene, or taking medications.	285592006 - Personal care activity (observable entity)	
Information on domestic activities, such as, doing housework or preparing meals.	272387007 - Domestic activity (observable entity).	

Continued on next page

[2] <http://browser.ihtsdotools.org>

Table 17 – *Continued from previous page*

Description	**Parent codes in SNOMED-CT ID**	**Children codes list**
Measurements on patient environment, such as odour, infestation, water supply	224153006 - Local environment and neighbourhood details (observable entity).	
Information about patient physical status, through remote patient physical examination,and monitoring her/his sign and symptoms.	404966002 - Physical ageing status (observable entity)	
Information about AoDL profile, detailing the status of patient environment and his/er social, personal care, domestic and leisure activities.	370885003 - Activities of daily living management	
Information about the status of patient rehabilitation regarding her/his care plan.	722138006 - Physiotherapy	
Information about the status of patient pharmacological treatment regarding her/his medication plan.	416608005 - Drug therapy.	
Information about the status of patient diet, regarding her/his nutrition plan.	185495006 - In-house dietetics.	
Information about the situation of patient position regarding her/his position intervention plan.	225430005 - Procedures relating to mobility.	
Information about the situation of a patient health condition based on her/his physical status, sign and symptoms, and intervention profile.	405157008 - Personal health status (observable entity)	
Information about the patient intervention profile, considering information of patient rehabilitation, and pharmacological and non-pharmacological interventions.	386053000 - Evaluation procedure	
Supporting to patient at reminding events needed to follow correctly her/his interventions and care plans, such as, reminding about keeping appointment, performing activity, performing procedure or taking a drug.	223452003 - Reminding (procedure)	
Establishment of patient situation regarding her/his quality of life, considering informations from patient AoDL profile and health conditions.	709503007 - Assessment of quality of life (procedure)	
Consolidation of the patient formal care profile that can be analysed by care team for assessing its adequacy to patient situations.	392134007 - Manage health care	
Prediction of emergency situations and establishment of an emergency plan to be executed.	281694009 - Finding of a risk; 225314003 - Risk management.	
Establishment of patient environment situation and definition of an environment intervention plan.	224249004 - Characteristics of home environment ; 129841008 - Finding related to environmental risk factor	

As result of the architectural analysis, a definition of the architecturally significant requirements (ASRs) for the software architecture of the concrete HSH system must be done. ASRs should contain the healthcare information types for the HSH system, and such requirements can be completed refining the ASRs of *HomecARe* (in Section 4.2).

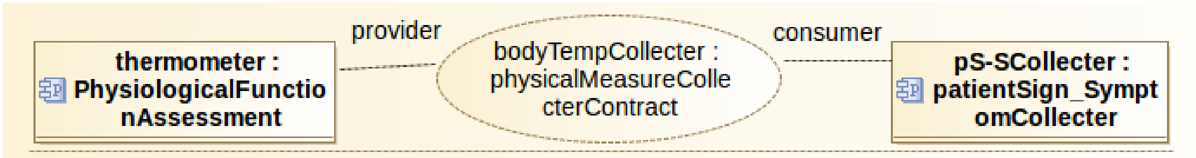


Figure 88 – Example of services instances in a services architecture of a BPSv.

Activity 3 - Architectural synthesis

In this phase, services architectures for the specific HSH system are proposed. As first step, each healthcare information type (represented by children codes of SNOMED CT) identified in last phase must be provided by a respective service, i.e., constituent system service (CSSv) or business process service (BPSv). For this, information contained in the Conceptual viewpoint of *HomecARe* can be used, specifically, Tables 8 and 10, that correspond, respectively, to *Services Layer* and *Business Process Layer* descriptions. Hence, for managing each information type, represented as a children code of SNOMED CT, an instance of the respective provider, i.e., CSSv or BPSv, must be allocated.

An example of this procedure is presented as follows. Whether the architect / analyst establishes that the patient vital sign `body temperature` with, `386725007` as a child code of `363789004 - General characteristic of patient` in SNOMED CT, is relevant to the HSH system, then, an instance of the service `CSSv001 - Physiological function assessment`, named for instance, `CSSv001-1 Thermometer`, must be created to provide such vital sign.

Moreover, depending of the identified BPSv, the architect must select the services architectures that realize each BPSv relevant for her/his HSH system. Table 18 details diagrams associated to each BPSv in *HomecARe* . Links to services architectures diagrams related to BPSv are presented in last column. More information about each services architecture is presented in the *Mission Viewpoint* in Section 5.2.

In sequence, consumers of each healthcare information type must be identified into a respective services architecture. Moreover, contracts allowing the exchange of each information type between providers and consumers services must be characterized. Services architectures diagrams and models in Section 5.2 can orient this tasks.

Moreover, each selected services architecture must fit the specificities of the particular HSH system. Instances of services interfaces contracts are created for each concrete service participating of the architecture of the HSH system. Figure 88 depicts an example of how the concrete `CSSv001-1 Thermometer`, an instance of the service `CSSv001 - Physiological function assessment` , provides the vital sign *body temperature* to the `pS-SCollecter`, an instance of the `patientSign-SymptomCollecter`, that in turns is an instance of the control service `CSv002- Collector service`. Communication between both services, the `thermometer`

```
MSH|^~\&|^CS007X|   THERMOMETER              |DIAMANT@HOME^HSH001||20171010225526
||ORU^R01|0000001|P|2.8||||||||||||
PID|||||Garcés^Lina||||||&Av. Trabalhador Sao Carlense&400^^SAO CARLOS^SP^13566-590^^H|
||||||||||||||||||||||||||||
OBR||||56342008^temperature taking^SNM3|||20171010225525|||||||||||||||20171010225526
|||F|||||||||||||||||||||||||||||
OBX||NM|386725007^body temperature^SNM3||36.5|258710007^Celsius^SNM3||||F|||
20171010225525|||56342008^temperature taking^SNM3|||||||||||||
```

Figure 89 – Example of message from a service providing *body temperature* information

and `pS-SCollecter` is made using an instance, `bodyTempCollecter`, of the contract `physicalMeasureCollecterContract`. Both services participate of the service architecture related to `BPSv001 - Patient status and sign & symptoms monitor`, introduced in Section 5.2.

To allow the communication and interoperation through messages by the Health Service Bus (HSB), standardized messages contained healthcare information relevant for the HSH system must be defined. Hence, services must provide their capabilities using standardized messages as those presented in Section 5.3. Messages can be constructed based on the guidelines offered in (VICENTE; GARCÉS; NAKAGAWA, 2017c). Figure 89 illustrates a possible message that the `Thermometer` must use to communicate the patient's vital sign `body temperature`.

After selecting all consumers and providers services, the exchanged healthcare information, and the messages to communicate such information in each of the selected services architecture for the specific HSH system, others architectural decisions must be done, principally in the *Information architecture layer* and *Governance & policies layer*, as follows.

Depending of heterogeneity of information provided by services, several interoperation standards must be considered in the *Information architecture layer* of the HSH system. For instance, if a service provides its information using the ICD standard, transformation rules from ICD to SNOMED CT must be defined. Hence, for each message and data format different from SNOMED CT and HL7, approaches for transforming such standards must be provided. Moreover, in this layer, all healthcare plans repositories must be defined considering specificities of the HSH system under construction. In addition, inference approach to predict emergency situations by the HSH system must be also defined.

Regarding decisions in the *Governance & policies layer*, quality plans must be established in order to the HSH system can achieve quality attribute requirements. A similar work must be done for establishing environment configuration plans needed to predict, avoid and resolve possible environmental problems at patient's home.

Table 18 – Relating missions views, BPSv and diagrams in *HomecARe*

ID	View Name	Requirement	Missions	Business Process Service	Diagrams			
					Business Process	Capabilities	Interfaces, Contracts and Protocols	Services Architecture
MV1	Physical Status and Signs & Symptoms Monitored Remotely	DR01	GM1.1.B; GM1.1.1.A	BPSv001	Figure 17	Figure 18	Figures 19 - 24	Figure 25
MV2	Patient Intervention Managed	DR02	GM1.1.1.C; GM1.1.1.C.1; GM1.1.1.C.2; GM1.1.1.C.3.3; GM1.1.1.C.3.1	BPSv003; BPSv004; BPSv005; BPSv006; BPSv008	Figure 26	Figure 28	Figures 29 - 38	Figure 39
MV3	Activities of Daily Life Supported	DR04	GM1.1.2	BPSv002	Figure 40	Figure 42	Figures 43 - 49	Figure 50
MV4	Chronic Disease Managed and Patient Quality of Life Improved	DR03; DR05	GM1.1; GM1.1.1	BPSv007; BPSv010	Figure 51	Figure 52	Figures 53 - 56	Figure 57
MV5	Emergency Problem Alerted and Interventions in Emergency Situations Executed	DR06; DR07	GM1.2.1; GM1.2.3	BPSv012	Figure 58	Figure 59	Figures 60 - 65	Figure 66
MV6	Formal Care Improved	DR08	GM1.2	BPSv011	Figure 67	Figure 68	Figure 69	Figure 70

In relation to architectural viewpoints and views to document the architecture of the HSH system, *HomecARe* offers three viewpoints, i.e., conceptual, missions and quality, that are complemented with several views. For instance, the missions viewpoint (see Section 5.2) contains six missions views that can be reused by the HSH system based on the required BPSv, since each mission view is directly related to BPSv, representing configurations needed to realize BPSv. The quality viewpoint, in Section 5.3, establishes two views, i.e., the integration and interoperability view, and the quality management view, that can be reused by the HSH system to represent how quality attributes requirements, regarding interoperability, integration, reliability, security, adaptivity and safety can be met by the architecture.

To formally represent the software architecture of the HSH system, *HomecARe* offers architectural models made in SoAML[3] and UML[4] that can be instantiated in the specific HSH system. To create such models the modelling environment Modelio 3.6 [5] was used. Figure 90 shows the project structure containing models for *HomecARe* layers: Service layer (decomposed in Constituent services layer and Control services layer), Quality of service layer, Health Service Bus layer, and Business process services layer. Documentation of each diagram was given in Chapter 5.

An additional deployment viewpoint can be established by the architect of the HSH system to represent how elements, i.e., services and repositories, defined in the *Services Layered View*, in Section 5.1, are allocated in physical elements (e.g., devices, sensors, set-top-box) and how are connected and distributed through physical networks (e.g., using cloud computing architecture, or wireless sensor networks).

Activity 4 - Architectural evaluation

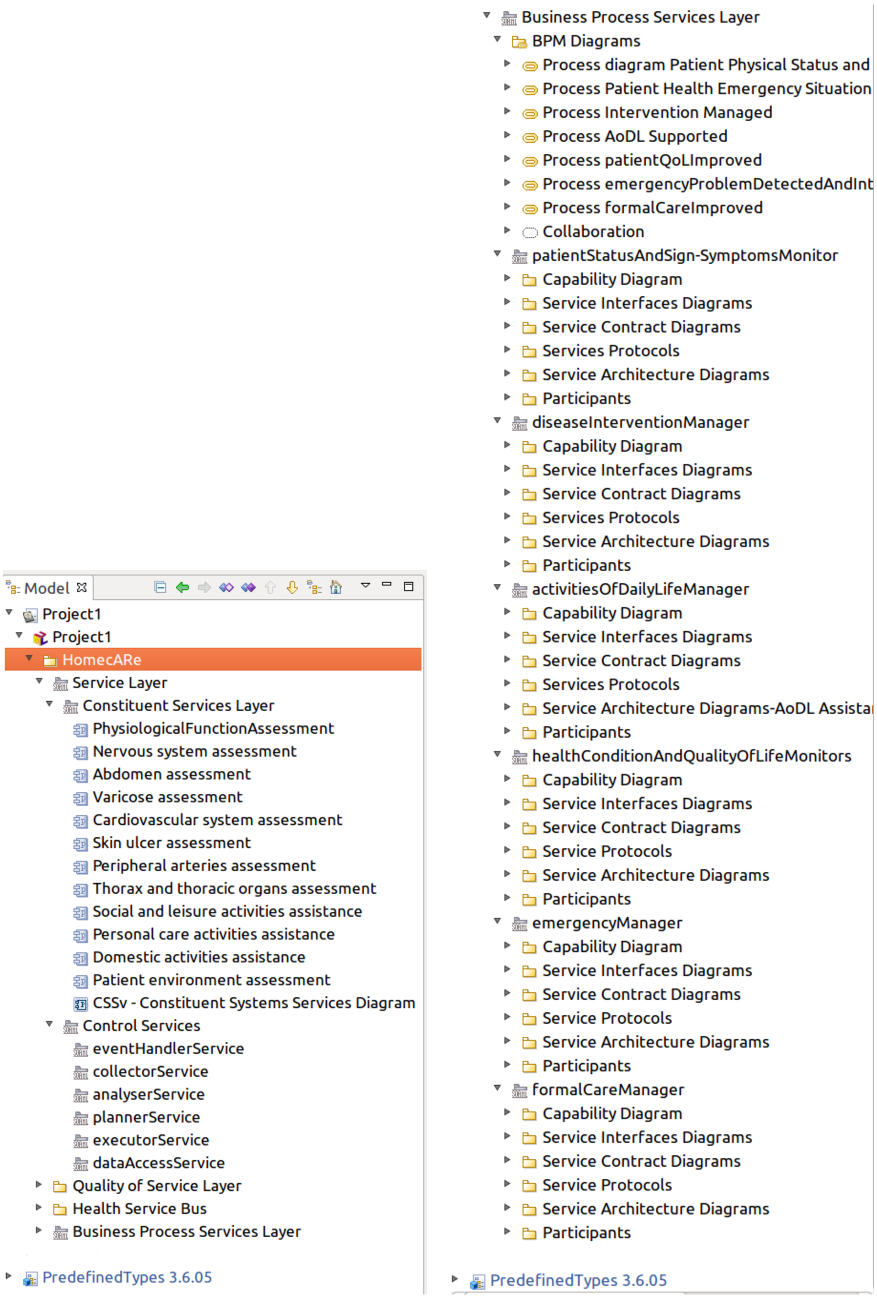
In this step, the architect must ensure if architectural design decisions that she/he made are the right ones to achieve the ASRs (HOFMEISTER *et al.*, 2005). Quantitative and qualitative approaches can be used to architectural evaluation.

An example of quantitative approach is executing simulation models of the software architecture. For this, ADLs (Architecture Description Languages) can be used to formally represent an architectural model and executing it to assess metrics related to quality attributes.

Regarding qualitative assessment, several methods that have proved useful to validate the architecture can be used (GORTON, 2011; HEESCH *et al.*, 2014): surveys, check-lists, interviews, decision based, scenario based and prototyping.

DCAR (Decision-Centric Architecture Reviews) (HEESCH *et al.*, 2014), is a lightweight approach, that aims to assess decision by decision made in a software architecture to let stake-

[3] Service Oriented Architecture Modelling Language
[4] Unified Modelling Language
[5] <https://www.modelio.org/>



(a) Services, HSB and QoS layers structure.

(b) Business process services layer structure.

Figure 90 – Project structure related to *HomecARe* architectural models.

holders analyse, understand and record the rationale behind architectural decisions. Presentations of architectural design, identification of decisions and rationale, and recording of forces in favour or against decisions are made during DCAR conduction. As result, decision prioritization that will guide further stages in the software development process are made.

Scenarios, a technique created at the SEI[6], is oriented to "manually" test issues of a software architecture concerning with quality attributes requirements (GORTON, 2011). Hence,

[6] Software Engineering Institute at Carnegie Mellon University.

using scenarios it is possible to highlight consequences of architectural decisions made during the architectural design. Examples of scenario based approaches are ATAM (Architecture Trade-off Analysis Method), SAAM (Software Architecture Analysis Method), CBAM (Cost Benefit Analysis Method) and ARID (Active Reviews for Intermediate Design).

When a software architecture have to be assessed regarding critical quality attributes such as performance or safety, perhaps, decision centric and scenario based techniques can not provide enough evidence to know if the proposed architectural design can achieve such attributes. In this context, prototypes can be considered to assess such concerns. Prototypes are used for two purposes (GORTON, 2011): as a proof-of-concept to know if the architecture design can be implemented in a way to achieve the requirements, or as a proof-of-technology, to identify if selected technology (e.g., middleware, integrated solutions) behave as expected.

APPENDIX

B

RESULTS OF INSTANTIATING *HOMECARE*

This appendix shows instances of *HomecARe* elements used to define the software architecture of *DiaManT@Home*. The first half of the spreadsheet 3 corresponds to elements of *HomecARe* (i.e., CSSv, BPSv, consumer services and healthcare plans) that were instantiated to create the software architecture of *DiaManT@Home*.

STEP : ARCHITECTURAL SYNTHESIS

Spreadsheet 3 - Services Definition

OBSERVATION: For each identified SNOMED CT children code, to create a service instance to provide such information.

	HomecARe				DiaManT@Home			
	Service ID	**Service name**	**Description**	**Parent codes in SNOMED CT**	**Children codes list**	**Service instance code**	**Service instance name**	**Functional Requirement**
Constituent Systems Services (CSSv)	**CSSv001**	**Physiological function assessment**	To offer measures on patient vital signs, such as, body temperature (BT), blood pressure (BP), pulse or heart rate (HR), and respiration rate (RR).	363789004 - General characteristic of patient (observable entity).	Body weight (observable entity) SCTID: 27113001 And Body weight measure (observable entity) SCTID: 363808001	**CSSv001-1**	Weight assessment	FR26
					Body height measure (observable entity) SCTID: 50373000 And Height / growth measure (observable entity) SCTID: 271603002	**CSSv001-2**	Body measures assessment	FR27
					Current chronological age (observable entity) SCTID: 424144002			FR28
					Body measure (observable entity) SCTID: 248326004			FR29
					Blood pressure (observable entity) SCTID: 75367002	**CSSv001-3**	Blood pressure assessment	FR37
					Blood pressure fall (qualifier value) SCTID: 255329004 and Blood pressure rise (qualifier value) SCTID: 255330009			FR68
	CSSv002	**Cardiovascular system assessment**	To offer measures on patient cardiovascular functions, such as, heart sound, cardiac flow or systole function.	70337006 - Cardiovascular function (observable entity).	Heart beat (observable entity) SCTID: 248646004	**CSSv002-1**	Heart beat monitor	FR38
					Abnormal heart beat (finding) SCTID: 361136003			FR69
	CSSv003	**Abdomen assessment**	To offer information on abdominal findings such as, abdominal aorta, mass or rigidity.	609624008 - Finding of abdomen; 249273002 - Finding of urinary tract.	Gastrointestinal tract finding (finding) SCTID: 386618008	**CSSv003-1**	gastrointestinal assessment	FR35, FR66
	CSSv004	**Nervous system assessment**	To offer measures on patient involuntary movements, such as, excessive blinking, tremors	267078001 - Involuntary movement finding.	Sleep, function (observable entity) SCTID: 258158006	**CSSv004-1**	Sleep monitor	FR39
					Dyssomnia (disorder) SCTID: 44186003			FR69b
	CSSv006	**Skin ulcer assessment**	To offer measures on patient skin ulcer, such as, pressure ulcer stage, surface area of ulcer.	439744001 - Ulcer observable (observable entity).	Infectious disease (disorder) SCTID: 40733004	**CSSv006-1**	Skin ulcer assessment	FR36, FR67

	HomecARe				DiaManT@Home			
	Service ID	**Service name**	**Description**	**Parent codes in SNOMED CT**	**Children codes list**	**Service instance code**	**Service instance name**	**Functional Requirement**
Constituent Systems Services (CSSv)	**CSSv012**	**Patient environment assessment**	To measure patient environment, such as odour, infestation, water supply	224153006 - Local environment and neighbourhood details (observable entity): 224249004 - Characteristics of home environment ; 129841008 -	Finding of tobacco smoking behavior (finding) SCTID: 365981007	**CSSv012-1**	Domotic services	FR34
					Smoking started (finding) SCTID: 266929003			FR65
	CSSv013	**Body substance assessment**	To measure body substances as body fluids, secretions, etc	430925007 - Measurement of substance (procedure)	Insulin measurement (procedure) SCTID: 16890009	**CSSv013-1**	Insuline measurement service	FR21
					Glucose measurement, blood SCTID: 33747003	**CSSv013-2**	Blood glucose assessment service	FR24, FR25
BUSINESS PROCESS SERVICES	BPSv001	Patient status and sign \& symptom monitor	Its aim is to establish patient physical status, through remote patient physical examination,and monitoring her/his sign and symptoms.	404966002 - Physical ageing status (observable entity)				
	BPSv002	**Activities of daily life manager**	It aims to provide the AoDL profile, detailing the status of patient environment and his/er social, personal care, domestic and leisure activities.	370885003 - Activities of daily living management	Physical activity (observable entity) SCTID: 68130003	**BPSv002-1**	Physical exercise manager	FR1
					Behavior showing reduced motor activity (finding) SCTID: 43994002			FR49, FR50, FR51
	BPSv004	**Pharmacological treatment manager**	Its aim is to establish the status of patient pharmacological treatment regarding her/his medication plan.	416608005 - Drug therapy.	Taking medication (observable entitiy)	**BPSv004-1**	Pharmacological treatment manager	FR19
					Patient forgets to take medication (finding) SCTID: 40836005			FR58
					Medication administered in error (event) SCTID: 370890000			FR59
	BPSv005	**Dietetic assistant**	Its aim is to establish the status of patient diet, regarding her/his nutrition plan.	185495006 - In-house dietetics.	Eating feeding / drinking observable (observable entity) SCTID: 364645004	**BPSv005-1**	Dietetic assistant	FR5
					Nutritional value, function (observable entity) SCTID: 84626001			FR9
					Unit of dietetics and nutrition (qualifier value) SCTID: 228870009			FR10
					Eating feeding / drinking observable (observable entity) SCTID: 364645004			FR12
					Nutritional value, function (observable entity) SCTID: 84626001			FR11

	HomecARe				DiaManT@Home		
	Service ID	**Service name**	**Description**	**Parent codes in SNOMED CT**	**Children codes list**	**Service instance code**	**Service instance name**	**Functional Requirement**
BUSINESS PROCESS SERVICES	**BPSv005**	**Dietetic assistant**	Its aim is to establish the status of patient diet, regarding her/his nutrition plan.	185495006 - In-house dietetics.	Alcohol intake (observable entity) SCTID: 160573003	**BPSv005-1**	Dietetic assistant	FR33
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR52
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR53
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR54
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR55
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR56
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR57
	BPSv007	**Health condition monitor**	It aims to predict the situation of a patient health condition based on her/his physical status, sign and symptoms, and intervention profile.	405157008 - Personal health status (observable entity)	Finding glucose level SCTID: 365811003	**BPSv007-1**	Diabetes monitor	FR20
					Insulin measurement (procedure) SCTID: 16890009			FR21
					Calculation technique (qualifier value) SCTID: 702873001			FR22
					Insulin measurement (procedure) SCTID: 16890009			FR23
					Hypoglycemia (disorder) SCTID: 302866003			FR60,FR61

	HomecARe				DiaManT@Home			
	Service ID	**Service name**	**Description**	**Parent codes in SNOMED CT**	**Children codes list**	**Service instance code**	**Service instance name**	**Functional Requirement**
BUSINESS PROCESS SERVICES	**BPSv007**	**Health condition monitor**	It aims to predict the situation of a patient health condition based on her/his physical status, sign and symptoms, and intervention profile.	405157008 - Personal health status (observable entity)	Hypoglycemia management (procedure) SCTID: 386328006	**BPSv007-1**	Diabetes monitor	FR62
					Hyperglycemia management (procedure) SCTID: 386326005			FR63
	BPSv0011	**Emergency manager**	Its purpose is to predict an emergency situation and to establish an emergency plan to be executed.	281694009 - Finding of a risk; 225314003 - Risk management.	Increased physical activity (finding) SCTID: 1171000175109	**BPSv0011-1**	Emergency manager	FR49
					Behavior showing reduced motor activity (finding) SCTID: 43994002			FR50
					Target physical activity (observable entity) SCTID: 391105003 Or Referral to physical activity program (procedure) SCTID: 390893007			FR51
					Inadequate food diet (finding) SCTID: 102610002 or Alteration in nutrition (finding) SCTID: 129844000			FR52, FR53, FR54, FR55, FR56, FR57
					Patient forgets to take medication (finding) SCTID: 40836005			FR58
					Medication administered in error (event) SCTID: 370890000			FR59
					Hypoglycemia (disorder) SCTID: 302866003			FR60
					Hyperglycemia (disorder) SCTID: 80394007			FR61
					Hypoglycemia management (procedure) SCTID: 386328006			FR62
					Hyperglycemia management (procedure) SCTID: 386326005			FR63
					Alcohol intake above recommended sensible limits (finding) SCTID: 160592001			FR64

	HomecARe				DiaManT@Home			
	Service ID	Service name	Description	Parent codes in SNOMED CT	Children codes list	Service instance code	Service instance name	Functional Requirement
BUSINESS PROCESS SERVICES	BPSv0011	Emergency manager	Its purpose is to predict an emergency situation and to establish an emergency plan to be executed.	281694009 - Finding of a risk; 225314003 - Risk management.	Smoking started (finding) SCTID: 266929003	BPSv0011-1	Emergency manager	FR65
					Gastrointestinal tract finding (finding) SCTID: 386618008			FR66
					Infectious disease (disorder) SCTID: 40733004			FR67
					Blood pressure fall (qualifier value) SCTID: 255329004 and Blood pressure rise (qualifier value) SCTID: 255330009			FR68
					Abnormal heart beat (finding) SCTID: 361136003			FR69
					Dyssomnia (disorder) SCTID: 44186003			FR69b
CONSUMER SERVICES - HOME HEALTHCARETEAM SERVICES	Patient Services				Eating feeding / drinking observable (observable entity) SCTID: 364645004		Meals manager	FR5
					Nutrients (substance) SCTID: 226355009 and Eating routine (observable entity) SCTID: 162549003			FR13
					Nutrients (substance) SCTID: 226355009 and Eating routine (observable entity) SCTID: 162549003			FR14
					Reminding about performing activity (procedure) SCTID: 304512002 Or Reminding about performing procedure (procedure) SCTID: 304513007		Reminder manager	FR40, FR41, FR42, FR45
					Reminding about taking drug (procedure) SCTID: 304514001			FR43
					Date treatment stopped (observable entity) SCTID: 413947000			FR44
					Reminding (procedure) SCTID: 223452003			FR46, FR47

	HomecARe				DiaManT@Home			
	Service ID	Service name	Description	Parent codes in SNOMED CT	Children codes list	Service instance code	Service instance name	Functional Requirement
CONSUMER SERVICES - HEALTH ORGANIZATION SERVICES		**Healthcare plans management services**			Target physical activity (observable entity) SCTID: 391105003 Or Referral to physical activity program (procedure) SCTID: 390893007		Exercise plan manager	FR2
					Target physical activity (observable entity) SCTID: 391105003 Or Referral to physical activity program (procedure) SCTID: 390893007			FR3
					Diet followed (observable entity) SCTID: 230125005		Nutrition plan manager	FR4
					Nutritional value, function (observable entity) SCTID: 84626001			FR6, FR7, FR8
					Drug or medicament (substance) SCTID: 410942007		Medication manager	FR15, FR16, FR17
					Substance use therapy (regime/therapy) SCTID: 385989002			FR18
		Report services			Report (record artifact) SCTID: 229059009		Report services	FR70, FR71, FR72, FR73
		Health Record services			Clinical history/examination observable (observable entity) SCTID: 363788007		Patient records services	FR30, FR31
					Allergic disposition (disorder) SCTID: 609328004			FR32

INFOMATION ARCHITECTURE - HEALTHCARE PLANS

HomecARe				DiaManT@Home			
Service ID	Service name	Description	Parent codes in SNOMED CT	Children codes list	Service instance code	Service instance name	Functional Requirement
	Medication Plan			Drug or medicament (substance) SCTID: 410942007		Medication plan repository	FR15
				Substance use therapy (regime/therapy) SCTID: 385989002			FR18
				Report (record artifact) SCTID: 229059009			FR72
	Patient DL Record			Finding of at risk (finding) SCTID: 281694009		Patient DL Records	FR48
				Glucose measurement, blood SCTID: 33747003			FR24
				Finding glucose level SCTID: 365811003			FR20
				Taking medication (observable entitiy) SCTID 129019007			FR19
				Clinical history/examination observable (observable entity) SCTID: 363788007			FR30
				Clinical history/examination observable (observable entity) SCTID: 363788007			FR31
				Allergic disposition (disorder) SCTID: 609328004			FR32
				Report (record artifact) SCTID: 229059009			FR73
				Report (record artifact) SCTID: 229059009			FR70
	Nutrition Plan			Nutritional value, function (observable entity) SCTID: 84626001		Nutrition plan repository	FR6, FR7, FR8, FR11, FR12, FR13
				Report (record artifact) SCTID: 229059009			FR71
	Calendar			Reminding about performing activity (procedure) SCTID: 304512002 Or Reminding about performing procedure (procedure) SCTID: 304513007		Calendar	FR40, FR41, FR42, FR45
				Reminding about taking drug (procedure) SCTID: 304514001			FR43
				Date treatment stopped (observable entity) SCTID: 413947000			FR44
				Reminding (procedure) SCTID: 223452003			FR46, FR47

APPENDIX

C

A TAXONOMY OF MEDIATORS FOR SOS

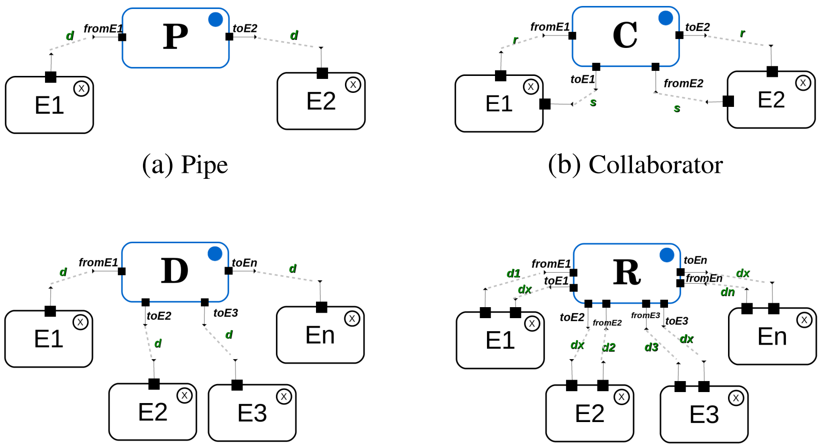
This taxonomy aims to aid on the selection of mediators types that will compose the software architecture of an SoS, in both, design and execution time. The taxonomy was established based on a deeper study on software connectors (LOPES; WERMELINGER; FIADEIRO, 2003; LAU; VELASCO; ZHENG, 2005; AMIRAT; OUSSALAH, 2009; KIWELEKAR; JOSHI, 2010; MEHTA; MEDVIDOVIC; PHADKE, 2000) and mediators (LI *et al.*, 2008; SPALAZZESE; INVERARDI; ISSARNY, 2009; ISSARNY; BENNACEUR, 2012; TOMSON; PREDEN, 2013; BENNACEUR; ISSARNY, 2015), and based on our previous experience in software architecture and SoS.

Twelve mediators in three categories are proposed in our taxonomy to provide capabilities of communication, conversion and control to SoS software architectures (GARCÉS; OQUENDO; NAKAGAWA, 2018). In this section, each mediator type in each category is presented. It is worth noting that, mediators proposed herein, can be used to mediate interactions among constituent systems, as well as among other mediators. Hence, in the remainder of this section, we refer to both, constituent systems and mediators, as entities E that can be mediated.

Communication Mediators

Mediators for communication allow transmission of data, messages or results among entities E_i , i.e., among constituent systems, among constituent systems and other mediators or among mediators. Four mediators for communication purposes are defined in the taxonomy: pipe, collaborator, distributor and router, as showed in Figure 91.

Pipe: This mediator is used to communicate data d from E_1 to E_2. It is an unidirectional transfer, as showed in Figure 91a. Pipes can be used when an SoS just receives output messages from constituent systems, but no request can be send to their constituents, such in virtual SoS. <u>Duties:</u> To transmit d from E_1 to E_2. <u>Behaviours:</u> $Transmit(d)$ using $output$ interface in E_1 to



(a) Pipe

(b) Collaborator

(c) Distributor

(d) Router

Figure 91 – Mediators types for communication

input interface in E_2 . Assumptions: E_1 to E_2 interfaces are known, and d can be processed or understood by E_2. Guarantees: d is transmitted to E_2.

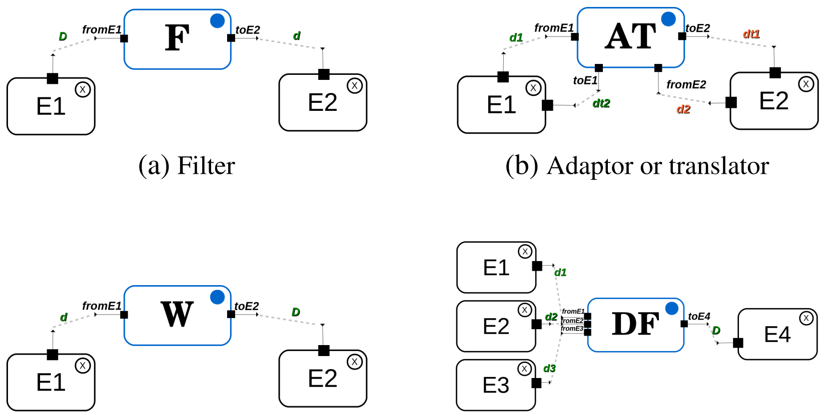
Collaborator: This mediator can be used to request/supply services/data between entities, E_1 and E_2, as illustrated in Figure 91b. Collaborators can be used in SoS with certain control level over its constituents systems, such in directed or acknowledged SoS. Duties: Processing of requests and provide responses. Behaviors: $requestToE2(d)$, $supplyToE1(r)$, where d is the requested service or data from E_1 to E_2, and r is the response to that request from E_2 to E_1. Assumptions: Requested services are provided by suppliers and answers can be understood by requesters. Guarantees: Synchronous/ asynchronous delivery of requests and answers. Buffering is needed when requests are streams.

Distributor: It is useful in distributed systems to communicate (i.e., broadcasting) data d to several receptors. Distributors can be used when an SoS needs to communicate reconfiguration requests or for coordination purposes among multiple entities. Distributor is depicted in Figure 91c. Duties: To distribute to interested entities. Behaviours: $distribute(d)$ to an entities list $entityList$ interested in receiving the data. Assumptions: Type of distribution is broadcast or directed, and receptors are previously known by emissor. Guarantees: To deliver d to interested entities, and d can be understood by receptors.

Router: It is used to control and coordinate data flow when multiple entities are transmitting and receiving data in an SoS, as presented in Figure 91d. Duties: Routing data between entities. Behaviours: $route(d_j)$ to $entityList$. Assumptions: d_j from E_i can be understood by other peers. Guarantees: Transmission of data between multiple entities.

Conversion Mediators

This type of mediators are concerned about interoperability among constituent systems and security issues when data are transmitted. Four mediators, showed in Figure 92, are defined to support interoperable and secure communications and collaborations among entities, through operations on exchanged data or protocols: filter, adaptor, wrapper and aggregator.



(a) Filter (b) Adaptor or translator

(c) Wrapper (d) Aggregator or data fusioner

Figure 92 – Mediators types for conversion

Filter: It is used to select and remove parts of input data D simplifying its structure. In SoS, filters can be used to select relevant information from outputs of constituent systems, which can contain more information than the needed for SoS operation. The filter mediator is presented in Figure 92a. Duties: To identify and filter d from input data D. Behaviours: $d = filter(D)$. Assumptions: D can be understood by the filter, and d is contained in D. Guarantees: Selection and filtration of data d from D.

Wrapper: It is used to allow interoperability among constituent systems and SoS mediators that use different communication or transport protocols. Moreover, it can be used to add encryption data or authentication information for security purposes. The wrapper adds extra information w to data exchanged d, as depicted in Figure 92c. Duties: To create D adding w to d. Behaviours: $D = wrap(d,w)$; $d = unWrap(D,w)$. Assumptions: It exists a protocol to wrap and unwrap data. Guarantees: D containing d plus additional information w.

Adaptor / Translator: This mediator allows semantic interoperability between constituent systems and SoS mediators exchanging data, though the translation or matching of different data formats dt_i. Moreover, adaptors can be used to match interaction protocols among constituent systems. Figure 92b shows the adaptor mediator. Duties: To adapt d_1 into d_2, where $d_1.Datatype \neq d_2.Datatype$, using the transformation rule tr. Behaviours: $d_2 = Adapt(d_1,tr)$. Assumptions: It exists a transformation rule tr to allow translation between exchanged data format. Guarantees: Correct transformation of data transmitted between two entities.

Aggregator / Data Fusion: Is used to collect and merge individual data d_1, d_2, d_n from different entities E_1, E_2, E_n and to create a single aggregated output D data for further transmission, where D contains (parts of) d_1, d_2, d_n and, if needed, some additional information w. Aggregated data are important to allow SoS understanding about its current status, and to decide future behaviours or reconfigurations in its structure. The aggregator mediator is presented in Figure 92d. Duties: To merge d_1, d_2, d_n into D. Behaviours: $D = aggregate(d_1, d_2, ..., d_n, w)$. Assumptions: d_1, d_2, d_n are syntactically interoperable. Moreover, a predefined order (or protocol) for data aggregation must be defined. Guarantees: D as a fusion of d_1, d_2, d_n, w.

Control Mediators

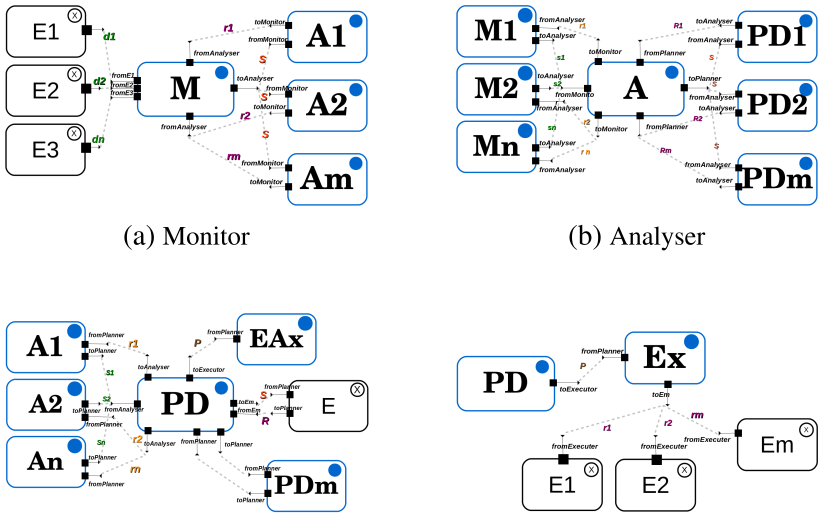
Mediators in this category are concerned with behavioural and control issues of SoS software architectures. Four mediators are defined, i.e., monitor, analyser, planner and executer, to deal with architectural reconfigurations due to execution of emergent behaviours, availability of constituent systems capabilities, or changes in SoS missions or its environment. Mediators types for control purposes are presented in Figure 93.

Monitor / Collector: It is used to collect information d_1, d_2, d_n from groups of SoS entities $entityList$, and provide their status s (e.g., constituents behaviour under execution, current environment status, constituent availability). Figure 93a depicts the monitor mediator. Duties: To infer situation based on entities current status. Behaviours: $situation = monitor(entityList)$; $sendSituation(situation)$. Assumptions: Information received from entities is semantically interoperable. Guarantees: Situation inference of entities participating of the SoS.

Analyser: It allows to establish situation of an SoS based on historical knowledge on SoS and their entities status. Moreover, an analyser predicts which emergent behaviour is under execution, which SoS mission is addressed or if the SoS is presenting problems in its operation. The analyser mediator is presented in Figure 93b. Duties: To infer SoS situation. Behaviours: $SoSituation(SoSModel)$. Assumptions: There is historical knowledge on SoS previous status, prediction models, situation of entities and SoS models. Guarantees: Situation of the SoS.

Planner / Decider: It helps to select the best configuration plan to be executed by the SoS (or by parts of it). Configuration plans allow to change the SoS architecture (e.g., adding or removing mediators) and its behaviours, and to request new functionalities to constituent systems. Figure 93c illustrates the planner mediator. Duties: To establish and provide reconfiguration plans. Behaviours: $planning - Reconf()$, $consultPlans()$, $requestPlan()$, $providePlan()$. Assumptions: A knowledge base with SoS current and historical status, reconfiguration plans and policies must be defined. Guarantees: Reconfiguration plans.

Executer / Actuator: It realizes the reconfiguration plan provided by planners, and distributes reconfiguration requests to specific SoS entities (i.e., entities under control of the SoS). The executer is showed in Figure 93d. Duties: To distribute reconfiguration requests to entities. Behaviours: $sendReconf(entityList, reconfRequest)$. Assumptions: Plans are semanti-



(a) Monitor

(b) Analyser

(c) Planner or decider

(d) Executor or actuator

Figure 93 – Mediators types for control

cally interoperable to be understood by entities. Guarantees: All reconfiguration requests in a reconfiguration plan are executed.

APPENDIX

D

DECLARATION OF ORIGINAL AUTHORSHIP AND LIST OF PUBLICATIONS

I, Lina María Garcés Rodríguez, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that this work was done wholly or mainly while in candidature for a double degree at the University of São Paulo and University of South Brittany. Where I have consulted the published work of others, this is always clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Excerpts of this dissertation have been either published or submitted for the appreciation of editorial boards of journals, conferences, and workshops, according to the list of publications presented below.

Publications Resulting from this Thesis

- **GARCÉS, LINA**; AMPATZOGLOU, APOSTOLOS ; AVGERIOU, PARIS ; NAKAGAWA, ELISA YUMI . Quality attributes and quality models for ambient assisted living software systems:A systematic mapping.

  Journal: Information and Software Technology, v. 82, p. 121-138, 2016. Qualis: A2. DOI: <http://dx.doi.org/10.1016/j.infsof.2016.10.005>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCÉS, LINA**; NAKAGAWA, ELISA YUMI . A process to establish, model and validate missions of systems-of-systems in reference architectures.

Event: 32nd ACM/SIGAPP Symposium on Applied Computing - SAC '17. New York: ACM Press, 2017. p. 1765-1772. Qualis: A2. DOI: <http://dx.doi.org/10.1145/3019612.3019799>.

Level of contribution: High – the PhD candidate is the main investigator.

- **GARCÉS, LINA**; OQUENDO, FLAVIO ; NAKAGAWA, ELISA YUMI . QM4AAL: quality model for ambient assisted living systems. Sao Carlos 2017

  Technical Report: Available online: <http://repositorio.icmc.usp.br//handle/RIICMC/6567>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCÉS, LINA**; OQUENDO, FLAVIO ; NAKAGAWA, ELISA YUMI . A REFERENCE ARCHITECTURE FOR HEALTHCARE SUPPORTIVE HOME SYSTEMS: MISSIONS ESTABLISHMENT AND VALIDATION. Sao Carlos 2017 (Technical Report). Available online: <http://repositorio.icmc.usp.br//handle/RIICMC/6566>.

- **GARCÉS, LINA**; AMPATZOGLOU, A. ; OQUENDO, FLAVIO ; NAKAGAWA, ELISA YUMI . Reference architectures and reference models for ambient assisted living systems: a systematic literature review. São Carlos 2017.

  Technical Report: Available online: <http://repositorio.icmc.usp.br//handle/RIICMC/6564>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCÉS, LINA**; AMPATZOGLOU, APOSTOLOS ; AVGERIOU, PARIS ; NAKAGAWA, ELISA YUMI . A SYSTEMATIC MAPPING ON QUALITY ATTRIBUTES AND QUALITY MODELS FOR AMBIENT ASSISTED LIVING SYSTEMS. São Carlos, SP: Biblioteca ICMC-USP, 2016.

  Technical Report: Available online: <http://repositorio.icmc.usp.br//handle/RIICMC/6646>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCES, LINA**; OQUENDO, FLAVIO ; NAKAGAWA, ELISA YUMI . A Quality Model for AAL Software Systems.

  Event: IEEE 29th International Symposium on ComputerBased Medical Systems (CBMS), 2016, Belfast and Dublin. p. 175-180. Qualis:B1. DOI: <http://dx.doi.org/10.1109/CBMS.2016.46>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCES, LINA**; AMPATZOGLOU, APOSTOLOS ; AVGERIOU, PARIS ; NAKAGAWA, ELISA YUMI . A Comparative Analysis of Reference Architectures for Healthcare in the Ambient Assisted Living Domain.

  Event: IEEE 28th International Symposium on ComputerBased Medical Systems (CBMS), 2015, Sao Carlos. Qualis: B1. DOI: <http://dx.doi.org/10.1109/CBMS.2015.29>.

  Level of contribution: High – the PhD candidate is the main investigator.

- **GARCES, LINA**; AMPATZOGLOU, APOSTOLOS ; AVGERIOU, PARIS ; NAKAGAWA, ELISA YUMI . A Reference Architecture for Healthcare Supportive Home Systems.

  Event: IEEE 28th International Symposium on ComputerBased Medical Systems (CBMS), 2015, Sao Carlos. p. 358-359. Qualis: B1. DOI: <http://dx.doi.org/10.1109/CBMS.2015.39>.

  Level of contribution: High – the PhD candidate is the main investigator.

Other Related Publications

- **GARCÉS, LINA**; FELIZARDO, KATIA ; OLIVEIRA, LUCAS ; NAKAGAWA, ELISA. An Experience Report on Update of Systematic Literature Reviews.

  Event: The 29th International Conference on Software Engineering and Knowledge Engineering, 2017, Pittsburgh, PA, USA, 2017. p. 91-96. Qualis: B1. DOI: <http://dx.doi.org/10.18293/SEKE2017-078>.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- GRACIANO NETO, V. V. ; MANZANO, W. ; **GARCES, LINA** ; GUESSI, MILENA ; OLIVEIRA, B. ; VOLPATO, T. ; Elisa Y. Nakagawa. Back-SoS: a Model-based Approach to Prevent Architectural Drift in Systems-of-Systems.

  Event: 33rd ACM/SIGAPP Symposium On Applied Computing, 2018, Pau, France. p. 1-3.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- GRACIANO NETO, V. V. ; PAES, CARLOS ; **GARCÉS, LINA** ; GUESSI, MILENA ; MANZANO, W. ; OQUENDO, FLAVIO ; NAKAGAWA, ELISA YUMI . Stimuli-SoS: A Model-Based Approach to Derive Stimuli Generators for Simulations of Systems-of-Systems Software Architectures.

  Journal: Journal of The Brazilian Computer Society, (2017) 23:13. DOI: <10.1186/s13173-017-0062-y>.

Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- GRACIANO NETO, V. V. ; **GARCES, LINA** ; GUESSI, MILENA ; PAES, C. ; MANZANO, W. ; OQUENDO, FLAVIO ; Elisa Y. Nakagawa . ASAS: An Approach to Support Simulation of Smart Systems.

  Event: 51th The Hawaii International Conference on System Sciences, 2018. p. 1-10.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- VOLPATO, TIAGO ; OLIVEIRA, BRAUNER R. N. ; **GARCÉS, LINA** ; CAPILLA, RAFAEL ; NAKAGAWA, ELISA YUMI . Two perspectives on reference architecture sustainability.

  Event: 11th European Conference on Software Architecture Companion Proceedings - ECSA '17. New York: ACM Press, 2017. p. 188-194.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- NAKAGAWA, ELISA YUMI.; DIAS, DIÓGENES.; ... **GARCÉS, LINA**. et al. ; Software architecture and reference architecture of software-intensive systems and systems-of-systems.

  Event: 11th European Conference on Software Architecture Companion Proceedings - ECSA '17. New York: ACM Press, 2017. p. 4-11.

  Level of contribution: Low – the PhD candidate participated in the paper writing.

- VICENTE, I. ; **GARCÉS, LINA** ; NAKAGAWA, ELISA YUMI . Establishment of a Software Architecture for Healthcare Supportive Home System to Assist Patients with Diabetes Mellitus.

  Event: 25o Simpósio Internacional de Iniciação Científica da USP - SIICUSP, 2017. p. 1-1.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- MORAIS, M. G. C. ; **GARCÉS, L. M.** ; NAKAGAWA, ELISA YUMI . A Health Care Supportive Home System for Cardiovascular Diseases.

  Event: XXIV Simpósio Internacional de Iniciação Científica da Universidade de São Paulo (SIICUSP 2016), 2016. p. 1-1.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- NETO, V. ; OLIVEIRA, L. B. R. ; GUESSI, M. ; OQUENDO, F. ; **GARCÉS, L. M.** ; Elisa Y. Nakagawa . A Conceptual Map of Model-Driven Development for Systems-of-Systems.

  Event: IX Workshop em Desenvolvimento Distribuído de Software, Ecossistemas de Software e Sistemas de Sistemas (WDES), 2015. p. 1-4.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- GRACIANO NETO, VALDEMAR VICENTE ; **GARCÉS, LINA** ; GUESSI, MILENA ; DE OLIVEIRA, LUCAS BUENO RUAS ; OQUENDO, FLAVIO . On the Equivalence between Reference Architectures and Metamodels.

  Event: 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures - CobRA '15. p. 21-24.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- GUESSI, MILENA ; OLIVEIRA, LUCAS BUENO RUAS ; **GARCÉS, LINA** ; OQUENDO, FLAVIO . Towards a Formal Description of Reference Architectures for Embedded Systems.

  Event: 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures - CobRA '15. p. 17-20.

  Level of contribution: Medium – the PhD candidate participated in the research conduction and paper writing.

- NETO, V. ; **GARCÉS, L. M.** ; BOSCARIOLI, C. ; Elisa Y. Nakagawa . Investigating Issues of Human-Computer Interaction for Systems-of-Systems.

  Event: 9th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, 2015, Belo Horizonte, Brazil, 2015. p. 1-4.

  Level of contribution: Low – the PhD candidate participated in the paper writing.

ICMC USP
SÃO CARLOS