METHODOLOGICAL SUPPORT TO DEVELOP INTEROPERABLE APPLICATIONS FOR PERVASIVE HEALTHCARE

João Luís Cardoso de Moraes

Methodological Support to Develop Interoperable Applications for Pervasive Healthcare

João Luís Cardoso de Moraes

CTIT CENTRE FOR TELEMATICS AND INFORMATION TECHNOLOGY

Enschede, the Netherlands, 2014

CTIT Ph.D. – Thesis Series, No. 14-321

Cover Design: Karoline V. H. Moraes
Printing: Ipskamp, Enschede, the Netherlands

Graduation committee:

Chairman, secretary:	Prof. Dr. P. M. G. Apers (University of Twente)
Promotor:	Prof. Dr. R. J. Wieringa (University of Twente)
Assistant Promotors:	Dr. L. Ferreira Pires (University of Twente)
Members:	Prof. Dr. W. Lopes de Souza (Federal University of São Carlos)
	Dr. L. Tricai Cavalini (Fluminense Federal University)
	Prof. Dr. Ir. M. Aiello (University of Groningen)
	Prof. Dr. Ir. L. J. M. Nieuwenhuis (University of Twente)
	Dr. Ir. M. J. van Sinderen (University of Twente)

CTIT Ph.D. Thesis Series No. 14-321
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE
Enschede, the Netherlands

ISBN 978-90-365-3729-2
ISSN 1381-3617 (CTIT Ph.D. Thesis Series No. 14-321)
DOI 10.3990/1.9789036537292
http://dx.doi.org/10.3990/1.9789036537292

METHODOLOGICAL SUPPORT TO DEVELOP INTEROPERABLE APPLICATIONS FOR PERVASIVE HEALTHCARE

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de Rector Magnificus,
prof.dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 5 November 2014 te 16.45 uur.

door
João Luís Cardoso de Moraes
geboren op 5 juli 1965
te Pirassununga, São Paulo, Brazilië

Dit proefschrift is goedgekeurd door:
Prof. Dr. Roel J. Wieringa (promotor)
Dr. Luís Ferreira Pires (assistent-promotor)

METHODOLOGICAL SUPPORT TO DEVELOP INTEROPERABLE APPLICATIONS FOR PERVASIVE HEALTHCARE

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the Rector Magnificus,
Prof. Dr. H. Brinksma,
on account of the decision of the graduation committee
to be publicaly defended
on Thursday the 5th of November 2014 at 16.45.

by
João Luís Cardoso de Moraes
born on 5th July 1965
in Pirassununga, São Paulo, Brazil

This dissertation has been approved by:
Prof. Dr. Roel J. Wieringa (promotor)
Dr. Luís Ferreira Pires (assistent-promotor)

TESE DE DOUTORADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIA DA COMPUTAÇÃO

"METHODOLOGICAL SUPPORT TO DEVELOP INTEROPERABLE APPLICATIONS FOR PERVASIVE HEALTHCARE"

AUTOR: JOÃO LUÍS CARDOSO DE MORAES
ORIENTADOR: PROF. DR. WANDERLEY LOPES DE SOUZA

SÃO CARLOS
DEZEMBRO/2014

CAIXA POSTAL 676
FONE/FAX: (16) 3351-8233
13565-905 - SÃO CARLOS - SP
BRASIL

Composição da Banca Examinadora:

Orientador:	Prof. Dr. Wanderley Lopes de Souza, Universidade Federal de São Carlos (UFSCar), Brasil
Membros:	Dr. Luís Ferreira Pires, Universidade de Twente, Holanda Dr. Antonio Tadeu Azevedo Gomes, Laboratório Nacional de Computação Científica (LNCC), Brasil Dra. Rossana Maria de Castro Andrade, Universidade Federal do Ceará (UFC), Brasil Dr. Cléver Ricardo Guareis de Farias, Universidade de São Paulo (USP), Brasil

Abstract

The healthcare model currently being used in most countries will soon be inadequate, due to the increasing care costs of a growing population of elderly people, the rapid increase of chronic diseases, the growing demand for new treatments and technologies, and the relative decrease in the number of healthcare professionals with respect to the population increase. This healthcare model, which is centered on highly specialized people, located in large hospitals, needs to change into a distributed model, in order to produce faster responses and to allow patients to manage their own health. This centralized healthcare model also implies that patients and healthcare professionals have to move to the same place (a hospital or clinic) for the healthcare services to be delivered, and it is often expensive and inefficient. A distributed healthcare model that pervades the daily lives of the citizens is more appropriate to provide less expensive and more effective and timely healthcare, and characterizes Pervasive Healthcare. In Pervasive Healthcare, novel information and communication technologies are applied to support the provision of health services anywhere, at anytime and to anyone.

This thesis aims at demonstrating the feasibility of using healthcare standards, ubiquitous computing technologies, service-oriented architecture principles, and software agents, to develop interoperable applications for exchanging messages in Pervasive Healthcare environments. This thesis presents a Reusable Architecture and a Message Generator that employ these technologies for achieving its aim. Our Reusable Architecture and Message Generator also meet the requirements of interoperability between heterogeneous healthcare information systems that are usually found in Pervasive Healthcare environments.

In this work, a case study was performed in a realistic distributed healthcare environment, where three usage scenarios were defined in the cardiology domain: delivery of laboratory analysis results, schedule of an appointment for pacemaker evaluation, and support of a medical staff meeting to prepare for a cardiac surgery. Using the Reusable Architecture and the Message Generator, three applications were built for these scenarios, and they were employed and evaluated by both healthcare professionals and patients. The experiments were conducted in a distributed environment that consists of three cardiology clinics, one analysis laboratory, and the cardiology department of the Santa Casa Hospital of Marília (São Paulo, Brazil). The evaluation of these applications was carried out using the Technology Acceptance Model (TAM), since this model is generally considered as appropriate to explain technology acceptance. The evaluation involved a limited but representative set of healthcare professionals and patients.

We showed that the methodology proposed in this thesis truly facilitates the development of Pervasive Healthcare. The proposed methodology preserves the investment in the legacy healthcare information systems and allows developers to add new features to them, aiming to fill the automation gap and satisfying the need for increased user mobility, while providing smart assistance to the end-users. The results obtained so far are highly promising, and encourage us to continue exploring our methodology, the Reusable Architecture and the Message Generator to develop communication services that are suitable for other healthcare scenarios.

Acknowledgements

Completing a PhD thesis is truly like running a marathon. They are both endurance events, they last a long time and they require a consistent and carefully calculated amount of effort to complete them and not burn out. One of the joys of completion is to look over the journey past and remember all the friends and family who have helped and supported me along this long but fulfilling road.

This thesis project started five years ago at Computer Science Department at Federal University of São Carlos in the context of Ubiquitous Computer Group. During the first three years of my PhD research, I have experienced one of the memorable times in my life. After that, I was invited to continue my research project at University of Twente. During the last two years working in the warm and friendly environment of SCS group, I felt at home living with the SCS members.

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here. First and above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully. I wish to praise and thank for the gift of our Mother Thrice Admirable, Queen and Victress of Schoenstatt. Thank you for the wisdom and perseverance that He has been bestowed upon me during this research project, and indeed, throughout my life: '*I can do everything through Him who gives me strength.' (Philippians 4: 13)*.

This thesis appears in its current form due to the assistance and guidance of important people. I would therefore like to offer my sincere thanks to all of them. I will never forget the help and support of our secretaries Suse and Bertine. You have been always supportive to me indeed like old friends. I never forgot your time and effort to contact different companies and agencies in order to help me solve many personal problems in the Netherlands. Bertine, thank you for your support and kind words. Suse, thank you for all your help, and the arrangements for my defense would not have been possible without you. Geert Jan, my appreciation for the technical assistance you have provided for our office. I would also like to express my warm thanks to Cristina and Dr. Hermes (UFSCar) who provided me with the facilities being required and conductive conditions for my PhD project in Brazil.

My thanks and especially grateful also go out to the support I received from my commander Colonel Robert de Oliveira for always believing in me and encouraging me to follow my dreams, and without whom, I would not have had the possibility to embark on this journey in the first place.

I would especially like to thank Dr. Rubens Tofano de Barros and all the healthcare professionals in the cardiology department at CCCM (Marília/SP-Brazil). You have been there to support me when I asked about clinical concepts needed for my PhD thesis. I would like to express my appreciation to all those healthcare providers who have offered me their time when I collected necessary data for my case study.

I would like to express my sincere appreciation to my supervisor Prof. Dr. Wanderley Lopes de Souza at Federal University of São Carlos for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. I would like to express the deepest appreciation as you continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching.

I would like to express my heartfelt gratitude and thanks to my assistant-promoter at University of Twente Dr. Luís Ferreira Pires, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Throughout my thesis-writing period, you provided good teaching, good company, and lots of good ideas. Your guidance helped me in all the time of research and writing of this thesis. This was really motivating to work harder and go further. Your sound advice on both research as well as on my career have been priceless.

I would like to thank my promoter Prof. Dr. Roel J. Wieringa for your support in the last year. You taught me to define the research methodology of my thesis in a clear and concrete way, which was quite helpful for my PhD research and even for my personal life. I am really proud of being your student and enjoyed working in your group. The experience has been an interesting and rewarding one.

Besides my advisors, I would also like to thank my committee members: Dr. Luciana Tricai Cavalini (Fluminense Federal University), Prof. Dr. Ir. Marco Aiello (University of Groningen), Prof. Dr. Ir. L. J. M. Bart Nieuwenhuis (University of Twente), Dr. Ir. Marten J. van Sinderen (University of Twente), Dr. Antonio Tadeu Azevedo Gomes, National Laboratory for Scientific Computing (LNCC), Dr. Rossana Maria de Castro Andrade, Federal University of Ceará (UFC), and Dr. Cléver Ricardo Guareis de Farias, University of São Paulo (USP). Thank you all

for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions during the preparation of this thesis. I believe my thesis has quite improved by applying your comments.

I would like to express my gratitude to my esteemed co-supervisor (in Brazil) Dr. Francisco Antonio do Prado for the useful comments, remarks and engagement through the learning process of this thesis. Furthermore I would like to thank you for introducing me to the topic as well for the support on the way.

My sincere thanks also go to Dr. Luciana Tricai Cavalini for your suggestions to work on diverse exciting projects. Your good advice, support and friendship have been invaluable on both academic and personal level, for which I am extremely grateful.

I thank my fellow labmates in São Carlos (Brazil) for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last years. Also I thank my colleagues in University of Twente for providing a good atmosphere in our group.

I want to dedicate some words also to my dear friend and English teacher Holly Dilatush. You mean a lot to me and I would like to share this important moment with you as well. Whatever happens in life, some people are and will always be important. I will always keep the beautiful moments spent together as precious memories in my heart during our English classes on Skype. I am so thanks for your help and improvement in my English skills. Holly, keep your dreams alive on LEWWWP. Remember all things are possible for those who believe.

I thank the Brazilian National Council of Technological and Scientific Development (CNPq) for sponsoring our research in the context of the Brazilian National Institute of Science and Technology in Medicine Assisted by Scientific Computing (INCT-MACC). I also thank the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES) for sponsoring my stay at the University of Twente.

My siblings have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice. I would also like to thank all of my fraternal brotherhood who supported my family and encouraged me to strive towards my goal.

A special thanks to my family. Words cannot express how grateful I am to my mother-in-law and father-in-law for all of the sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far.

Thank you my parents for giving birth to me at the first place and supporting me spiritually throughout my life. You have shown me how to believe in the power of prayer, and you have helped me to believe in the

power that I possess. Thanks for not letting '*can't*' be in my vocabulary. To my father, João (*paizão*), thanks for your encouragement and letting me know that you are proud of me. To my mother, Zaira (*maezona*), thanks for being an inspirational matriarch for our family.

I cannot conclude these acknowledgements without mentioning my beloved family. Their endless love, passion and patience as well as their sharing in both happiness and sadness have motivated my every effort and inspiration in producing this thesis. It is to them I also wish to convey my considerable thanks, and it is also to them I would like to dedicate this thesis.

Deeply from my heart with love and faith, I wish to thank to my beloved children, Karoline and João Luís, who are my life and make me a better father and man. I want to express my gratitude and deepest appreciation to my lovely and friendly sweet daughter for her great patience and understandings and for being a nice Brazilian girl. Thank you my friendly son for doing your best to understand a father who had to be confined to his study for such a long time. I will never forget our greatest moment in which we were watching Game of Thrones series together. My energy and spirit were regularly renewed during our holidays at home, in the Netherlands or in Brazil, when we joined for fun. You have always been there for me, either at a distance or visiting every time you could, and always did your best to make my life run as smoothly and happily as possible.

Last but not the least, I must acknowledge my beloved wife and best friend, Elaine, without whose love, encouragement and editing assistance, I would not have finished this thesis. You spent sleepless nights with me and was always my support in the moments when there was no one to answer my queries. I would like to thank you for your personal support and great patience at all times. Your firm and kind-hearted personality has affected me to be faithful and never bend to difficulty. You always let me know that you are proud of me, which motivates me to work harder and do my best. Being both a mother and father while I was away was not an easy thing for you. You took every responsibility and suffered all the bitterness to take care of our children and my parents. Words would never say how grateful I am to you. I doubt that I will ever be able to convey my appreciation fully, but I owe you my eternal gratitude. It was you who provided the fundamental needs of our family, and I understand it was difficult for you, therefore, I can just say thanks for everything and may God give you all the best in return.

João Luís Cardoso de Moraes
November, 2014 - Enschede, the Netherlands

Contents

Chapter 1

Introduction

With the worldwide deployment of wireless and mobile networks and dynamics mobile society, many innovative designs of healthcare applications will be able to be supported. Much progress has been made to fulfill the vision of Pervasive Healthcare to provide healthcare anywhere, at anytime and to anyone.

This thesis addresses the methodological and technological support to develop interoperable applications to exchange context-aware messages in Pervasive Healthcare environments. This support has been developed by applying healthcare standards, ubiquitous computing technologies, service-oriented architecture and software agents. This chapter presents the motivation of this thesis, discusses our main objectives and outlines the adopted research design.

This chapter is organized as follows: Section 1.1 provides the relevant background for our research, Section 1.2 motivates the work in this thesis, Section 1.3 outlines our main research objectives, Section 1.4 elaborates on the research questions to be answered in order to achieve our objectives, Section 1.5 presents the research design adopted in this thesis, Section 1.6 describes the scope of this work, and finally Section 1.7 presents the structure of this thesis.

1.1 Background

The healthcare model currently applied in most countries nowadays is centered on highly specialized people, located in large hospitals, and it focuses on treatment of acute cases. However, this healthcare model is becoming inadequate, due to the increasing healthcare costs of a growing population of elderly people, the rapid increase of chronic diseases, the growing demand for new treatments and technologies, and the relative decrease in the number of health professionals compared with the population increase. Recently, the United States Census Bureau has estimated that the expected number of inhabitants in the United States aged more than 65 will be approximately 70 million in 2030, twice more than in 2000 [1]. In Ontario, the most populous province of Canada, the healthcare is expected to represent 66% of the government expenditure in 2017, and 100% in 2026 [2].

The traditional healthcare model needs to change into a distributed model, in order to produce faster responses and to allow patients to manage their own health. The centralized healthcare model implies that patients and healthcare professionals have to move to the same place (a hospital or clinic) for the healthcare services to be delivered, and it is often expensive and inefficient. A distributed healthcare model that pervades the daily lives of the citizens is more appropriate to provide less expensive and more effective and timely healthcare. According to [3], the goal of Pervasive Healthcare is to enable the management of health and wellness by using information and communication technologies in order to make healthcare available anywhere, at anytime and to anyone.

Ubiquitous Computing [4] encompasses technologies that explore the advances of wireless connectivity in order to allow information to move along with its user. In healthcare, these technologies are being mainly employed to build supporting infrastructures for Health Information Systems (HIS), and in the development of mobile applications that extend the functionality of healthcare applications formerly limited by traditional computing technologies. However, the Pervasive Healthcare model will only be widely accepted in realistic healthcare scenarios if it supports efficient and secure information exchange among the healthcare professionals and their patients, which requires some more research.

The exchange of health information among heterogeneous Electronic Healthcare Record (EHR) systems in Pervasive Healthcare environments requires communication standards that enable the interoperability between these systems. Although Health Level Seven (HL7)[1] is a widespread international standard for the message exchange between heterogeneous HISs, it has some well-known limitations for representing clinical knowledge, such as its combined use of structured components and coded terms, which can result in inconsistent interpretations of clinical information [5]. *open*EHR[2] is a foundation dedicated to the research of interoperable EHRs. *open*EHR defined an open architecture based on two-level modeling that separates information from knowledge, in this way addressing some limitations of HL7.

[1] http://www.hl7.org

[2] http://www.openehr.org

Service-oriented architecture (SOA) [6] is an architectural style for distributed applications in which services and service compositions are designed and built in accordance with the service-orientation paradigm. SOA-based environments allow developers to overcome many distributed enterprise computing challenges, such as application integration, transaction management and security policies, while allowing multiple platforms and leveraging numerous access devices, such as sensors and actuators, and legacy systems. In the healthcare domain, SOA can provide consistent information for healthcare professionals, support clinical workflow, and leverage data from multiple departments, facilities and sources, so that different HISs can be integrated in a secure manner, making healthcare more efficient and effective.

Software agents [7] are software entities that employ Artificial Intelligence techniques to choose the best set of actions to be performed in order to reach the goals specified by their users. An agent is a computational entity with autonomous behavior, which allows it to decide and perform its own actions. A software agent is an entity that operates continuously and autonomously in a particular environment, and is capable of intervening in it without requiring constant human attention. Although multiple agents can inhabit a shared environment, each agent must have its own purposes and characteristics for ensuring its autonomy. In the healthcare domain, software agents are particularly useful, since they can help healthcare professionals communicate with each other, which allows these agents to coordinate the daily activities of these healthcare professionals.

1.2 Motivation

Effective and suitable communication between healthcare professionals and patients is vital to provide high quality healthcare, prevent medical errors, and avoid patient harm. In healthcare environment, traditional communication mechanisms based largely on paper records and prescriptions are old-fashioned, inefficient, and unreliable [8]. In an age of electronic record keeping and communication, some healthcare providers are still tied to paper documents, which can be easily mislaid, are often illegible and also easy to forge [9]. When various healthcare professionals and healthcare providers are involved in providing healthcare to a patient, the provided healthcare services are often uncoordinated.

In most countries, the necessary record keeping and communication are difficult to establish because of the highly diverse and decentralized nature of healthcare IT infrastructure. Physicians' offices, clinics and hospitals use computer systems that are not interoperable. In addition, the existence of multiple healthcare systems has resulted in data that cannot be easily translated from one system to another.

Legacy healthcare information systems range from centralized hospital systems, such as electronic medical record, to departmental systems such as laboratory information systems, cardiology systems and emergency systems. Many legacy information systems have been designed for users working in fixed environments, and they normally do not take into account recent advances in information technologies, such as mobile devices [10]. These legacy systems have been operating for many years and are difficult to integrate and modify. Furthermore, the information flow still requires face-to-face conversation or phone calls between human actors in order to perform routine tasks.

Pervasive healthcare has been inspired by the health challenges that most countries will face in the near future. The following challenges have been identified in many studies [11-13], among others:

(i) a huge increase in the ratio of elderly people as compared to the working population has been predicted from the most recent demographic development;
(ii) chronic diseases and healthcare costs are increasing as people grow older;
(iii) some bad habits of our current lifestyle (e.g., smoking, obesity and inactivity) contribute to an increased prevalence of chronic degenerative diseases;
(iv) the constant expansion of the scope of what medicine can do thanks to innovation and advances in medicine and healthcare technologies increases average life expectancy but also healthcare costs;
(v) an increasing lack of clinical professionals is being observed due to retirement and a limited number of medical and nursing students.

In this context, pervasive healthcare has emerged both as a solution to many existing healthcare problems and as an opportunity to provide better healthcare services to an increasing number of people, using limited financial and human resources. In pervasive healthcare it is possible to move from a centralized model with highly specialized medical professional inside hospitals that treat patients, to a much more decentralized model where people themselves are active participants in caring for their own well-being [13]. Advances in the pervasive healthcare technologies can provide support for continuous well-being and care of patients, allowing patient treatment to be moved from hospital to home-based facilities.

Various maturity models for the introduction of technology in organizations have been proposed over time. These models differ mainly in the number of stages, variables and factors for evolution. Each of these models identifies certain characteristics that represent the target at different stages of maturity. Maturity models have been applied to organizations to measure their stage of maturity. From these measurements and evolution prescribed in these models, organizations can build an evolution plan in order to reach higher maturity levels. We consider a maturity model [14] to assess the current position of the healthcare organizations involved in our work and to define roadmaps for improving their position.

This thesis aims at demonstrating the feasibility of using healthcare standards, ubiquitous computing technologies, the service-oriented architecture, and software agents, to develop interoperable applications for exchanging context-aware messages in Pervasive Healthcare environments. In this way, we contribute to the implementation of the Pervasive Healthcare model.

1.3 Objectives

The main goal of this research is to define a Reusable Architecture, supported by healthcare standards, ubiquitous computing technologies, service-oriented architecture and software agents. This Reusable Architecture is applied to support the message exchange and reusability of healthcare information systems overcoming the weaknesses typically found in legacy healthcare information systems, thereby adding value to these legacy systems and making them interoperable. This work addresses the architectural and technological challenges of combining these technologies in order to achieve our goals.

1.4 Research Questions

This thesis starts from the potential benefits of a Reusable Architecture for developing interoperable applications to exchange messages in a Pervasive Healthcare environment. In our work, we adopted the term *Reusable Architecture*, which strongly resembles the term *Reference Architecture* used in the area of software architecture [15]. We used the term Reusable Architecture because we emphasize reusability in our work. Below we pose the research questions we answered in order to identify and clarify the objective of this research. These questions helped break down the communication problems found in healthcare systems into smaller sub-problems, so that by solving these sub-problems we could reach our research objectives. In this thesis we addressed three research questions, which are categorized as *technical research questions (TQ)* and *empirical questions (EQ)* [16].

RQ1 (EQ): *What are the current problems found in the context of Pervasive Healthcare environments?*

RQ1.1: Which stakeholders and with which goals can be identified in the context of healthcare environments?

RQ1.2: What are the problematic phenomena perceived by these stakeholders?

RQ1.3: What are the criteria and requirements that influence the stakeholders' goals?

In RQ1, we elaborated the *empirical questions*, by which we acquired more knowledge about the stakeholders and their goals in the context of healthcare environment. This research question aims at determining the problematic phenomena and the current state of the practice in the context of Pervasive Healthcare environment. Based on the problem investigation we formulate the criteria and requirements for our solution, which contribute to the stakeholders' goals.

RQ2 (TQ): *How to develop a reusable architecture for Pervasive Healthcare environments that supports the requirements imposed by stakeholders?*

RQ2.1: What are the available solutions for message exchange in Pervasive Healthcare environment?

RQ2.2: How to design an architecture for message exchange in Pervasive Healthcare environments?

RQ2.3: What are the typical components of an architecture for Pervasive Healthcare environments?

In RQ2, we elaborated the *technical research questions* to identify the available solutions to the communication problems between legacy systems in Pervasive Healthcare environments. In order to provide a technical solution in its context, we designed a reusable architecture that complies with the requirements imposed by the stakeholders. We explained how the components of this architecture interact with each other in each application instance that can be built based on the architecture.

RQ3 (EQ): *What is the effect of using the architecture on application development based in Pervasive Healthcare environments?*

RQ3.1: What is the effect of applying our reusable architecture when it is instantiated in different applications scenarios in a Pervasive Healthcare environment?
RQ3.2: How to validate the perceived usefulness and ease of use of the applications designed based on our reusable architecture by the stakeholders?
RQ3.3: What is the level of acceptance of the applications designed based on our reusable architecture by the stakeholders?
RQ3.4: How can our reusable architecture be used to support other applications in the healthcare domain in Pervasive Healthcare environment?
RQ3.5: How does our reusable architecture compare with similar solutions?

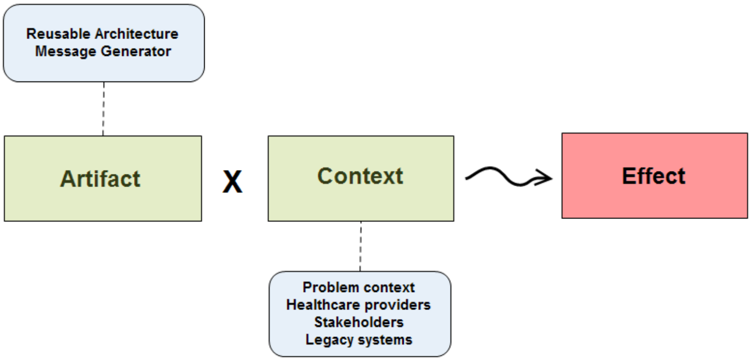
In RQ3, we elaborated the *empirical questions* about the contribution of our reusable architecture to application development, and its effects on the pervasive healthcare environment. We defined our validation criteria based on perceived usefulness and usability in order to evaluate the acceptance of the applications designed based on the reusable architecture in a realistic pervasive healthcare environment. In this research question, we assess the quality trade-offs of our proposed solution in other healthcare application areas.

1.5 Research Design

Our research has followed the design science methodology as described in [17]. We addressed the research questions of this thesis according to the Technical Action Research (TAR) methodology [17-20], which describes the interaction between an *artifact* and a *problem context* for producing *effects*, as shown in *Figure 1-1*. In the context of this thesis, we defined the artifact, context and effects as follow:

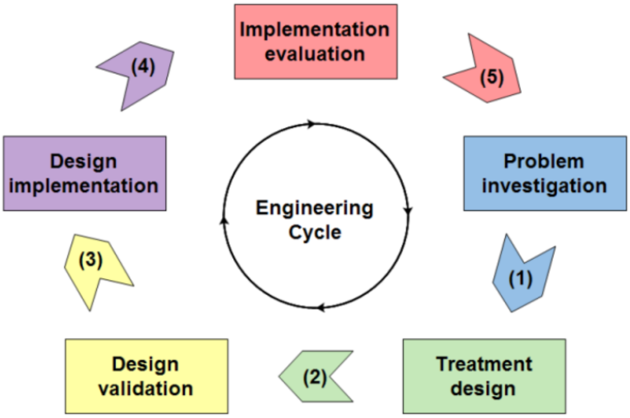
- *Artifact*: Reusable architecture for the exchange of context-aware messages in pervasive healthcare environment;
- *Context*: Several stakeholders, such as physicians, nurses, assistants and patients, and the hospital environment in which the stakeholders use the legacy systems;
- *Effect*: Message exchange between legacy systems using the reusable architecture, which satisfy the requirements to support the interoperability and cooperation between these systems.

Figure 1-1 Design science framework



Inspired by software engineering projects [17-19, 21-24], our research comprises five phases of the engineering cycle, as shown in *Figure 1-2*: (1) *problem investigation*, (2) *treatment design*, (3) *design validation*, (4) *design implementation* and (5) *implementation evaluation*.

Figure 1-2 Research design



We started the *problem investigation (Phase 1)* by examining the potential problem with the messages exchange between legacy systems in our *context*. We studied the problem domain by investigating the problems of interoperability and cooperation existent between legacy healthcare information systems. In this phase, we also identified and studied the stakeholders and their goals in the social context of pervasive healthcare environment, and the current communication mechanisms used by the stakeholders to cooperate to perform their daily tasks. Furthermore, we conducted a comprehensive literature review on technological topics of relevance to this work, like healthcare standards, ubiquitous computing, service-oriented architecture and software agents.

Based on the results of the *problem investigation*, we designed the *treatments*, which consist of an *artifact* that interacts with the *problem context (Phase 2)*. We designed a reusable architecture supported by healthcare standards, ubiquitous computing technologies, service-oriented architecture and software agent technologies. We defined the following requirements of our reusable architecture for message exchange in pervasive healthcare environment, so that the architecture contributes to the stakeholders' goals in the *problem context*:

R0: The reusable architecture must allow stakeholders to exchange messages in pervasive healthcare environments.

R1: The reusable architecture must be easy to use for the stakeholders.

R2: The reusable architecture must be useful for the stakeholders.

We designed our reusable architecture to fulfill our requirements, taking into consideration the different circumstances in which the stakeholders have to perform their tasks. We also designed a message generator for exchange messages in order to guarantee the interoperability with other systems at semantic level.

After the *treatments* were designed, we started the *design validation (Phase 3)* in order to predict whether our reusable architecture satisfies the specified requirements for the *implementation* phase. In order to validate our architecture, we defined three application scenarios:

(1) *delivery of laboratory analysis results*;

(2) *scheduling of an appointment for pacemaker evaluation*; and

(3) *support of a medical staff meeting to prepare for a cardiac surgery*.

These scenarios were defined by interviewing healthcare professionals, in collaboration with IT professionals. The study of these scenarios helped us understand the healthcare process, before an application prototype of our reusable architecture was built. We have conducted a laboratory test in the cardiology domain at three cardiology clinics, one analysis laboratory, and the Cardiology Department of the Santa Casa Hospital of Marília (São Paulo, Brazil) to show the feasibility of our proposed architecture, by using the three realistic scenarios in a pervasive healthcare environment. The Cardiology Department of Hospital Santa Casa defined a project to develop a Cardiology Healthcare Network for several healthcare providers in Marília (São Paulo, Brazil) in order to allow the integration of heterogeneous healthcare information systems in the Cardiology Healthcare Network. In this context, we performed a case study for each of the three scenarios we had designed before with a total duration of 3 months. In these case studies, we collected evidences of how the artifacts were suitable to the scenario specification and validated our reusable architecture. We performed interviews with actual healthcare professionals and patients. The interviews were analyzed to evaluate and validate the feasibility of the applications designed reusing our architecture in each scenario, in order to predict the intention of the healthcare professionals to use the applications. In this phase we performed direct observation and interview with healthcare professionals and patients during their daily tasks.

In the *design implementation (Phase 4)*, we developed three communication systems after we validated our reusable architecture. The communication systems were developed to be tested in a cardiology healthcare environment as our field test. To validate our instantiated communication systems, we performed a field test by applying Clinical Proof-of-Concept [25] with a total duration of 7 months.

In the *implementation evaluation (Phase 5)*, the implemented communication systems were conducted in a real-world setting at the cardiology healthcare environment. The interviews were analyzed to evaluate the perceived usefulness and ease of use of the applications designed reusing our architecture in each scenario, in order to predict the intention of the healthcare professionals to use the applications. In this phase we used the Technology Acceptance Model (TAM) [26], which is known to be an appropriate model to explain the technology acceptance process in the healthcare domain.

Our research design consists of an iterative method. For example, the insights we obtain from our case studies contributed to the further improvement of our architecture, which triggered a new iteration of the five steps shown in *Figure 1-2*.

1.6 Scope

The scope of this thesis is the development of a reusable architecture to support the design and implementation of suitable applications that exchange messages in a distributed healthcare environment. An important premise during the development of our reusable architecture was to maximize the reuse of the capabilities offered by existing healthcare systems, and creating communication interfaces between legacy systems, without losing their original capabilities. To validate our results, we applied our reusable architecture to three different scenarios in the cardiology domain in order to demonstrate the reusability of the architecture.

In this thesis, we do not explicitly address user privacy and security issues. Each country around the world has its own sets of regulations to ensure that healthcare information is kept private and secure, such as the *Personal Information Protection and Electronic Documents Act* (PIPEDA) in

Canada, the *Personal Information Protection Act* (JPIPA) in Japan, the *Personal Data Privacy Ordinance* in China, the *European Union's Data Protection Directive*, the *National Health Service Directives* in the United Kingdom, and the *Health Insurance Portability and Accountability Act* (HIPAA) in the United States [27]. In this work we do not explicitly address compliance with these regulations

We applied standards in the healthcare domain in order to guarantee the syntactic and semantic interoperability between healthcare information systems. However, we do not provide cross-domain interoperability between different domains that may collaborate in the provision of healthcare, such as the interoperation of the healthcare and insurance domains.

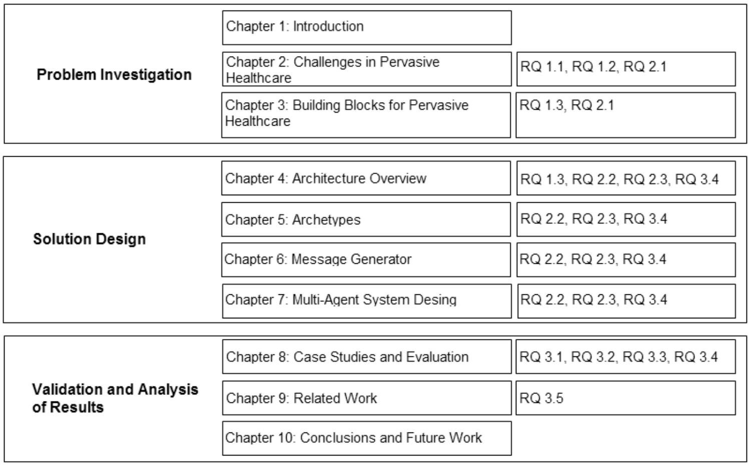
1.7 Structure

The structure of this thesis reflects the adopted research design, as discussed below. *Figure 1-3* presents the structure of this thesis, indicating how the chapters relate to the research question. The remainder of this thesis is structured as follows:

- *Chapter 2 (Challenges in Pervasive Healthcare)* introduces the maturity level of healthcare organizations, provides an overview of the state of practice in applications of information systems in the healthcare domain, describes the major challenges that we found in the communication between legacy healthcare information systems, and explains the current situation and the problems identified in the cardiology domain, as a more concrete representative example;
- *Chapter 3 (Building Blocks for Pervasive Healthcare)* introduces the general concepts and technologies used throughout this thesis, related to ubiquitous computing technologies, healthcare standards, service-oriented architecture and software agents;
- *Chapter 4 (Architecture Overview)* introduces the requirements and the scenarios derived from interviews and defined in cooperation with healthcare professionals. This chapter gives an overview of our architecture for message exchange in pervasive healthcare environments, and discusses the components and the relationships among these components. This chapter also describes the methodology adopted in our thesis;
- *Chapter 5 (Archetypes)* discusses the archetypes that we have designed to contribute with the international archetype development process and how to reuse the archetypes from a central repository, in order to achieve semantic interoperability between different healthcare

information systems, and the archetype development process adopted in our research;

- *Chapter 6 (Message Generator)* describes the message generator that we have designed according to web service interfaces for achieving interoperability in the healthcare domain to support the message exchange among heterogeneous healthcare information systems in pervasive healthcare environments;
- *Chapter 7 (Multi-Agent System Design)* describes in detail the agent-based models developed to support the scenarios that are used in this thesis. This chapter describes the ontology modeling and agent modeling process;
- *Chapter 8 (Case Studies and Evaluation)* evaluates the applicability of our methodology by means of three case studies: (1) delivery of laboratory analysis results; (2) scheduling of an appointment for pacemaker evaluation; and (3) support of a medical staff meeting to prepare for a cardiac surgery;
- *Chapter 9 (Related Work)* gives an overview of related work on data integration in healthcare environment, and compares the related work with our solution;
- *Chapter 10 (Conclusions and Future Work)* gives the conclusions of this thesis by discussing the main results of this work and the limitations of our results. Finally, we identify topics that require further investigation as future work.



Problem Investigation	Chapter 1: Introduction	
	Chapter 2: Challenges in Pervasive Healthcare	RQ 1.1, RQ 1.2, RQ 2.1
	Chapter 3: Building Blocks for Pervasive Healthcare	RQ 1.3, RQ 2.1
Solution Design	Chapter 4: Architecture Overview	RQ 1.3, RQ 2.2, RQ 2.3, RQ 3.4
	Chapter 5: Archetypes	RQ 2.2, RQ 2.3, RQ 3.4
	Chapter 6: Message Generator	RQ 2.2, RQ 2.3, RQ 3.4
	Chapter 7: Multi-Agent System Desing	RQ 2.2, RQ 2.3, RQ 3.4
Validation and Analysis of Results	Chapter 8: Case Studies and Evaluation	RQ 3.1, RQ 3.2, RQ 3.3, RQ 3.4
	Chapter 9: Related Work	RQ 3.5
	Chapter 10: Conclusions and Future Work	

Figure 1-3 Thesis structure: chapters and research questions

Chapter 2

Challenges in Pervasive Healthcare

In recent years, healthcare providers have explored the opportunities of ICT to improve the quality of healthcare and of their managerial processes, while simultaneously reducing their cost. Healthcare information systems (HIS) consist of different sub-systems that are integrated and orchestrated to support care in a patient-centric view of organizations and processes [28, 29]. The application of information systems to healthcare share a number of core challenges, which include the definition of commonly accepted vocabularies and standard data elements to guarantee the interoperability between the healthcare information systems.

This chapter presents the main challenges related to e-health, and it is organized as follows: Section 2.1 discuss the evolution of HIS in terms of a maturity model. Section 2.2 introduces the state of practice of healthcare organizations, which includes understanding the organization, management and use of information in healthcare. Section 2.3 presents the major problems faced by the healthcare professionals for using pervasive healthcare. Section 2.4 explains the current situation and the problems presented in a specific healthcare domain, namely the cardiology domain.

2.1 Evolution of Healthcare Information Systems

Information and Communication Technologies (ICT) in healthcare, also known as *e-health*, has gradually evolved and is concerned with streamlining management and providing information to healthcare professionals. E-health is a broad field, including developments such as computerized diagnostic, decision support systems for medicine based on evidence, Electronic Health Record (EHR) and medical imaging technology [30].

A maturity model shows the change and improvement of an organization over time. Maturity models have proven to be a helpful instrument because these models allow one to better position an organization in terms of its maturity level and help stimulate continuous improvement [31, 32]. In this work, we consider a maturity model to evaluate the current position of the healthcare organizations involved in our work and to define roadmaps for improving their position.

Since the seventies, ICT adoption and maturity have been extensively discussed in the management literature. The concept of maturity in ICT was introduced in [33]. Maturity models for ICT management have instigated extensive discussion, and many researchers have conducted studies to validate them, which led to several extensions of these models [14, 34, 35]. Overall, maturity models provide an overview of the structure of elements that represent the effectiveness of ICT management processes in organizations. In the healthcare domain, maturity models range from specific models, such as the *PACS Maturity Model* [36] to generic models, such as the *Electronic Healthcare Maturity Model* (eHMM) [37, 38].

The application of ICT in healthcare is relatively recent. ICT has been applied in healthcare for no more than five decades, but it brought enormous benefits to healthcare. The following facts were observed [39]:

(i) the shift from paper-based to computer-based processing and storage, as well as the increase of more availability of information in healthcare settings;

(ii) the shift from institution-centered departmental and, later, hospital information systems towards regional and global HIS;

(iii) the inclusion of patients and health consumers as HIS users, besides healthcare professionals and administrators;

(iv) the use of HIS data not only for patient care and administrative purposes, but also for healthcare planning as well as clinical and epidemiological research;

(v) the shift from focusing mainly on technical HIS problems to those of change management as well as of strategic information management;

(vi) the shift from mainly alphanumeric data in HIS to images; and

(vii) the steady inclusion of new technologies, such as ubiquitous computing environments and sensor-based technologies for health monitoring.

2.1.1 Maturity Models

The use of a maturity model allows an organization to have its methods and processes evaluated according to management best practices, against a clear set of external test results. Maturity model assumes the availability of established patterns, created in terms of stages of an organization's development. Typically, the stages are: (1) sequential in nature; (2) occur as a hierarchical progression that is not easily reversed; and (3) involve a wide range of organizational activities [40].

After investigating the use of ICT in large organizations in the US, Nolan proposed an evolutionary model initially containing four stages of maturity, and later two more stages have been added to the initial model [33]. The *Revised Model of Maturity Stages* [41] was the most comprehensive and detailed maturity model published so far, and gained consensus, because it defined seven maturity influence factors with equal importance. This model assumes that an organization may find itself at different maturity levels under different influence factors, such as strategic approach, business culture, information technology and process management [42].

The definition of development and growth stages is still being extended and applied in organizations. There are several examples of maturity models that focus on different organizational and ICT fields, such as *Capability Maturity Model Integration* (CMMI), for software development process [43], and the *Picture Archiving and Communication Systems* maturity model (PMM) [36], for medical imaging technology maturing.

CMMI inspired the development of dozens of other maturity models, which were developed and applied in various domains and contexts. CMMI is a process model that provides a clear definition of what an organization should do to promote behaviors that lead to improved performance. *Figure 2-4* shows the five maturity levels defined in CMMI as the most important elements that are required to produce or deliver better products and services [44].

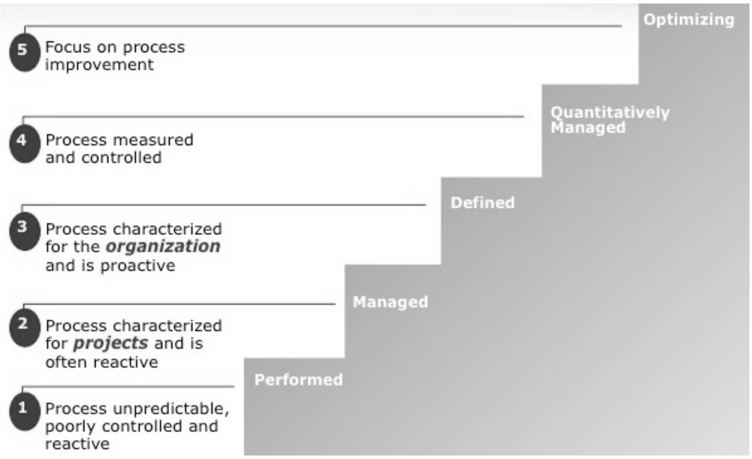

Optimizing
5 Focus on process improvement
Quantitatively Managed
4 Process measured and controlled
Defined
3 Process characterized for the ***organization*** and is proactive
Managed
2 Process characterized for ***projects*** and is often reactive
Performed
1 Process unpredictable, poorly controlled and reactive

Figure 2-4 CMMI Maturity Levels (Source: Sang and Lin, 2009)

The use of maturity model allows: (i) to provide a way of measuring the current situation of an organization by means of maturity level indicators; (ii) to facilitate the establishment of an improvement process that best suits the organization, while remaining within the prescribed best practice parameters of the particular domain; and (iii) the organization to quickly understand the benefits of current processes and can proactively understand what benefits to expect in the future [45, 46].

2.1.2 Healthcare IT Maturity

Healthcare organizations initially have realized that their fundamental problem was the lack of technological infrastructure and an inability to properly manage the healthcare processes. An analysis of the current healthcare context shows clearly the impact of introducing IT in healthcare environments. However, the benefits of modern technology and better methods and tools in healthcare cannot be achieved in undisciplined healthcare processes. These are the reasons why maturity models are useful to manage ICT in healthcare organizations [37, 47].

An immature healthcare organization can be defined as reactive in nature, with personnel who usually concentrate on taking care of issues; it absences of destination premise basis for judging product (service) quality or for tackling product or process problems; and offers healthcare product quality.

In contrast, a mature facility can be characterized as an organization that has the capacity to manage healthcare quality and expense adequacy; in which the healthcare processes ordered are reliable with the ways the work accomplishes really; where improvements are obtained through controlled pilot-tests and/or expense profit investigation; and with managers who are capable of checking the quality of the healthcare services and customer satisfaction.

Understanding the level of capability of IT in healthcare organizations is a challenge in the context of healthcare nowadays. Maturity models have been proposed in the literature for healthcare IT management, such as the *Electronic Healthcare Maturity Model* (eHMM) [37], *International Data Corporation* (IDC) *Health Insights Maturity Model* [48], *United Kingdom National Health Service* (NHS)[3] and *Electronic Medical Record Adoption Model* (EMRAM)[4], which was introduced by the *Healthcare Information and Management Systems Society Analytics* (HIMSS).

[3] http://www.connectingforhealth.nhs.uk/systemsandservices/nimm

[4] http://www.himssanalytics.org/emram/

2.1.3 Electronic Medical Record Adoption Model (EMRAM)

An Electronic Medical Record (EMR) has been defined as a digital version of health-related information on an individual that is created, gathered, managed, and consulted by licensed clinicians and staff from a single organization who are involved in the individual's healthcare [49]. Electronic Healthcare Record (EHR) consists of aggregating electronic records of health-related information on an individual that is created and gathered cumulatively across more than one healthcare organization and is managed and consulted by licensed clinicians and staff involved in the individual's healthcare [50].

HIMSS aimed at IT decision-makers who are responsible for delivering and running clinical e-health systems as well as more traditional IT systems [51]. The results of HIMSS provide a solid foundation to trigger senior level decisions within the hospital to benefit from constrained IT organizational capabilities, which are considered essential for delivering and running better healthcare information systems and services. HIMSS identifies the levels of electronic medical record (EMR) capabilities [52]. HIMSS Analytics Electronic Medical Record Adoption Model (EMRAM) incorporates methodology and algorithms to automatically score the more than 5.400 U.S. and approximately 650 Canadian hospitals in the HIMSS Analytics database relative to their EMR capabilities [53].

Ranging from limited ancillary department systems through a paperless EMR environment, EMRAM scores provide peer comparisons for hospital organizations as they strategize their path to implementing a complete EMR and participation in an EHR. The stages of the model are as follows:

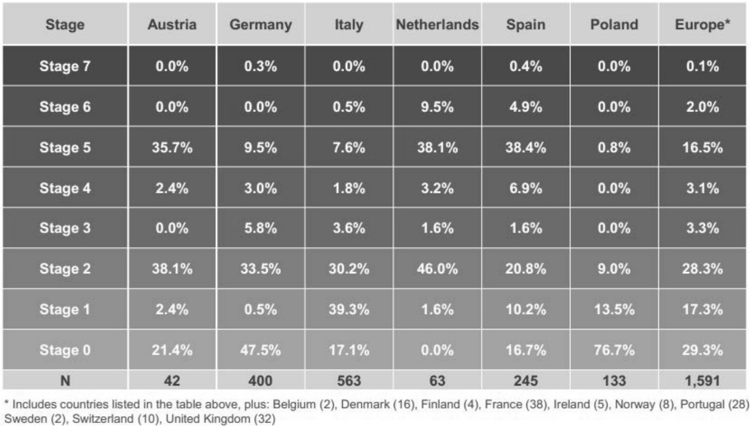
- Stage 0: some clinical automation may exist. The healthcare provider has not installed the ancillary clinical systems (laboratory, pharmacy and radiology);
- Stage 1: all three major ancillary clinical systems are installed, namely laboratory, pharmacy and radiology;
- Stage 2: major ancillary clinical systems feed data to a clinical data repository (CDR) that provides access to physicians for retrieving and reviewing results. CDR contains a controlled medical vocabulary (CMV) and the clinical decision support system and rules engine for rudimentary conflict checking;
- Stage 3: clinical documentation is available, including vital signs, nursing notes and care planning charts. The first level of clinical decision support is implemented to conduct error checking with order entry, such as drug/drug, drug/food, drug/lab conflict checking normally performed by the pharmacy. Some level of medical image access from a picture archive and communication systems (PACS) are

available to be accessed by healthcare professionals via an intranet or other secure networks;

- Stage 4: computerized physician order entry (CPOE) for services (e.g., laboratory, medication prescription and surgery room) is assigned to the nursing documentation and CDR environment. Second-level of clinical decision support related to evidence-based medicine protocols are implemented;
- Stage 5: the closed loop medication administration environment is fully implemented in at least one patient care service area. The electronic medication administration record and bar coding or other auto-identification technology are implemented and integrated with CPOE and pharmacy to maximize point-of-care patient safety processes for medication administration;
- Stage 6: full physician documentation is implemented for at least one patient care service area. A full environment of PACS systems is implemented, i.e., all film-based images are available to physicians via an intranet or other secure network;
- Stage 7: the healthcare provider has a paperless EMR environment: clinical information can be readily shared via continuity of care (CCD) electronic transactions with all entities within healthcare networks, such as hospitals, ambulatory clinics, sub-acute environments, employers, payers and patients; IT systems allow the healthcare provider to support the true sharing and use of health and wellness information by consumers and providers alike; and healthcare providers use data warehousing and mining technologies to capture and analyze care data, and improve care protocols via decision support.

According to the HIMSS Analytics Database[5] (2012), the majority of European hospitals are in the early stages of EMR adoption. Currently 29 percent of European hospitals have not achieved Stage 1 and are at Stage 0, 17 percent have achieved Stage 1, 28 percent have achieved stage 2, approximately three percent have achieved stage 3, approximately three percent have achieved Stage 4, approximately 16 percent have achieved Stage 5 and merely two percent of hospitals have achieved stage 6 and stage 7, as shown in *Figure 2-5*.

[5] http://himss.eu/analytics



Stage	Austria	Germany	Italy	Netherlands	Spain	Poland	Europe*
Stage 7	0.0%	0.3%	0.0%	0.0%	0.4%	0.0%	0.1%
Stage 6	0.0%	0.0%	0.5%	9.5%	4.9%	0.0%	2.0%
Stage 5	35.7%	9.5%	7.6%	38.1%	38.4%	0.8%	16.5%
Stage 4	2.4%	3.0%	1.8%	3.2%	6.9%	0.0%	3.1%
Stage 3	0.0%	5.8%	3.6%	1.6%	1.6%	0.0%	3.3%
Stage 2	38.1%	33.5%	30.2%	46.0%	20.8%	9.0%	28.3%
Stage 1	2.4%	0.5%	39.3%	1.6%	10.2%	13.5%	17.3%
Stage 0	21.4%	47.5%	17.1%	0.0%	16.7%	76.7%	29.3%
N	42	400	563	63	245	133	1,591

* Includes countries listed in the table above, plus: Belgium (2), Denmark (16), Finland (4), France (38), Ireland (5), Norway (8), Portugal (28) Sweden (2), Switzerland (10), United Kingdom (32)

Figure 2-5 EMRAM Scores, % Hospitals, Q4/2013 (Source: HIMSS Analytics Database, 2012)

According to the HIMSS Analytics Database[6] (2013), the majority of Canadian hospitals are in the early stages of EMR adoption. Currently, approximately 45 percent of the Canadian hospital market has some level of EMR applications installed to support care delivery (stage 2 or higher). A further evaluation of the market shows the percentage of EMR adoption according to their stage as shown in *Table 2-1*.

Table 2-1 HIMSS Adoption Maturity Model (Source: Data from HIMSS Analytics Database, 2013)

Stage	Cumulative Capabilities	2013
Stage 7	Complete EMR; clinical information can be readily shared via standardized electronic transactions; data warehousing; data continuity for all hospital services	0.0%
Stage 6	Physician documentation (structured templates); full Clinical Decision Support (CDS), full radiology PACS	0.6%
Stage 5	Closed loop medication administration.	0.0%
Stage 4	Computerized Practitioner Order Entry (CPOE); CDS is related to evidence based medicine protocols (clinical protocols)	3.8%
Stage 3	Clinical documentation (flow sheets), first level for CDS (error checking); PACS available outside Radiology; full healthcare information exchange (HIE)	32.2%
Stage 2	Clinical Data Repository (CDR); CDR contains a controlled medical vocabulary; CDS with rudimentary conflict checking; Document Imaging may be supported	29.1%
Stage 1	All three ancillaries systems are installed	14.5%

[6] http://www.himssanalytics.org/data/annualStudy.aspx

Stage 0	At least one of three ancillary systems is not installed. Healthcare organizations are not able to participate in an EHR initiative without manually entering healthcare information into the EHR system	19.8%

The consolidated rate of Canadian healthcare organizations in stage 1 and stage 2 implies that major ancillary clinical systems are installed and they encourage electronic healthcare information for healthcare experts in healthcare environments. Once the healthcare information exchange is completely operational, it will empower healthcare information to be shared electronically among patients, healthcare professionals and healthcare providers.

According to the HIMSS Analytics results, the great amount of work and investment must be done by healthcare organizations to implement clinical systems to empower their cooperation in EHR initiatives. All the more significantly, further implementation of higher stage healthcare applications will empower the diminishment or elimination of medical lapses, while digital environment offers these profits. The higher stages of the model reflect enhanced patient care, as well as improvements in productivity and adequacy with which patient care services are conveyed by clinicians.

In our research, we have investigated healthcare organizations that operate with ancillary clinical systems, but have not yet achieved the semantic interoperability level necessary to exchange healthcare information between healthcare systems. We have considered healthcare organizations at the stages 1-3 of the HIMSS adoption model. These organizations aim to deliver the capabilities focusing on sharing patient information among all of the stakeholders.

2.2 State of Practice

Traditionally, the healthcare environment is quite complex because it involves a huge amount of stakeholders such as physicians, nurses, pharmacists, laboratory technicians who participate in the treatment of patients, so that each of these stakeholders have different goals and generate information needed by the others. Clinical information in the healthcare domain is also complex due to the dynamic nature of healthcare and because it covers distinctive sorts of data, such as clinical data, laboratory data and patient administration.

Attaining a completely paperless environment within a healthcare environment is more than likely an improbable desire. Maturity model should be a continuous solution concentrated on the strategic removal of paper from the healthcare organizations. Paper-based systems have been used within healthcare organizations and these systems are inefficient for dealing with the measures of clinical information that can affect patient forethought. For example:

- The conventional medical record may be illegible because it is hand written and poorly organized, making it difficult for physicians to locate the information they need about past medical reports and their results [54, 55];
- Healthcare professionals must often care for patients without any historical clinical information whatsoever. Patients frequently do not know or cannot communicate their clinical information, and current healthcare processes tend to fragment the typical patient's records among several hospitals, laboratories and medical clinics, which can cause delay, disruptions or errors in patients' healthcare [56, 57];
- Patients often do not have access to accurate and reliable information or are not able to retrieve, understand and use this information to meet their individual needs [58].

The current e-health standardization has aimed at achieving interoperability to exchange information in the healthcare domain. Interoperability is generally defined as the ability of different information systems and software applications to communicate, to exchange data accurately, effectively, and consistently, and use the information that has been exchanged [59]. Interoperability in the healthcare domain can be investigated in different contexts, such as in the information exchange between healthcare applications, EHRs or patient identifiers [60].

Interoperability can be classified as syntactic and semantic interoperability. *Syntactic interoperability* only ensures that the message is readable by the receiving system, but it does not guarantee that the content of the received information can be understood and processed by the receiving system. Therefore, to guarantee interoperability, semantic interoperability must be provided. *Semantic interoperability* can be defined as the ability of two or more computer systems to exchange information and automatically interpret the meaning of this information accurately enough to produce useful results to the end users of both systems [61, 62].

Healthcare systems associate some specific meaning to the concepts they use during their operation. However, since each system places its own meaning on concepts, problems may occur when systems are expected to interoperate with each other because these meanings may be disparate [63]. One way to address this problem is to apply standards that encode the meaning of the concepts. These standards must be structured in such a way that can be understood by different computers, but also by humans.

Nevertheless, as beneficial as semantic interoperability is to healthcare, at present, it is largely an unreached goal. This is chiefly because the information systems used within the healthcare environments have a large number of heterogeneous, distributed, and mostly proprietary models for representing and recording patients' information [64, 65].

Numerous solutions have been proposed to achieve interoperability in the healthcare, with different degrees of success. These include the use of standards, ontologies, web services and archetypes [60, 66, 67]. However, the effective and meaningful information exchange of healthcare information is yet to be achieved across healthcare information systems.

2.2.1 Interoperability Challenges

In healthcare scenarios, clinical information can be transmitted without the overhead of express semantics representation. For instance, a specialist can return results of a laboratory test without sending the descriptions of the equipment used to produce the test results. Information exchange can cause interoperability problems in healthcare, because the clinical information without the semantic model can be misinterpreted or overlooked. For example, the format for describing test results used by a laboratory might be so different from the format used by a clinic where the patient is being treated that results may be misinterpreted, due the issues related to the reference ranges on different equipment designed to perform the same measurements as well the constantly changing reference ranges. Incorporation of these data and semantic models in the information transferred between systems enables automation to interpret the data and to translate them in each specific system.

The goal of standardization is to provide consistent specifications to be shared by all parties manufacturing the same products, or providing the same services [68]. Standards are being developed in different domains, in order to have a common concept on syntax and semantics to ensure interoperability in each specific domain. The major goal of standards in the healthcare domain is to improve patient care by allowing interoperability between heterogeneous systems. In the healthcare domain, information is expected to be exchanged in the form of standardized reports.

Standards have been developed in healthcare domain, such as the *HL7 Clinical Document Architecture* (CDA)[7], *open*EHR *dual model*[8], *CEN ENV 13606 Electronic Healthcare Record Communication* (EHRcom)[9], *Digital imaging and communication in medicine* (DICOM)[10], *International Classification*

[7] http://www.hl7.org/

[8] http://www.openehr.org/

[9] http://www.en13606.org/

[10] http://medical.nema.org/

of Disease (ICD)[11]. Healthcare organizations, however, often do not conform to a single standard, and the use of multiple standards can also hamper interoperability.

Standard vocabularies provide the means for organizations to exchange healthcare information. Existing efforts aimed at achieving semantic interoperability within the healthcare domain rely on agreements about the meaning of concepts represented in terminology systems such as nomenclatures, thesauri, or ontologies [69]. However, the growth of incompatible heterogeneous terminology and ontologies in healthcare domain actually make the problem of interoperability worse, e.g., due to possible multiple representations of the same clinical concepts [70].

2.2.2 Legacy Healthcare Information Systems

In computing, 'legacy' denotes hardware and software that have been superseded but are difficult to replace because of their widespread use [71]. Mainly, a legacy software system is often considered as any existing software artifact that cannot be discarded or easily converted to new technologies or re-engineered due to its unstructured design. Legacy systems that have been designed with outdated technology compensate for the dominant part of the systems in numerous client application situations [10].

Legacy systems are frequently hard to improve, replace and maintain in light of the fact that there is a general absence of understanding of how these systems have been designed. The designers of a legacy system may have left the organization, leaving nobody ready to clarify how it works. This circumstance is often aggravated throughout the years by insufficient documentation. Moreover, most legacy systems were designed using outdated programming languages and paradigms. Legacy system integration with more current systems may also be troublesome in light of the fact that new systems may use distinctive technologies.

Legacy healthcare information systems implemented prior to the introduction of common standards have limited interoperability capabilities, and these systems are still in use today. Their data storage and data structure are specific and often proprietary. Further, most of these legacy systems have been designed for a single task. Many of these systems have been originally designed so that interoperability with other vendors' applications is not straightforward, in order to ensure market share and to encourage their purchase by healthcare facilities.

Large healthcare institutions (hospitals) have made significant investments in computer systems, but paper-based records and fax-based communication is still the norm in many places, where computers are used

[11] http://www.who.int/classifications/icd/en/

primarily for administrative functions [72]. It is necessary to modernize the legacy systems by providing an integration solution that allows them to communicate with each other in order to accomplish their semantic interoperability [73-75].

The integration of legacy systems with other systems of the healthcare network is one of the requirements to achieve interoperability in the healthcare domain. This is because legacy systems are in operation and it is expensive to replace them, and thus they cannot be isolated from other systems. A middleware system can be used to achieve the interoperability between healthcare systems by performing the necessary message translations.

2.3 Consequences of Pervasive Healthcare

Pervasive healthcare may be defined from two perspectives: (*technological*) as the application of pervasive computing technologies to healthcare, and (*societal*) as the provision of healthcare anywhere, anytime and to anyone [76, 77]. In essence, to be effective pervasive healthcare must simplify daily activities with open standards-based applications. It must enable healthcare professionals be more efficient, leaving them with more spare time in their daily tasks [78].

Pervasive healthcare is closely related to ubiquitous computing, which is often mentioned as a potential fosterer for improving healthcare [79-81]. While the term *pervasive* stands for the tendency to expand or permeate, *ubiquity* is the property of being omnipresent. In this sense, the goal of pervasive healthcare is to become the means for achieving ubiquitous healthcare [82-84]. Pervasive healthcare can benefit from ubiquitous computing technologies by: (i) enabling distributed computing and processing of health information, (ii) decreasing healthcare costs by providing fast and appropriate care to patients, (iii) changing into a distributed model, in order to produce faster responses and to allow patients to better manage their own health.

Nowadays, modern biomedical data acquisition devices and the deployment of fast wireless networks have enabled the introduction of several pervasive healthcare applications.

2.3.1 General Requirements

In pervasive healthcare environments, the way a user interacts with an environment, with other people and with computers becomes more intensive. The way in which people interact is enriched with a hybrid mix of communication technologies and interaction devices [85]. Proactivity and transparency should be balanced during the interaction. To accomplish this balance, a system should be able to infer by observing user behavior and context information.

The pervasive healthcare model requires a reliable wireless infrastructure in order to provide access and exchange of clinical information. Many of the existing and emerging wireless networks, such as cellular-oriented, wireless LAN, satellites and short-range technologies such as *Bluetooth* and *Radio Frequency ID* (RFID) can support the exchange of information in healthcare environments [86-89].

A number of situations in pervasive healthcare could be covered by using wireless communication technologies in a context-aware environment. This includes *passive* context-awareness, where the system becomes aware of its context but does not adapt to the context changes, and *active* context-aware in which the system adapts to context changes. Wireless healthcare monitoring system should be actively context-aware, which should help improve decision making by healthcare professionals on the patients' current conditions and healthcare needs. In healthcare monitoring, users of pervasive healthcare systems can be patients whose primary role is being monitored for one or more health conditions and receive the necessary medical attention, or healthcare professionals, whose primary task is to receive accurate information on the monitored patients and take decisions based on these conditions and the patients' required care [86].

One of the challenges in healthcare monitoring using wireless networks is due to the reliability of message exchange. Privacy and security issues are also potential problems, as healthcare information should be available only to authorized persons. Privacy and security have not yet been properly dealt with in relation to legacy healthcare information systems [90, 91].

The usability of pervasive healthcare environments is another challenge, at least in the near future. People who are less technically knowledgeable are generally willing to use pervasive devices if these devices facilitate their daily tasks, give them more independence, and offer intuitive interfaces [92].

2.3.2 Challenges

In a pervasive healthcare environment, the amount of actual end-users and devices tends to increase, as well as the interactions between them. A huge number of applications are distributed and installed separately for each device class, processor family, and operating system. As the number of devices grows, these applications tend to become unmanageable. In this sense, pervasive healthcare should be introduced to hide the heterogeneity of the underlying technology.

These challenges encompass several issues:

(i) sensing, computing and communication functions need to become *invisible* technologies within the environments as part of daily life, according to Weiser [93, 94];

(ii) the high degree of *availability* and *quality* of data transmissions must be offered by applications where the timely delivery and processing of sensor data is crucial;

(iii) the *scalability* is a crucial problem in pervasive healthcare, since the number of interactions between devices increases as more users are involved;

(iv) the *heterogeneity* of technologies and infrastructures involved in healthcare environments requires the reduction of IT complexity applied in these environments to overcome the amount of pervasive technologies; and

(v) the *smart environments*, consisting of embedded computers and multi sensors that should adapt to the changes in an environment, recognizing end-users and providing services to them.

Challenges in pervasive healthcare are about understanding healthcare issues from a technological and societal point of view and about designing and assessing technologies that help integrate healthcare more seamlessly into daily life. Designing such systems is a complex process, requiring an interdisciplinary approach and team collaborations. In pervasive healthcare environment, the benefits might easily overcome the challenges. For example, we expect that most patients are willing to give away part of their privacy for their well-being on medical treatment; however this must not lead to disadvantages outside this context.

2.4 Example: Cardiology Domain

Our research has been inspired by difficulties identified in the cardiology domain, which is one of the best examples of the importance of teamwork. During the course of our research, we analyzed a hospital stay in which a cardiac patient may interact with several stakeholders, such as physicians, nurses and technicians. Effective and efficient clinical practice thus involves many circumstances where critical information must be accurately communicated. In our research, we observed that when the cardiology healthcare professionals do not communicate effectively, patient safety is at risk for a number of motives, such as misinterpretation of information, absence of critical information, and unclear orders over the telephone between healthcare professionals. Stakeholders' collaboration is essential in healthcare environments, and particularly in the cardiology domain.

Modern healthcare professionals use technology to support their decisions in order to ensure accuracy and treat patients effectively and efficiently. However, sometimes human-error is a decisive factor in patient care decisions. This is one of the main reasons for inaccurate diagnosis. In the cardiology domain, inaccurate and inefficient diagnosis can lead to awful consequences [95, 96]. Therefore, cardiology is a delicate area that needs an even higher accuracy level because inaccuracy in this area may incur in life threatening situations.

In order to illustrate this we considered the example of a cardiovascular surgery. Due to the complexity of the resources involved, cardiovascular surgery requires the full integration of individual efforts with maximum efficiency to make sure that each action plan is performed successfully. Cardiovascular surgery is performed by a work group of highly trained staff, termed the *Heart Team* [97], consisting of:

- *Cardiovascular surgeon*, who leads the surgical team;
- *Assistant surgeon*, who provides aid in exposure, hemostasis, closure, and other intraoperative technical functions that help the cardiovascular surgeon carry out a safe operation with optimal results for the patient;
- *Cardiovascular anesthesiologist*, who administers the drugs needed to keep patients asleep during surgery;
- *Perfusionist*, who operates the cardiopulmonary bypass machine; and
- *Cardiovascular nurses*, who are specially trained to assist during cardiac surgery. Prior to initiation of anesthesia, most cardiac surgery patients undergo the insertion of a peripheral intravenous catheter, an arterial line, and a pulmonary artery catheter, and cardiovascular nurses are responsible for this.

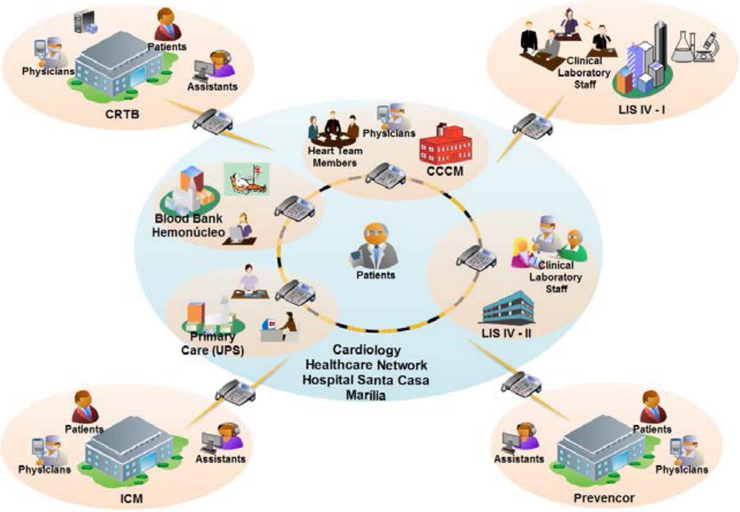
Diagnosis of cardiovascular diseases is complex since combinations of tests have to be used to diagnose such a disease. Resources such as clinical details, electrocardiogram, echocardiography, stress testing, coronary angiography, cardiac catheterization, blood pressure, pulse rate, cholesterol level and blood test reports, and even patients' history, gender and age all are important when diagnosing a cardiovascular disease.

2.4.1 Cardiology Healthcare Network

Patients and healthcare professionals are constantly being moved from one location to another, and rapid access to information can become vital at any time of day in healthcare environments. Healthcare providers understand that IT networks, data storage and data management have become more critical than ever, and collaboration with other healthcare providers is no longer an option, but rather a necessity. In order to provide quality healthcare to everyone in the communities they serve is necessary a network of healthcare providers.

The Cardiology Healthcare Network of Santa Casa Hospital Marília consists of a collaboration among healthcare providers and provides ongoing follow-up care for cardiovascular diseases in city of Marília and region, as illustrated in *Figure 2-6*. The Cardiology Healthcare Network was designed so that healthcare professionals provide high quality healthcare to everyone in the communities they serve. In this healthcare network, the CCCM is the Cardiac Department of the Santa Casa Hospital, in which the healthcare professionals are committed to preventing and treating cardiovascular diseases to their patients in a collaborative environment. *Figure 2-6* shows that to achieve their goals, the CCCM's healthcare professionals provide an array of healthcare management services involving many providers, such as the blood bank (Hemonúcleo), clinical laboratories (LIS-IV I and LIS-IV II) and primary care (UPS) of the Santa Casa Hospital; two cardiology clinics (Prevencor and ICM), which provide support to perform tests to diagnose cardiovascular diseases; and CRTB clinic, which provides support to patients that carry pacemaker.

Figure 2-6 Cardiology Healthcare Network



In the Cardiology Healthcare Network, as we observed in the Santa Casa Hospital Marília, the healthcare information systems and the plain old telephone system (POTS) were used to exchange information amongst the stakeholders. In this distributed healthcare environment, all the resources must be gathered in order to diagnose heart diseases efficiently. Several scenarios in the cardiology domain still require that particular outcomes are achieved by communicating via the phone, for example, in the following circumstances:

(i) a CCCM member calls a blood bank member to obtain information about the availability of blood stocks;
(ii) a CCCM member calls an LIS-IV II member to obtain information about the availability of clinical laboratory results;
(iii) a CCCM member calls a surgical center member to confirm the availability of an surgery room;
(iv) a CCCM member calls the relevant cardiology clinics (ICM and Prevencor) to request medical reports about the patient's EHR;
(v) a CCCM member calls his staff members to make an appointment to discuss a patient's cardiovascular surgery;
(vi) a CCCM member calls a patient to make an appointment for a cardiovascular surgery to be performed; and
(vii) a CRTB member calls a patient who carries a pacemaker to make an appointment for a pacemaker evaluation.

The Cardiology Healthcare Network has faced many challenges related to an increasing volume of patients and information. Moreover, an aging population has aggravated the problems in the cardiology sector by hypertension, diabetes mellitus and cardiac arrhythmias [98]. The cardiology network is committed to improving patient care and healthcare management, thereby making the patients central in the healthcare process, who should receive the best healthcare possible.

2.4.2 Problems in the Cardiology Healthcare Network

We investigated the healthcare organizations and legacy healthcare information systems that work within the Cardiology Healthcare Network in order to better understand the daily tasks and organizational processes. In the daily tasks of the Cardiology Healthcare Network, we observed several problems:

(i) the CCCM information system does not allow the Heart Team members to know when the health resources become available to prevent and treat cardiovascular diseases, such as blood stocks and surgery rooms;
(ii) the healthcare professionals who use the legacy healthcare information systems in the cardiology network spend more than necessary time searching, retrieving, consulting, and interchanging clinical information about their patients;

(iii) in synchronous communication forms, such as in a phone call, the participating parties must be both present; and therefore, time is wasted if one party cannot reach the other;

(iv) because of the time-consuming use of distributed legacy healthcare information systems and phone calls in many scenarios, the Heart Team members have less time available for their medical activities;

(v) even though the medical reports are available online in several cardiology clinics, the quality of these reports depends on human factors. For instance, any mistake made by an assistant, such as exchanging records of two namesake patients when requesting medical reports from a physician, may have dramatic consequences for patients;

(vi) assistants are responsible for the scheduling of cardiovascular treatments and the notification of appointments dates to patients. If patients miss their appointments, time is wasted, obstructing the provision of healthcare and decreasing the number of patients that can be serviced;

(vii) on a patient's arrival, if healthcare professionals do not have the patient clinical history at hand, clinical staff and healthcare resources are wasted;

(viii) following a treatment is essential to preventive care, and it has been a huge problem in this complex healthcare environment, resulting in a risk of post-discharge recovery complications and duplication of efforts (i.e., the same patient may be readmitted for the same treatment in other medical clinic).

The problems identified above are a consequence of inadequate communication between the legacy systems in the Cardiology Healthcare Network. These problems characterize the so called *automation gap* [99, 100], which typify the lack between the current practices and the desired maturity level. A technological support that automates error-prone human processes is necessary to solve the above-mentioned problems.

Chapter 3

Building Blocks for Pervasive Healthcare

Pervasive healthcare has become one of the most active fields of research within ubiquitous computing. In our research, we selected a set of technologies that can play the role of building blocks for pervasive healthcare: (i) ubiquitous applications that use context-aware computing to enable the transparency of the computers that pervade the everyday life, (ii) healthcare standards for sharing and reuse of information from many healthcare domains, (iii) Service-Oriented Architecture, which offers capabilities to support reuse and sharing of resources across the healthcare organizations, and (iv) agent technologies, which assist healthcare providers and patients in performing activities in their daily lives. This chapter defines the basic terminology and fundamental concepts of these technologies, which we have used in this thesis.

This chapter is organized as follows: Section 3.1 discusses the ubiquitous computing technologies applied in this work. It also discusses the notions of context and context-aware computing, which are strongly associated with this line of research. Section 3.2 discusses the state-of-the-art in health-related standards. Section 3.3 presents the *open*EHR dual model approach. Section 3.4 explains the Service-Oriented Architecture concepts. Section 3.5 discusses the use of software agents in application infrastructures.

3.1 Ubiquitous Computing

The term *ubiquitous computing* was introduced by Mark Weiser (1991) [93] to denote what was called the third wave of computing. In the early years of computers, the *mainframes* were stand-alone devices shared by many people. This first wave of computing ended with the introduction of *personal computers*. The second wave, which is currently fading out, is characterized by one to one relationships between the user and computers. Ubiquitous computing was then the next wave, in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone 'using' ubiquitous computing engages in interaction with many computational devices and systems simultaneously, and may not necessarily even be aware that he is doing so.

Weiser envisioned that our interaction with computer technology would no longer be limited to the personal computer and conventional input and output devices, but that we would have constant interaction with multiple interconnected computers and sensors embedded in rooms, furniture, clothes, tools and other items that surround us. In this way, persons, places, and physical objects in the world would become potential elements of computer interaction. By enabling computers embedded in the physical environments to sense and adapt to events occurring in the real world, that is, making them context-aware, they would effectively become seamlessly integrated with our daily activities.

3.1.1 Ubiquitous Computing Dimensions

Various terms have been used to refer to one of the lines of research that this work follows, such as *Ubiquitous computing*, *Pervasive computing*, *Context-aware computing*, and *Mobile computing*. Lyytinen and Yoo [101] suggest that there are conceptual differences between these terms, and each term employs different ideas about how computing services should be managed and organized according to the level of embeddedness and mobility, as shown in *Figure 3-7*. *Mobile computing* is fundamentally about increasing our capability to physically move computing services with us. Another dimension in *Figure 3-7* is *Pervasive computing*, which is based on the idea of embedding computer technology in the physical environments in which we move, and that these embedded computers are able to detect and respond to changes in their local environments. *Ubiquitous computing* builds on the ideas of high device mobility and that the applications running on these devices are able to adapt to changing environments as they are carried around by users. Ubiquitous computing is characterized by physical and cognitive embedded systems, networking, ubiquity, and context-awareness.

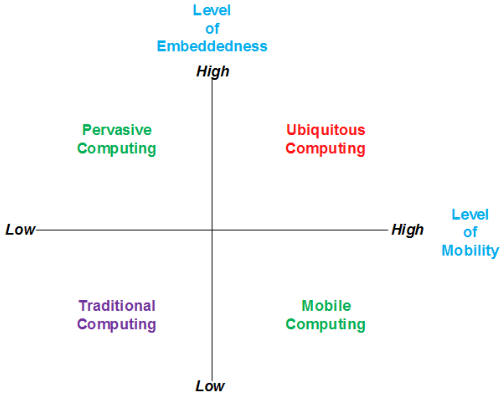


Figure 3-7 Dimensions of Ubiquitous computing (Source: Lyytinen and Yoo, 2002)

The ultimate goal of ubiquitous technologies is to implement *calm technologies*, a world where computers do not cause stress, but enhance our lives and facilitate many tasks [102]. Calm technologies are those that stay in the background and get invisible for their users. Mostly they provide important services to the user, but simultaneously allow the user to forget about them and therefore are quite powerful.

The following features characterize ubiquitous computing: decentralization or modularity of the systems and their widespread network; embedding of the computer hardware and software in equipment and objects of daily use; mobile support for the user through information services anywhere and anytime; context-awareness and adaptation of the system to current information requirements; automatic recognition and autonomous processing of repetitive tasks without user intervention [103, 104].

3.1.2 Technologies

Ubiquitous computing utilizes the whole range of modern information and communication technologies. Hence, advances in communication technology, microelectronics, energy supply, user interfaces, information security, sensors and localization technology are of particular significance for ubiquitous computing. Communication technology, in particular mobile communication, is considered the key technology in ubiquitous computing. Besides consolidated technologies like Ethernet and UMTS, increasingly self-organizing *ad hoc* networks and a number of powerful, close-range technologies are available nowadays [105].

Ubiquitous computing has a considerable economic potential to increase efficiency and thus competitiveness. In contrast, ubiquitous computing enables a great number of new services, whose usefulness for citizens and economic viability must still be demonstrated. In order for this potential to be actually accomplished, however, a number of pre-conditions must be fulfilled [103, 106]:

(i) ubiquitous computing requires the installation of devices in the environment, like sensors and actuators. Thus, device standardization is needed to allow that ubiquitous computing projects should continue justifying the investments and;

(ii) creation of timely access to ubiquitous computing technology for medium-sized companies and their participation in the standardization processes. This not only ensures the realization of network externalities but also helps prevent a domination of large, often globally active enterprises.

Due to their general character, many applications of ubiquitous computing are possible in various domains. This study concentrates on applications in healthcare environments. We expect that an increase in process efficiency and productivity will limit the costs in the healthcare system, and ubiquitous computing can open up possibilities for quality improvements in healthcare. The proponents of pervasive healthcare see possible applications in diagnostic, therapeutic, nursing and documentation functions [103].

The healthcare area is certainly a complex environment for the introduction of ubiquitous computing, for multiple reasons. Medical data are the most privacy-sensitive and thus require appropriate data protection measures, such as tiered access procedures, the prevention of new transient data access and undesirable secondary use [107]. The government and insurance companies may be reluctant to finance pervasive healthcare under the existing reimbursement regulations, since this can trigger distribution battles among the various actors, for instance, about the issue of whether the home environment should be promoted as a healthcare and nursing location [108]. For these reasons alone, applications in the healthcare domain have rather a long-term perspective and must be implemented step-by-step. Finally, a number of ethical issues arise, which can be summarized under the captions security, autonomy and participation [109].

3.1.3 Context-awareness

The concept of *context-awareness* or *context-aware computing* plays a central role in the ubiquitous computing paradigm. Schilit and Theimer [110] used the term *context-aware computing* to refer to software that is able to adapt to its location of use, physically proximate people and objects, in addition to changes to those objects occurring over time. Often, context-aware computing enable computer services that can take automatic actions on behalf of the user, such as presenting information relevant to the user's location.

Since the introduction of the term *context-aware computing* during the early 1990s, the notion of *context* and what it entails have been intensely debated within the relevant lines of research. Various definitions of *context* have been suggested in literature [110-112]. Typically, these definitions specify various types of situational information, such as location, identity, time and activity that characterize a specific usage situation [113]. In healthcare, all these contexts can be acquired from various kinds of sensors, which are distributed in a healthcare environment. This requires a new data management model to represent context in a sharable and reusable manner, and also requires systems to deal with a huge amount of data in real-time.

Dey [114] defined *context* as "any physical object that can be used to characterize the situation of an *entity*". An entity is a *person, place,* or *object* that is considered relevant to the interaction between a user and an application, including the user and the application themselves.

When dealing with *context*, three entities can be distinguished [111]: *places* (e.g., rooms, buildings), *people* (e.g., individuals, groups) and *things* (e.g., physical objects, computer components). Each of these *entities* may be described by various attributes which can be classified into four categories: *identity*, which implies that each entity can be uniquely identified, *location*, which corresponds to an entity's position, co-location, proximity, etc., *status* or *activity*, which refers to the intrinsic properties of an entity, e.g., temperature and lightning for a room, processes running currently on a device, etc. and *time*, which is used by timing mechanisms to accurately identify situations, ordering of events, etc.

In context-aware systems in healthcare environments, applications should have knowledge of their surrounding physical and computing environment. This environment is composed of people, mobile and fixed computing devices, and things such as doors, walls, desks and chairs, amongst others.

3.1.4 Location-based Services

Knowledge of the location of users and devices is a pre-requisite for the support of context-aware healthcare applications. Locating physical objects is a separate research area, which involves the development of location hardware devices, software storage structures and mechanisms to determine the location of an object. Location has been already widely investigated for its use in context-aware systems [115-119].

Location information used in isolation has limited applicability. This information can better be used together with information about the domain in which it is gathered, to enable its visualization for presentation in a human readable form. For instance, information about a patient's echocardiogram can be displayed in a monitor for cardiac surgeon located in a meeting room. Of particular relevance to context-aware applications is the knowledge about the physical objects in the immediate vicinity of a target object, such as a *person*. Many applications are not interested in the absolute location of an object, but only its relative location to other objects nearby, such as 'if a person is standing in front of the workstation' [120].

Ubiquitous computing solutions often combine interactive media with sensor technology. Regarding these technologies, this thesis has mainly concentrated on location-aware sensor technologies. Location-aware sensor technologies can detect users' presence or physical position in space. This information can be used to trigger digital services or present information related to the user's current location. Examples of presence and positioning technologies that enable location-based interaction include GPS, WLAN positioning, and *Bluetooth Access Point* (BAP) [121]. The later has been used in this work and is discussed in the sequel.

Bluetooth

We applied Bluetooth technology in our research for positioning due to the high availability of Bluetooth interfaces in current mobile devices, which makes this technology one of the most used in healthcare environments [122]. Projects like *MyHeart* [123], *MobiHealth* [124] and *SMART* [125] acknowledge its usability in real system implementations mainly working on top of the communication protocols provided by each vendor.

The Bluetooth wireless technology provides a low-cost, low-power, short-range radio link for mobile devices and Local Area Network (LAN) access points. It offers fast and reliable digital transmissions of both voice and data, and operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band, which is globally available and license-free [126]. Bluetooth technology comprises hardware, software and interoperability requirements. It has been adopted in diverse areas, such as the automotive industry and the healthcare sector. Bluetooth is ideal for the office environment, and the ability of this technology to create *ad hoc* networks reduces connection delays, while permitting connectivity at a low-cost [127].

Bluetooth belongs to a category of *Short-Range Wireless* (SRW) technologies originally intended to replace the cables connecting portable and fixed electronic devices. It is typically used in mobile phones, cordless handsets and hands-free headsets. The specifications describe operation in three different power classes: for distances of 100 meters (long range), 10 meters (ordinary range) and 10 cm (short range). Bluetooth has introduced the notion of a *Personal Area Network* (PAN), a close range wireless network. The *Service Discovery Protocol* (SDP) allows Bluetooth devices to discover which services are available or to find a Bluetooth device that supports a specific service [128]. A key feature of the Bluetooth specification is that it allows devices from different manufacturers to cooperate. Bluetooth does not just define a radio system, but it also defines a software stack (*Bluetooth Protocol Stack*) to enable applications to find other Bluetooth devices in the area, discover what services they offer and use the services they require [129, 130].

Even though Bluetooth technology has not been originally designed to support object localization, it offers this capability with advantages when compared with technologies like GPS since it is pervasively available in most mobile devices, is relatively cheap and the power consumption of Bluetooth components is relatively low. In contrast, because it has not been designed for object localization it does not offer the same accuracy as the technologies that have been specifically designed for this purpose [131]. For example, Bluetooth is not suitable for real-time target tracking. Nonetheless, the advantages mentioned earlier make Bluetooth a suitable technology for indoor co-localization, mainly because it can be realized, for example, with a simple bracelet that a person can wear [132, 133].

Healthcare systems could benefit from the knowledge about location of healthcare professionals, patients and devices by using the wireless infrastructure. Knowledge about location in healthcare environments can be helpful for finding people and to exchange information [134]. In this research, we applied BAP for estimating the location of devices in a healthcare environment through the device discovery phases performed by the Bluetooth master units [135].

3.2 Health-related Standards

The development and adoption of healthcare standards for e-health standardization is essential for sharing clinical information between healthcare professionals in a multi-disciplinary environment, proper message exchange between healthcare organizations, and supporting information exchange between software from different vendors not being dependent of a single software manufacturer.

Many different standards development organizations are working on e-health standardization, such as HL7, *open*EHR, CEN/ISO EN13606, MLHIM, ISO/IEEE 11073, LOINC and SNOMED. HL7 and ISO are considered healthcare standards. *open*EHR and MLHIM provide open source specifications and reference implementations of future proof EHR systems. LOINC and SNOMED-CT are two of a multiplicity of controlled vocabularies. Below we discuss the most prominent standardization efforts in e-health.

3.2.1 Health Level 7

Health Level 7[12] is a non-profit organization that develops a group of standards for communication of clinical information. These standards include message protocols (HL7 v2.x, v3), conceptual standards (e.g. HL7 RIM), document standards (e.g., HL7 CDA), and application standards (e.g., HL7 Clinical Context Object Workgroup CCOW).

Message protocols in HL7 have been designed to be triggered by events. A trigger event is an event in some clinical process, such as, e.g., a patient admission. A trigger event generates a request message that is sent to another system [136]. The HL7 v3 model contains concepts and data entities that are communicated in message exchanges, and shows semantic connections between those entities. It is used along with medical data standards such as LOINC or SNOMED CT to encode data in messages in an unambiguous way.

3.2.2 *open*EHR Dual Model

The *open*EHR specifications are maintained by the *open*EHR Foundation[13], a not for profit foundation that supports the open research, development, and implementation of EHR based on *open*EHR. The *open*EHR specifications are based on a combination of 15 years of European and Australian research and development into EHR and new paradigms, including what has become known as the archetype methodology for specification of content. With the *open*EHR approach, clinical information can be structured, stored, managed and exchanged in a safe and reliable way between different healthcare providers [137].

Even though HL7 is a well-known standard for the message exchange between heterogeneous HISs, it has some limitations for representing clinical knowledge, such as its combined use of structured components and coded terms, which can result in inconsistent interpretations of clinical information [138]. *open*EHR defined an architecture based on two-level modeling that separates information from knowledge, in this way addressing some limitations of HL7. *open*EHR dual model is relevant for our work and it has been applied for the message exchange between HIS, and it is discussed in more detail in Section 3.3.

[12] http://www.hl7.org/

[13] http://www.openehr.org/

3.2.3 Other Types of Interoperability in Healthcare

Many different types of technologies aim to achieve interoperability in healthcare. These key healthcare integration technologies include CEN/ISO EN13606, MLHIM, ISO/IEEE 11073, LOINC and SNOMED.

CEN/ISO EN13606

CEN/ISO EN13606[14] is a European norm from the European Committee for Standardization (CEN), and has also been approved as an international ISO standard. EN13606 is a communication standard for medical information in electronic health records and focuses on interfaces for data exchange and structured data packaging for communication. Information exchange can take place between entities such as clinical applications, central data repositories and software components. Health records can be transmitted as a whole or in fragments.

MLHIM

Multilevel Healthcare Information Modeling (MLHIM) specifications are a fully open set of specifications for the development of health information systems based on multilevel modeling. The MLHIM specifications are based on the concepts of the dual model approach inspired by *open*EHR specifications and the healthcare-specific data types as defined by the ISO 21090 standard [139, 140]. MLHIM is an approach to programming computer systems that need to transmit data with unambiguous, shared meaning between different implementations. MLHIM advocates of an abstract framework embodying the syntactic structure and semantic context of sharable data, in sharable models based on the ubiquitous XML language technology suite [141].

ISO/IEEE 11073

The ISO 11073 [142] family of standards describes protocols and data formats for communication between electronic medical devices. It focuses on bedside devices that are used in acute care settings, and has been designed especially for supporting real-time interoperable plug-and-play of devices, simple implementation of protocol stacks and efficient message processing.

LOINC

LOINC[15] stands for *Logical Observation Identifiers Names and Codes* and is a naming and coding system for clinical observations. LOINC is especially well suited to express laboratory results. It does not explicitly cover actual

[14] http://www.en13606.org/

[15] http://loinc.org/

diagnoses, which are usually described and encoded by the ICD coding system (*International Classification of Diseases*).

SNOMED-CT

SNOMED-CT[16] stands for *Systematized Nomenclature of Medicine - Clinical Terms* and is a standard terminology of medical concepts, and aims at achieving semantic interoperability. Each concept is assigned a unique numeric code consisting of six to eighteen digits, such as, e.g., 22298006 for *myocardial infarction*.

3.3 *open*EHR

Dual modeling has been shown to be a viable solution for the semantic interoperability of software systems, without imposing any specific programming languages or persistence models [141]. In dual modeling, there is a clear separation of information and knowledge levels. The information model is based on stable and generic concepts, whereas the knowledge model expresses the variability and dynamics of the problem field. The dual model approach allows domain experts to introduce clinical knowledge without changing the basic functions of the system, since this knowledge remains rigid within the information model [143]. The underlying idea of dual modeling is that the changes in the structure and business rules of a healthcare application are reflected on the knowledge model and not on the information model [144].

3.3.1 Dual Model

The *open*EHR dual model prescribes the use of archetypes for describing clinical knowledge. *open*EHR defined the Archetype Definition Language (ADL) 1.4 release [145] and the Archetype Query Language (AQL) [146] to enable the definition of archetypes and queries on them. An archetype is a clinical model employed by a domain expert to represent a specific concept of this domain, allowing health professionals to become directly involved in the development of EHRs. The Clinical Knowledge Manager (CKM)[17] is a collaborative system for archetype development, management and publishing.

The *open*EHR approach has been developed based on a two-level model, as shown in *Figure 3-8*. On the first level, a *Reference Model* (RM) is defined for modeling an EHR structure using a predefined set of classes. On the second level, an *Archetype Model* (AM) defines specific concepts by

[16] http://www.ihtsdo.org/snomed-ct/

[17] http://www.openehr.org/knowledge/

restricting the RM classes in terms of archetypes, which are expressed in ADL and can be translated to any object-oriented implementation language. Data input from users are stored according to the RM, and must also comply with the concepts expressed by the archetypes. Since archetypes are designed by health experts and not by ICT professionals, this facilitates the interpretation of the knowledge extracted from the messages exchanged by HIS in different applications.

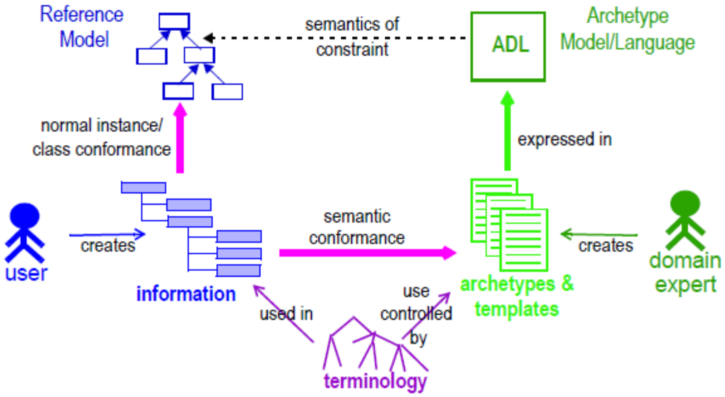


*Figure 3-8 open*EHR Dual Model (Source: Beale, 2007)

3.3.2 Reference Model

Figure 3-9 illustrates the main RM classes. The COMPOSITION class refers to one or more instances of the SECTION class, each containing ENTRY objects. The ENTRY class represents the actual recording of clinical content during a patient Observation, Examination, Assessment, or Intervention. ENTRY is defined as an abstract type that has four concrete subclasses: (1) OBSERVATION, which can be used to represent clinical observations, such as blood pressure; (2) EVALUATION, which can be used to represent assessments made after a clinical observation is completed, such as risk assessment; and (3) INSTRUCTION represents medication, surgical procedures, and other clinical interventions, and (4) ACTION, describes what was really done as a consequence of an INSTRUCTION.

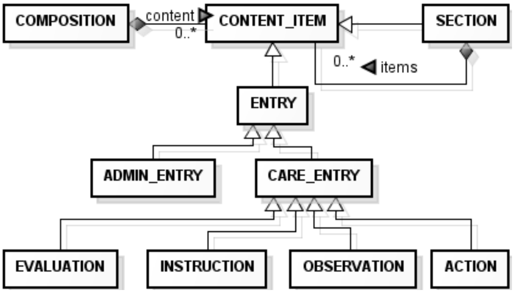


*Figure 3-9 open*EHR RM classes (Source: Beale, 2007)

3.3.3 Archetype Model

A clinical archetype has to be written for a particular information model. Since an ADL 1.4 archetype is an instance of the abstract *open*EHR AM, it is not represented in terms of instances of the RM classes. *Figure 3-10* depicts an overview of the AM classes, where the main *ARCHETYPE* class is composed by the *ARCHETYPE_ONTOLOGY* and *C_COMPLEX_OBJECT* classes. Hence, when processing an ADL archetype, a collection of AM objects is obtained. ADL uses three syntaxes: *cADL* (constraint form of ADL), *dADL* (data definition form of ADL), and a version of *first-order predicate logic* (FOPL), to describe constraints on data which are instances of some information model. The *cADL* syntax is used to express the archetype *definition*, while the *dADL* syntax is used to express data which appears in the *language*, *description*, *ontology*, and *revision_history* sections of an ADL archetype. A full implementation of the AM is required in order to construct, in memory, representations of the archetype (in ADL) in order to perform validation of the *dADL*.

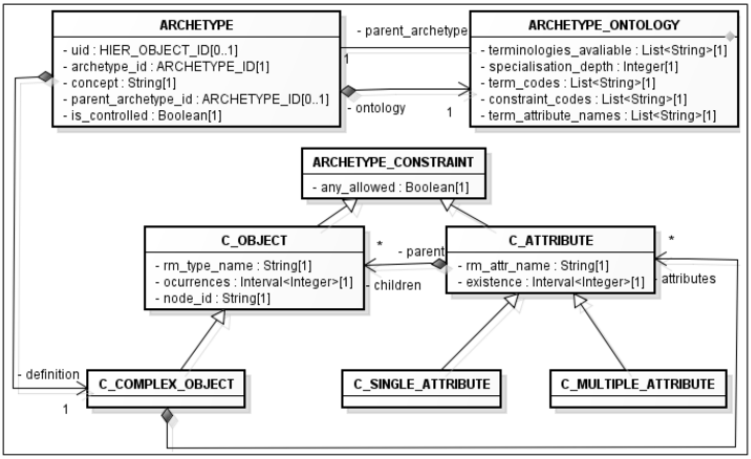

ARCHETYPE
- uid : HIER_OBJECT_ID[0..1]
- archetype_id : ARCHETYPE_ID[1]
- concept : String[1]
- parent_archetype_id : ARCHETYPE_ID[0..1]
- is_controlled : Boolean[1]
- parent_archetype
1
- ontology
1
ARCHETYPE_ONTOLOGY
- terminologies_avaliable : List<String>[1]
- specialisation_depth : Integer[1]
- term_codes : List<String>[1]
- constraint_codes : List<String>[1]
- term_attribute_names : List<String>[1]
ARCHETYPE_CONSTRAINT
- any_allowed : Boolean[1]
C_OBJECT
- rm_type_name : String[1]
- ocurrences : Interval<Integer>[1]
- node_id : String[1]
*
- parent
- children
C_ATTRIBUTE
- rm_attr_name : String[1]
- existence : Interval<Integer>[1]
*
attributes
- definition
1
C_COMPLEX_OBJECT
C_SINGLE_ATTRIBUTE
C_MULTIPLE_ATTRIBUTE

*Figure 3-10 open*EHR Archetype Model (Source: Beale, 2007)

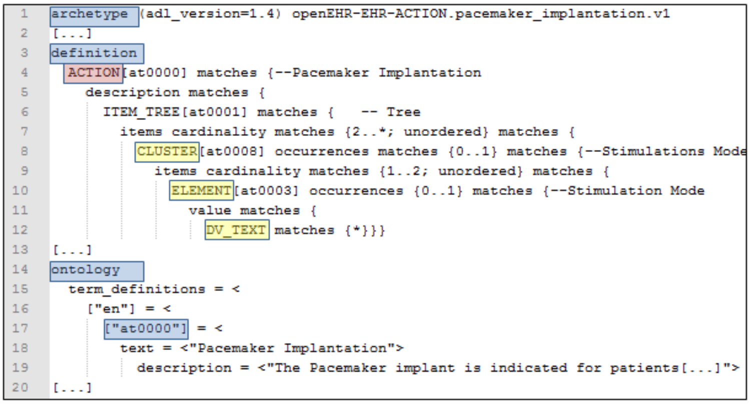
The *ARCHETYPE_ONTOLOGY* and *C_COMPLEX_OBJECT* classes contain information related to clinical concepts. The *ARCHETYPE_ONTOLOGY* contains attributes such as *terminologies_available*, *term_codes*, *constraint_codes*, and *term_attribute_names*, which are defined as strings. The *C_COMPLEX_OBJECT* is a specialization of the *C_OBJECT* abstract class, in which the attribute *rm_type_name* is also defined as a string. The *C_COMPLEX_OBJECT* has attributes of the *C_ATTRIBUTE* class, which in its turn contains a value *rm_attr_type* also defined as a string.

CKM contains a set of ADL files that represent the *open*EHR archetypes, which express clinical concepts to be reused in several kinds of health applications. In our experiments, we reused CKM archetypes, such as *Person* and *Clinical Synopsis*, but we also developed new archetypes to represent cardiology concepts, such as *Pacemaker Implantation* and *Coronary Cardiac Surgery*. CKM is able to retrieve one archetype if the word '*pacemaker*' is used for search[18]. The *Medication Order* archetype will be the search result, but the word '*pacemaker*' in this archetype refers to an exclusion criteria. Our clinical team wishes to submit these archetypes to the *open*EHR CKM community in order to contribute with the international archetype development process related to the cardiology domain archetypes. The archetype development process will be better explained in Chapter 5.

An *open*EHR archetype consists of three sections: a *header* contains an identifier for the archetype; a *definition* expresses the restrictions in a tree-like structure that classify the archetypes in terms of entries of the RM, and constrains the cardinality and the content of the information model instances that comply with the archetype; and an *ontology* contains the codes associated to nodes, the constraints on text and terms, and the bindings to terminologies, such as, e.g., SNOMED-CT.

Figure 3-11 shows an excerpt of the *Pacemaker Implantation* archetype defined according to *open*EHR. The *header* includes the name of the archetype (line 1). The *definition* (lines 3-13) contains the structure and restrictions associated to the clinical concept defined by the archetype. This part defines that *Pacemaker Implantation* specializes the *ACTION* class (line 4) of the RM, while the *CLUSTER* part (line 8) refers to the *stimulations mode* for pacemaker implantation and consists of one or two (1..2) *ELEMENT* with a value of type *DV_TEXT*. The *ontology* (line 14-20) includes the terminological definitions, and the linguistic expression '*Pacemaker Implantation*' is associated with the code '*at0000*'. The archetypes designed in this work are explained in more detail in Chapter 5.

[18] http://www.openehr.org/ckm/



```
archetype (adl_version=1.4) openEHR-EHR-ACTION.pacemaker_implantation.v1
[...]
definition
  ACTION[at0000] matches {--Pacemaker Implantation
    description matches {
      ITEM_TREE[at0001] matches {   -- Tree
        items cardinality matches {2..*; unordered} matches {
          CLUSTER[at0008] occurrences matches {0..1} matches {--Stimulations Mode
            items cardinality matches {1..2; unordered} matches {
              ELEMENT[at0003] occurrences {0..1} matches {--Stimulation Mode
                value matches {
                  DV_TEXT matches {*}}}
[...]
ontology
  term_definitions = <
    ["en"] = <
      ["at0000"] = <
        text = <"Pacemaker Implantation">
          description = <"The Pacemaker implant is indicated for patients[...]">
[...]
```

Figure 3-11 Excerpt of the pacemaker implantation archetype

3.4 Service-Oriented Architecture

Service-Oriented Architecture (SOA) defines how two computing entities interact in such a way as to enable one entity to perform a unit of work on behalf of another entity [6]. SOA is not a concrete architecture, but it is rather a style that leads to concrete architectures. That is, SOA is not a concrete tool or framework that can be purchased. SOA allows entities that need certain distributed capabilities to locate and make use of those capabilities. It facilitates interactions between service providers and service consumers, enabling the realization of business functionalities in a heterogeneous and distributed environment.

In an architecture that complies with the SOA principles, software resources are packaged as *services*, which are self-contained modules that provide well-defined functionalities and are independent of the state or context of other services. *Services* are described in a standard definition language, have a published interface, and communicate with each other by requesting execution of operations in order to collectively support a common business task or process [147]. Service interactions are defined using a description language. Each interaction is self-contained and loosely coupled, so that each interaction is independent of any other interaction.

3.4.1 Design Principles

The principles of SOA that allow SOA-based architectures to cope with the characteristics of heterogeneous systems are [148]:

(i) *Services* represent self-contained business functionality, such as, e.g., ‘create a patient’, ‘get a birth date’ and ‘calculate the heart disease risk’;

(ii) *Standardized service interface* in order to guarantee that the service contracts within a domain adhere to the same set of design standards, which results in the interoperability between heterogeneous systems;

(iii) *Loose coupling* to minimize the dependencies between different systems in order to support fault tolerance, scalability and flexibility in a distributed environment;

(iv) *Reusability of services* to create services that have the potential to be reused across the enterprise. These reusable services are designed in a manner so that their solution is independent of any particular business process or technology;

(v) *Discoverability of services* in which it should be possible to discover and interpret service descriptions in service repositories to be able to use these services; and

(vi) *Composability of services* provides the design of services that can be reused in multiple solutions so that they can be used in a service composition.

3.4.2 Interaction Patterns

Components in a SOA-based architecture may interact following different interaction patterns. Common interaction patterns include *request-response* and *publish-subscribe*. The *request-response* interaction pattern is realized by two related one-way message flows, one to request the execution of a service operation (request) and a corresponding (opposite) one to convey a reply to this request (response). *Figure 3-12* depicts a schematic view of the *request-response* interaction pattern. The *request* message, sent by a service consumer, results in the communication from that service consumer to a service provider. The *response* is sent by the service provider, resulting in *response* to the *request*.

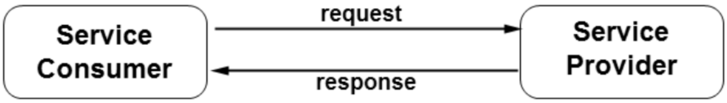


Figure 3-12 Request-response interaction pattern (Source: Erl, 2009)

The *publish-subscribe* interaction pattern provides a loosely coupled form of interaction that can be decomposed along the following three dimensions:

- *Space decoupling*: interacting parties do not need to know each others location. *Publishers* publish events through an event service and *subscribers* get these events indirectly through the event service;
- *Time decoupling*: interacting parties do not need to participate actively in the interaction at the same time. *Publishers* might publish some

events while *subscribers* are disconnected and the *subscribers* get these event when they connect to the event service;
- *Synchronization decoupling*: *subscribers* may asynchronously receive notification messages, and *publishers* are not blocked while producing information. The production and consumption of events do not happen in the main flow of control of the publishers and subscribers, and therefore do not happen in a synchronous manner.

The realization of the *publish-subscribe* interaction pattern typically requires the introduction of a mediator party, often called *broker,* such as, e.g., the event channel of the CORBA notification service. The mediator plays the role of a *service provider*, while both *publishers* and *subscribers* play the role of *service consumer*. Subscribers have the ability to express their interest in events and are subsequently notified of any event generated by a publisher that matches their registered interest. *Figure 3-13* illustrates the *publish-subscribe* mechanism.

Publishers issue one-way messages to subscribe and be able to publish event notifications with the mediator. Subscribers issue subscribe messages indicating their interest in receiving notifications about particular events. Variations of the *publish-subscribe* mechanism allow subscribers to specify their interest in event notifications based on, for example, topic, content or type. *Topic-based* subscriptions allow participants to publish event notifications and subscribe to particular topics or subjects, which are identified by keywords. *Content-based* subscriptions allow the participants to publish and subscribe to event notifications based on the actual contents of the event notifications. Finally, the *type-based* subscriptions allow participants to publish and subscribe to event notifications according to event notification types (structures). When a subscriber's interest matches a publisher's event notification, the mediator issues a message with a notification to the interested subscriber.

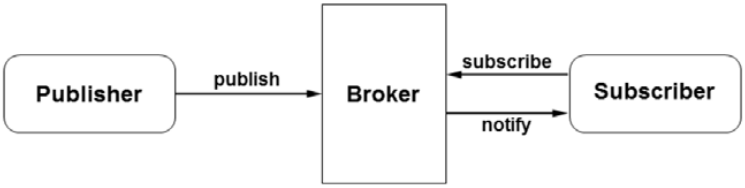


Figure 3-13 Publish-subscribe interaction pattern (Source: Erl, 2009)

3.4.3 SOA Technologies

Service-Oriented Architecture is commonly implemented by using *Web Services* technologies, because these technologies have been designed to support interoperable *provider-to-consumer* interaction [149]. *Web services* technologies are based on standards, such as SOAP-based web services, which are based on *Web Services Description Language* (WSDL) and SOAP, and REST*ful* web services, which rely on HTTP and XML/JSON, for instance.

A Web service is a service available via the network, typically the Internet, which performs tasks, solves problems and conducts transactions on behalf of a user or application [150]. Web services are used for composing applications by discovering and invoking available services rather than building new applications to accomplish some task [151].

REST defines a set of architectural principles by which one can design Web services that focus on a system's resources, including how representations of resources are addressed and transferred over HTTP by a wide range of clients written in different programming languages [152]. In the REST architectural style, data and functionality are considered *resources*, which are referred to by unique identifiers that clients use to interact with them, called *Uniform Resource Identifiers* (URIs). REST defines *resources* and *representations*. Resources and clients exchange resource representations to perform interactions. REST endpoints exchange *representations* of the resources they identify with the URIs [153].

The REST architectural style applies the four basics functions of persistent storage to a set of resources: create, retrieve, update and delete (CRUD). REST prescribes a client/server architecture and is designed to use a stateless communication protocol, typically HTTP, which has a vocabulary of operations called *request methods* that match the CRUD operations. HTTP 1.1[19] defines eight methods: HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS and CONNECT. REST*ful* Web services use HTTP requests to create, read, update and delete data. HTTP methods can be mapped to CRUD operations, such as *retrieving the representation of the resource identified on the URI* (GET), *updating the representation of the resource identified on the URI* (PUT), *creating a new resource* (POST), and *deleting a resource* (DELETE). In the REST architecture style, clients and servers exchange representations of *resources* by using a standardized interface and protocol [154]. The *representation* of a resource should be thought as the *state* of a resource. If the *representation* changes, an UPDATE operation on that resource should be performed. The current *representation* is typically returned in response to a GET [155, 156].

REST promotes the notion of self-describing messages, in which representation formats are based on agreed-upon standards and are specified within the messages themselves. Different messages can also specify different formats in the HTTP *content-type* and *accept* headers,

[19] http://www.w3.org/Protocols/rfc2616/rfc2616.html

indicating message data-payload format and data formats the caller is prepared to receive in response, respectively [157].

REST*ful* Web services satisfy a set of architectural constraints that, when applied as a whole, emphasize scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. In this work, we adopt this architectural style in the design of a message generator that enforces interoperability of healthcare applications.

3.5 Software Agents

An agent is a computational entity with autonomous behavior, which allows it to decide and perform its own actions. A software agent is an entity that functions continuously and autonomously in a particular environment often inhabited by other agents, and is capable of flexibly and intelligently intervene in an environment, without requiring human guidance or intervention [7]. Autonomy, the central concept in the notion of agents, means that an agent does not need to wait to receive instructions about what to do in each moment of its life. Instead, the programmer embeds beliefs, desires, and intentions in the agent, and the agents guide their actions by their behaviors [158].

Every agent is embedded in an environment, with which the agent interacts through sensors and actuators [159]. The particularities of the environment in which the agent lives affect directly the success or failure of the agents. The decisions taken by the agent are derived from the perceptions arising from the environment, and these decisions are translated into actions performed in the same environment. Agents can be classified according to their intelligence level. Agents with higher cognition abilities are classified as *cognitive* agents, whereas less cognitive agents are classified as *reactive* agents. Furthermore, if necessary, an agent is able to move in its environment, for example, by traveling between various hosts in a computer network [160].

Whenever the resolution of a goal requires the collaborative effort of two or more agents, we talk about a Multi-Agent Systems (MAS). Although agents can inhabit a common environment, we must ensure the autonomy of each agent, which implies that each agent must have its own specific purposes and characteristics. The main objective of a Multi-Agent System can be achieved through the cooperation and coordination abilities of its agents, combined with well-defined communication rules [7].

3.5.1 Agent Communication Language

Agents in a MAS require the use of a language that is understandable to them in order to properly exchange information. The *Agent Communication Language* (ACL) proposed by the *Foundation for Intelligent Physical Agents* (FIPA)[20] is often used for this purpose, since it is a standard language for communication between agents. FIPA-ACL is based on the theory of speech acts and it is beneficial due to the formal definition of its performative library. FIPA-ACL has a set of performatives[21] that define the message structure, such as *accept-proposal, agree, inform* and *request*. For instance, the performative *accept-proposal* is a general-proposal acceptance of a proposal that was previously submitted, in which the agent sending the acceptance informs the receiver that it intends that the receiving agent performs the action. A FIPA-ACL message is encapsulated as one of the attributes of the message exchanged by the agent (*content* attribute).

The FIPA specification defines a reference model for the basic operations within a MAS, and some entities to perform these management tasks, which are *Agent Management System* (AMS), *Directory Facilitator* (DF), and *Message Transport Service* (MTS)[22]. AMS is responsible for the management operations, such as creation and deletion of agents. Each agent must register with the AMS before it begins its execution, and unregister in order to finish its execution within the platform. It is also the responsibility of AMS to maintain the directories of all agents present in the platform and their current state, such as *active, suspended* and *waiting*. DF was defined in the agent management specification as an optional component of the platform that is responsible to provide yellow pages services to other agents. It maintains the complete list of the agents that have public services in order to help agents find services in the platform. MTS is a service responsible for the transport and delivery of ACL messages between agents on the platform [161].

FIPA standardized the language FIPA-SL (*Semantic Language*)[23], which is based on first-order logic and aims to enable agents to express properties and relations between objects from a specific domain. FIPA-ACL provides a formal definition of the communication language and its semantics. The *Semantic Language* (SL) is the formal language used to express the semantic content of an FIPA-ACL message [162].

FIPA *Contract Net Interaction Protocol* (IP)[24] is a protocol for negotiation between the agents, in that one agent (*Initiator*) sends out a number of *calls*

[20] http://www.fipa.org/

[21] http://www.fipa.org/specs/fipa00037/SC00037J.html

[22] http://www.fipa.org/specs/fipa00023/SC00023J.html

[23] http://www.fipa.org/specs/fipa00008/SC00008I.html

[24] http://www.fipa.org/specs/fipa00029/SC00029H.html

for proposal (*cfp*) to a set of participants. Some of these participants *refuse* the call, while others may come up with a *proposal*. The *initiator* then considers the proposals and *reject*s and *accepts* some of them. For those that are *accepted*, three different results can be sent back to the *initiator*: *failure, inform-done* and *inform-result* [163].

3.5.2 Agent Implementations

Often it is necessary to incorporate intentionality in cognitive agents. The BDI architecture is based on intentionality, and considers *Beliefs*, *Desires* and *Intentions* as mental attitudes that generate human action [158]. Rao and Georgeff [164] adapted the model proposed by Bratman [165], transforming it into a formal theory and an execution model for software agents based on the notion of beliefs, goals and plans. The basic assumption of the BDI model is that actions are derived from a process called *practical reasoning*, which consists of two steps, namely *deliberation* and *reasoning,* to reach a state.

The following example illustrates this process: after an intense semester of classes, a student has many options for leisure during his vacation. Among the alternatives, the student can travel, read, attend festivals, and so on. In the *deliberation* step, one of these alternatives is chosen. Suppose the student decides to travel to some city, in the *reasoning* step the student reaches a state in the process, which results in a plan, enabling the student to arrive at his chosen destination. In this case, a plan consists of purchasing tickets, boarding, and so on.

Various implementations of the BDI architecture are available. JACK [166], JAM [167], JASON [168] and JADEX [169, 170] are frameworks that enable the development of agents in a certain language and provide an execution platform for agents. JADEX allows the creation of goal-oriented agents and provides a set of development tools to simplify the creation and testing of agents. It supports the two-step practical reasoning process: *goal deliberation* and *means-end* process. This means that JADEX allows the construction of agents with explicit representation of mental attitudes (*believes, goals* and *plans*) and that automatically deliberate about their goals and subsequently pursue them by applying appropriate plans [170]. JADEX is based on a BDI metamodel defined in a XML Schema, which is used to validate the agent models specified in XML Agent Definition Files (ADF).

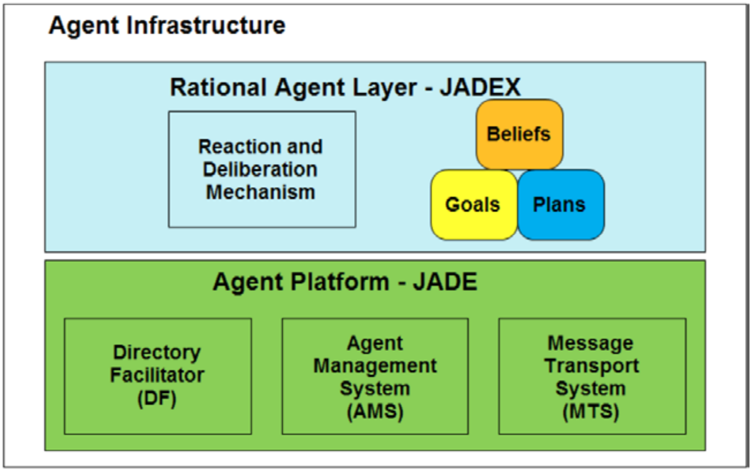
JADE[25] platform focuses on implementing the FIPA reference model providing the required communication infrastructure and platform services. JADE offers a task-based model in which the designer can perform any kind of agent behavior. JADE platform has been slightly extended to support the launching of JADEX agents. The JADEX

[25] http://jade.cselt.it

interpreter itself is realized as a special type of JADE agent, which loads an agent model supplied at startup, and creates its own instance of the reasoning engine according to the settings given in the model, such as initial beliefs, goals and plans. The functions performed by the execution model components (message receiver, dispatcher, scheduler) are implemented as cyclic behaviors that continuously run inside the agent. These behaviors call the reasoning engine to process incoming messages and perform internal reasoning [171]. In each JADE agent cycle, the reasoning engine is called to process one event and execute one plan step. Using a reference to a JADE agent object, JADEX plans have direct access to all operations of the JADE API as well, such as for handling of FIPA ACL messages. JADEX agents can potentially run on any middleware platform that provides some basic services concerning agent management and messaging. Adapters for JADEX have been designed for the agent platform JADE and for a JADEX Standalone platform, as shown in *Figure 3-14*.

In this work, we applied the BDI architecture for representing a structure capable of producing adaptive behaviors, by providing a set of reasoning rules. JADEX was chosen in our architecture because it is an environment that allows the formal description of cognitive agents based on the BDI architecture, allowing the programming of intelligent software agents in XML and Java.

Figure 3-14 Platform Integration


Agent Infrastructure
Rational Agent Layer - JADEX
Reaction and Deliberation Mechanism
Beliefs
Goals
Plans
Agent Platform - JADE
Directory Facilitator (DF)
Agent Management System (AMS)
Message Transport System (MTS)

MAS offer a suitable programming paradigm for simulating and modeling healthcare information systems, where resources, data, control and services are widely distributed. Healthcare information systems generally consist of discrete and heterogeneous entities, with a significant demand for remote access by healthcare providers and healthcare professionals. Most agent-based e-health applications concern to the use of agent technology to monitor patients, supervise treatments and perform data mining [172-176].

Chapter 4

Architecture Overview

In this chapter, we introduce our architecture and its main components. To design our reusable architecture we first identified the functional requirements that a pervasive healthcare model should fulfill. Further, we identified non-functional requirements from the stakeholders involved in the pervasive healthcare environment. The requirements were identified by interviewing the stakeholders and performing a literature review the technologies currently applied in the pervasive healthcare. We employed healthcare standards, ubiquitous computing technologies, service-oriented architecture and software agents in the design of a reusable architecture, meant to be generally applicable to different legacy healthcare information systems, from which applications can be built to be used in realistic pervasive healthcare scenarios. In the sequel, we designed a methodology to support the development of the interoperable applications to exchange messages in pervasive healthcare environments.

This chapter is organized as follows: Section 4.1 classifies the identified functional and non-functional requirements of the reusable architecture. Section 4.2 presents the use cases identified for each scenario. Section 4.3 gives an overview and presents the main components of our architecture and their relationships. Section 4.4 explains the methodology we adopted to design our reusable architecture.

4.1 Requirements

Requirements are often divided into functional and non-functional requirements. Functional requirements state *what the system should do*, and the non-functional requirements are *attributes of or constraints on a system* [177]. To design our architecture, we first identified the functional requirements that a pervasive healthcare model should provide in the cardiology domain. Pervasive healthcare applications have been designed to fulfill the functional requirements from the healthcare professionals and patients who participated in our experiments. Further, we identified non-functional requirements from the stakeholders involved in message exchange in pervasive healthcare environments. The non-functional requirements affected the design of the applications and thus, how the applications designed satisfied the identified functional requirements. The requirements were identified by interviewing the stakeholders (mainly functional requirements) and performing a literature review of existing technologies applied in pervasive healthcare environments (mainly non-functional requirements). Several non-functional requirements have been mentioned in [177], however, with respect to the scope of our research and the identified functional requirements, we limit our attention to a subset of these non-functional requirements.

The functional requirements were identified by interviewing the stakeholders and with respect to our scenarios in the cardiology domain. The non-functional requirements were identified through interviews and a literature study on legacy healthcare information systems. Use cases were defined to document the identified functional requirements, as described in Section 4.2. The non-functional requirements were documented in a textual format as a bullet list. We concentrated on the main functionality of the applications in the different scenarios, so that requirements related, for example, to user interface usability were considered out of scope of our research.

4.1.1 Stakeholders Interviews

We have identified and worked out three scenarios in the cardiology domain: *delivery of laboratory analysis results, schedule of an appointment for pacemaker evaluation,* and *support of a medical staff meeting to prepare for a cardiac surgery*. To identify the functional requirements, we have performed two interviews for each scenario with several healthcare professionals in a focus group-based research [178, 179]. All the healthcare professionals sat together with the software engineer, and answered and discussed our open-ended questions about our three scenarios.

Based on the results of the first interview for each scenario, the software engineer identified the common tasks and their corresponding application scenarios. Then we designed our architecture in terms of its components and their relationship. In our second interview with the healthcare professionals, we validated the functional and non-functional requirements of our designed architecture and updated the architecture accordingly [180].

During both interviews with the healthcare professionals, we also asked open-ended questions regarding the non-functional requirements, such as, for instance, the acceptable application response time. Then, we listed all these non-functional requirements as natural text and sent them back to the healthcare professionals by email to validate the non-functional requirements as well. The functional requirements were described in use cases that document the requirements of the pervasive healthcare applications.

4.1.2 Functional Requirements

We classified the functional requirements according to three applications scenarios identified by the healthcare professionals and software engineer.

Delivery of Laboratory Analysis Results

Laboratory test results are a critically important source of information for medical decision-making and an integral piece of information of the EHR. Missed laboratory results or delayed recognition results potentially leads to patient diagnostic errors, adverse events and liability claims. Most paper-based methods of communication between laboratories and ordering physicians are especially vulnerable to failures. The exchange of laboratory test results is one of the most critical aspects of the total testing process. This process has been centered for decades on the human component, with positive effects as well as potential adverse consequences. Manual delivery of results is a time-consuming activity and requires the constant presence of laboratory professionals in the laboratory with negative economic and organizational impacts. To overcome these inherent limitations, healthcare professionals need an automatic and expert system for the exchange of laboratory results. Expert systems that notify providers about abnormal test results in integrated EHR systems offer a potential solution to this problem. These systems can communicate through ‘alerts’ (computerized notifications of abnormal clinical information) transmitted directly to the healthcare professional, possibly accelerating the review of patient information.

Schedule of Appointments for Pacemaker Evaluation

A pacemaker is a medical device to regulate the heartbeat, and people carrying pacemakers require periodic clinical check-ups, including certain tests such as ECGs, which record the electrical activity of the heart. In addition, the status of the pacemaker should be regularly checked or ‘interrogated’ (often done remotely using a telephone system) to provide information regarding the type of heart rhythm, the functioning of the pacemaker leads, the frequency of utilization of the pacemaker, the battery life, and the presence of any abnormal heart rhythms. All contemporary devices are programmable with information and settings that can be changed and stored. Information is obtained by transmitting data from the pulse generator to a programmer, usually done during a follow-up office visit.

The pulse generators are usually powered by lithium batteries that work for an average of five to eight years before they need to be replaced. When the batteries start to wear out, they do so in a slow and predictable fashion, allowing sufficient time to detect it and plan a pulse-generator replacement. Replacing the pulse generator usually requires a simple procedure in which a skin incision is made over the old incision, the old generator is removed, and a new generator is implanted and joined with the existing leads.

In our scenario, the cardiologist physician provided on-going follow-up care for patients with pacemakers, and used a legacy system to keep track of the care provided to its patients. Pacemaker evaluation was often scheduled via the phone. The usual schedule for pacemaker check-ups was:

- *First visit*: 7 to 10 days after the implant surgery;
- *Second visit*: 3 to 4 months after implant surgery;
- *Follow up visits*: every 6 months until the pacemaker battery starts to show signs of depletion.

During the evaluation, the physician usually spent a significant amount of time searching for previous information related to patient's pacemaker implantation.

Support of Medical Staff Meetings to Prepare for a Cardiac Surgery

Cardiac surgery is one of the best examples of teamwork in healthcare, and is performed by highly trained staff: a *cardiovascular surgeon*, an *assistant surgeon*, a *cardiovascular anesthesiologist*, a *perfusionist*, and *cardiovascular nurses*.

Cardiac Surgeons usually order routine tests before admission to the hospital to identify potential problems that might complicate surgery. The most common tests performed before a cardiac surgery include chest X-rays, electrocardiogram, laboratory tests, cardiac stress tests, angiograms, and holter monitoring. Normally, these tests are performed in several medical clinics, generating a huge amount of healthcare information about a patient in a distributed healthcare environment.

Before going to a cardiac surgery of a certain patient, the team must have a meeting in order to prepare this surgery. In general, scheduling of cardiac surgeries involves a procedure that consists of the following steps: (i) check the availability of resources in the hospital, such as blood bank, Intensive Care Unit (ICU) and surgical center surgery room; (ii) set a date for a discussion meeting of the Heart Team about the patient's surgery, where medical reports containing information on the patient's EHR need to be obtained from the cardiology clinics in which the patient has been treated; and (iii) notify the patient about the time slot when the surgery is going to take place. Quite often, the actors involved in this procedure (and in other clinical activities) rely on informal offline communication (telephone or fax) to perform their tasks.

Due to the complexity of the involved resources, cardiac surgery requires the full integration of individual efforts with maximum efficiency to make sure that its action plan is performed successfully.

4.1.3 Non-Functional Requirements

Non-functional requirements should determine the way in which the stakeholders interact with the pervasive healthcare applications. The following non-functional requirements were identified:

- *Reliability* is the ability of a system to perform its required functions under specified conditions for a definite period of time. Reliability can be considered under two separate headings: availability (is the system available for service when requested by end-users?), and failure rate (how often does the system fail to deliver the service as expected by end-users?);
- *Performance* concerns the speed of operation of a system, and may be related to response time (how quickly the system reacts to a user input), throughput (how much the system can accomplish within a specified amount of time), and scalability (the capability of a system to increase total throughput under an increased load when new resources are added);
- *Security* is considered in a system to ensure that only authorized access to the system and its data takes place, and to ensure the integrity of the system from accidental or malicious actions. Examples are that the access permissions for system data may only be changed by the system's data administrator, all system data must be backed up every 24 hours and the backup copies should be stored in a secure location which is not co-locate with the system, and all external communications between the system's data server and clients must be encrypted; and
- *Usability* is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of the system or a component. Usability requirements include well-structured user manuals, informative error messages, help facilities, and intuitively appealing and consistent graphical user interfaces.

Non-functional requirements are difficult to express due to certain constraints related to the design solution that are unknown at the beginning of the development process. Certain constraints are highly subjective and can only be determined through complex empirical evaluations.

4.2 Use Cases

Below we discuss the functional requirements with respect to the three aforementioned scenarios applications. For each scenario application, first we show a use case to explain its main functionality and the stakeholders involved in the application. The stakeholders could be physicians, medical students, nurses, assistants and patients of the Hospital Santa Casa and Cardiology clinics in Marília (São Paulo, Brazil) in which the stakeholders use the legacy systems.

4.2.1 Delivery of Laboratory Analysis Results

A stakeholder (order placer) may enter a laboratory order into a patient's EHR. A laboratory requisition is generated and is communicated to the laboratory. The laboratory requisition information is entered manually or captured electronically into the laboratory information system (LIS). After the specimens have been collected and, if necessary, delivered to the laboratory, the laboratory processes the specimens. The laboratory performs the tests. If the tests are successful, results are obtained and released in the LIS. A person authorized by the laboratory reviews and approves the laboratory test results to be sent to the ordering stakeholder. The LIS transmits the results to the stakeholder's information system. The ambulatory system (EHR) incorporates the results into the patient's electronic record. The stakeholder logs into his information system and views the laboratory results in order to inform patient care decisions.

Figure 4-15 shows the use case for the exchange of clinical laboratory results. The patient visits his primary care physician (stakeholder) for a routine or emergency visit. The primary care physician determines that laboratory test should be performed, so the patient schedules an appointment with a local reference laboratory. Once the laboratory tests have been completed, the laboratory professional delivers the results to the primary care physician.

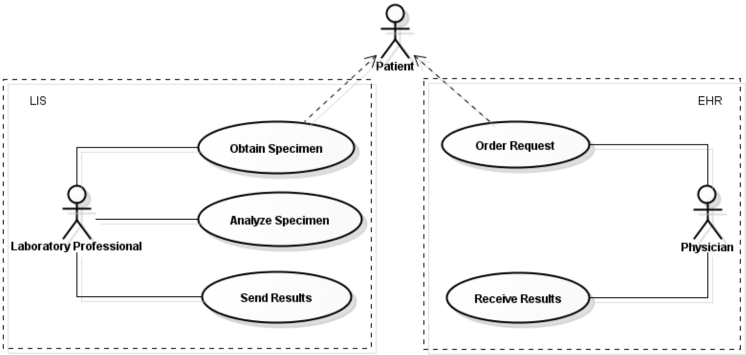


Figure 4-15 Use Case Diagram – Delivery Laboratory Results

These use cases require that interoperability requirements for healthcare information exchange are fulfilled to maximize efficiency in the healthcare environment. Established healthcare standards facilitate the exchange of information among stakeholders, by ensuring that data are structured in the same way. However, the structured data according to the defined fields in a message or file (syntax) is a necessary but not sufficient condition for interoperability among EHR systems.

For interoperability, the information to be exchanged has to comply with syntax but also semantic structure. Structured data is driven by the goal of improving the interoperability of data between disparate systems, and simplifying the process of harmonizing data from disparate sources.

Some assumptions are necessary to guarantee proper information exchange between the laboratory and the ambulatory EHR:

- Healthcare providers access clinical information through an EHR system;
- Appropriate transport protocols, a patient identification mechanism, requisition (order request) identification mechanism, and vocabulary standards have been agreed to by all relevant participants;
- These use cases only address the exchange of laboratory results that are associated with the Cardiology Healthcare Network;
- All relevant parties have agreed on a structured database schema of the laboratory test results;
- Established network infrastructure to enable consistent, appropriate, and accurate information exchange across healthcare provider systems, data repositories and locator services, such as methods to identify and authenticate users, and methods to identify and select healthcare providers;
- Laboratory Information Systems (LIS) is the sender of laboratory test results while Electronic Health Record Systems (EHR) is the receiver;
- The laboratory results need to be available to the patient; and

- The transport mechanism provides guaranteed delivery and error handling within the Cardiology Healthcare Network.

Any failure to achieve the above assumptions has a significant negative impact on the electronic communication of laboratory results, and the ability of healthcare professionals to meet the assumptions of meaningful use in all phases.

The primary focus of these use cases is on the request and delivery of laboratory results as standardized structured data, and expressed with simple data types (positive/negative, numeric values and units of measure) or using standard terminology (e.g. LOINC and SNOMED-CT), with a limited set of text results and defined structure.

4.2.2 Schedule of Appointments for Pacemaker Evaluation

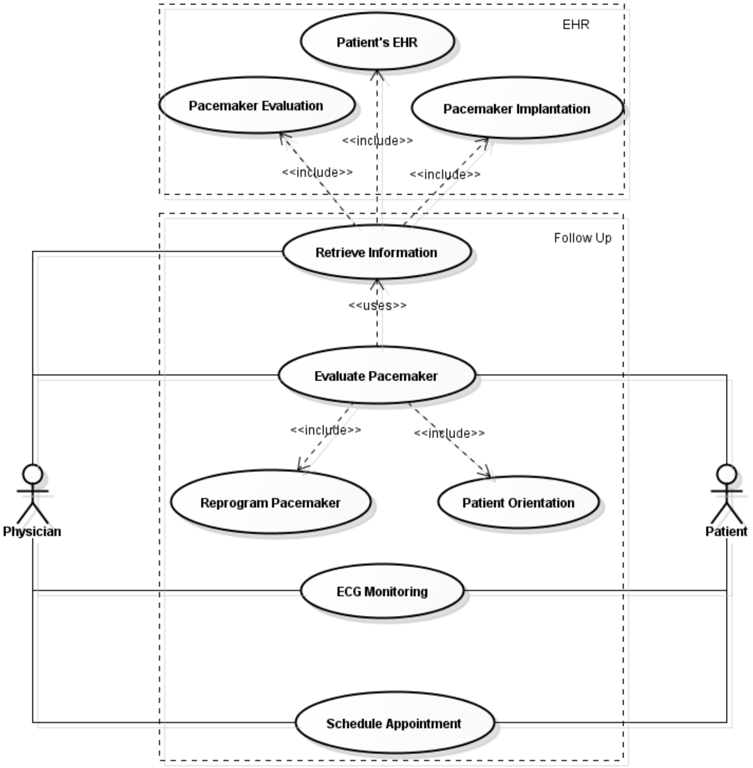
Pacemaker implantation is the initial phase in the lifelong management of a patient with a pacemaker. The challenge of this treatment lies in the comprehensive follow up of the device. As the number of implanted devices increases so does the burden of follow up. This is aggravated by the increasing amount of data provided by devices and the increasing sophistication of programming therapy and detection algorithms [181]. Like most medical interventions, pacemaker follow up has to be tailored to the individual.

Despite the increasing use of pacemaker therapy, assessment of pacemaker function and electrocardiogram (ECG) interpretation continue to challenge even experienced critical care healthcare professionals. Accurate assessment of pacemaker function is essential in the evaluation of patients, especially patients with symptoms that may be related to pacemaker malfunction, such as syncope or palpitations [182, 183]. In order to follow up patients who carry pacemakers, a dedicated area for pacemaker evaluation should be provided, which enables the patients to have their appointment performed in a private and safe environment. Relevant pacemaker programming devices and appropriate information for all evaluated pacemakers should be available. *Figure 4-16* shows the use cases diagram for scheduling an appointment for pacemaker evaluation. The checklist used by healthcare professionals to perform the pacemaker follow up are listed below:

- Evaluation of correct device function and record essential information about device, such as pacing lead impedance, sensed P/R wave amplitude, pacing threshold, and battery voltage and impedance;
- Optimization of device function and maximization of device longevity;
- Basic resources provision for pacemaker follow up, such as ECG monitor and facilities to admit patients in the event of an emergency;

- Provision of patient and family with appropriated support and counseling about device therapy to prevent consequences in patients;
- Reprogramming the pacemaker to change the stimulation mode to prevent arrhythmia;
- Access to smart system for automatic information collection about patient's EHR and previous follow up data, and appropriate storage of data;
- Scheduling for next visit or device replacement to perform the pacemaker evaluation.

Figure 4-16 Use Case Diagram – Scheduling an Appointment for Pacemaker Evaluation



The facility for trans-telephonic monitoring (TTM) of pacemakers has been in place for many years [184]. Its uptake has varied between healthcare information systems. As the sophistication of pacemaker devices increases the modes of follow up have become possible. The vast amount of data that can be retrieved from a pacemaker is likely increasing. Most pacemaker manufacturers have systems in place to allow remote pacemaker follow up. The real advantage of these systems is the ability for more regular device monitoring and the potential to evaluate the pacemaker during the appointment with a huge amount of clinical information.

4.2.3 Support of Medical Staff Meetings to Prepare for a Cardiac Surgery

The organization setting of a cardiac surgery is particularly complex because it involves specific procedures for cardiac function evaluation (i.e., catheterism cardiac and radiology), heart surgery, and intensive care unit monitoring. Cardiac surgical meetings form a complex environment that includes many highly trained people working together and using a large amount of information to treat people with severe heart diseases.

In order to be successful, a cardiac surgery requires highly skilled individuals to work effectively as a team. Surgeons, anesthesiologists, perfusionists, nurses and assistants must communicate effectively in order to ensure successful outcomes. Breakdowns in communication, decision-making or leadership could prove disastrous for the patient and, in consequence, for the reputation of the healthcare provider and medical staff [185].

Given the huge amount of heterogeneous sources of patient information, both administrative and clinical, integration is crucial to allow effective medical decision-making, effective care planning and proper resource control, and to provide healthcare professionals with the right information in the right place at the right time. Before performing an elective cardiac surgery of a certain patient, the team must have a meeting in order to prepare this surgery. *Figure 4-17* shows the use cases diagram with actors and use cases that represent the interactions for supporting a medical staff meeting to prepare for an elective cardiac surgery. The scheduling of elective cardiac surgeries involves a procedure that consists of the following steps:

- Check the availability of resources in the hospital departments, such as the blood bank, the Intensive Care Unit (ICU) and the surgical center surgery room;
- Set a date for a discussion meeting of the Heart Team about the surgery;
- Gather medical reports containing information about preoperative tests performed on the patients;
- Retrieve information on the patient's EHR from the cardiology clinics in which the patient was treated; and
- Notify the patient about the time slot when the surgery is going to be take place.

Figure 4-17 Use Case Diagram – Meeting for Cardiac Surgery

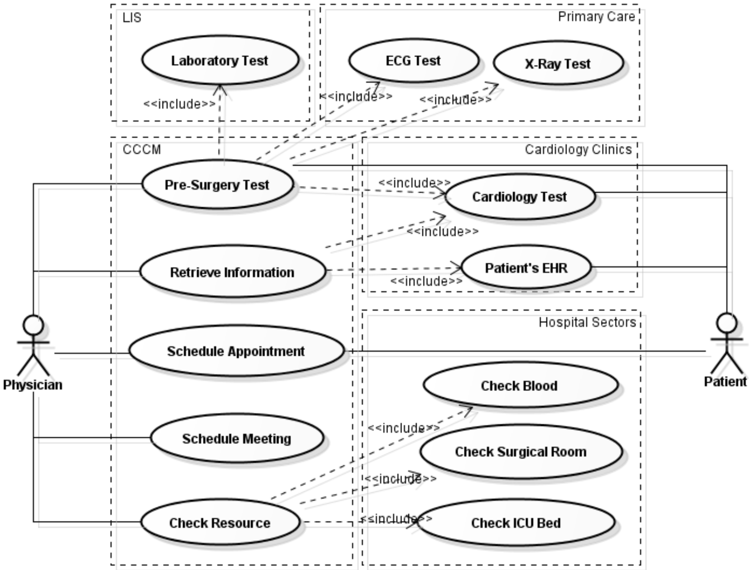


Figure 4-17 illustrates the building blocks of different sources of information, such as the laboratory information system, the primary care, the CCCM cardiology department, the cardiology clinics and the involved hospital departments. The information used by cardiologists for diagnosis and treatment of their patients varies from personal notes (history and physical examination), to signals (ECG), images (echocardiograms and angiograms) and reports from previous procedures. All this information should be provided in a digital and structured format, such as an electrocardiography report, a holter summary, an exercise test report, an echocardiography report, a stress test report, a radiology report and a cardiac catheterization report. In a cardiology healthcare environment, healthcare professionals require solutions that integrate information from different legacy systems in a standardized manner.

The medical staff meeting to prepare for a cardiac surgery requires an effective information exchange between stakeholders, and an intelligent healthcare environment, which provides access to smart systems for collecting automatic information about resources involved in a cardiac surgery, patient's EHR and preoperative tests. These use cases require patient's clinical information in electronic format to set up patient information from healthcare departments within Cardiology Healthcare Network repository. Each healthcare department as well as the cardiology clinics and laboratory analysis results were currently provided with computer-based systems for recording patient information.

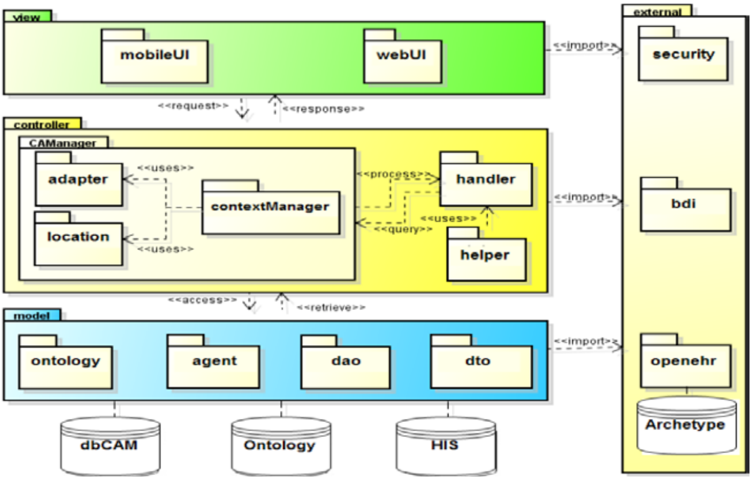
4.3 Architecture

The biggest challenge for our Reusable Architecture is to support the mobility and collaboration among healthcare professionals when they perform clinical tasks. The main problems to be solved are related to information overload and the heterogeneity of the mobile devices used by such professionals. We applied healthcare standards, ubiquitous computing technologies, service-oriented architecture and software agent to address the requirements and the problems mentioned on above section.

4.3.1 Overview

Figure 4-18 gives an overview of our Reusable Architecture, which was developed according to the MVC pattern (Model-View-Controller) [186]. This pattern is a proven solution to separate the business logic from the presentation logic, for the sake of flexibility and reuse. In our architecture, the *view* package contains the *mobileUI* package, which copes with interactions between the mobile end-user and the other components of the architecture, and the *webUI* package, which displays information to the end-users.

Figure 4-18
Architecture Overview



The *controller* package contains the *CAManager* package, which manages the exchange of context-aware messages, the *handler* package, which processes the inputs and outputs and acts as a message generator to a web service that provides services to allow the communication between heterogeneous systems, and the *helper* package, which adapts the data model to the view. The *model* package represents the domain models, and contains the *ontology* package, which represents the domain knowledge, the *agent* package, which issues requests and notifications within the architecture, and the *dto* and *dao* packages, which represent the *data transfer object* and *data access object* design patterns [187], respectively. The architecture has also an *external* package with some additional auxiliary packages. We designed our architecture combining design patterns that represent the state-of-the-art in web application design.

4.3.2 Components

One of the challenges of our architecture has been to cope with the different circumstances in which health professionals have to perform their tasks, including the heterogeneity of their devices. Context-aware systems are highly suitable for delivering healthcare services, especially for pervasive healthcare. This is because, in some healthcare services, the description of a situation by using '*what'* (activity), '*who'* (identity), '*where'* (location) and '*when'* (time) may not be enough, in which case more richness and higher reliability are required. In these cases we may have to include '*how'* (process), '*with whom*' (source), and '*so what*' (needed action). This increases the complexity of context-aware system and the probability of error in the delivery of healthcare services. Increased complexity could also lead to longer response delays, which may not be acceptable for some services, such as, e.g., emergency services.

Context-aware systems can be implemented in many ways. The architecture depends on the requirements and conditions such as the location of sensors (local or remote), the amount of possible users (one user or many), the available resources of the used devices (high-end-PCs or small mobile devices) or the extendability of the system. Furthermore, the method for context-data acquisition is important when designing context-aware systems because it restricts the architectural style of the system at least to some extent.

To cope with these different circumstances, we defined a *CAManager* package to processes the dynamic context information (e.g., end-user's location) and other context information (e.g., identity, user roles) that is obtained from various contextual sources. The *CAManager* includes the *contextManager* package, which interacts with the *adapter* package to handle content adaptation of the messages containing healthcare information, in order to address some of the specific characteristics of the end-users devices. If adaptation is necessary, *contextManager* asks the *view* package to adapt the graphical interface to the particular device.

In this research, we applied BAP technology for estimating the location of end-user devices through the coordination of the device discovery phases performed by the master units in pervasive healthcare environment.

The assumptions we adopted when designing the location system in this research were:

- Healthcare professionals and patients spend most of their time stationary in healthcare environments;
- Healthcare professionals were aware that they need to wear a device with Bluetooth capabilities in order to be localized;
- A homogeneous set of sensors were installed in all interest areas where it was possible to install sensors;
- The location system had to operate on a single floor and not being affected by Bluetooth devices on other floors;
- It was sufficient to perform relative location in offices and rooms within the healthcare environment in a short distance.

When the healthcare information systems interoperate with each other within Cardiology Healthcare Network, *CAManager* and *handler* exchange messages that possibly contain EHR extracts. Such messages are represented in accordance with the *open*EHR archetypes specification, in order to guarantee the interoperability with other legacy healthcare information systems. The *handler* package also supports both synchronous and asynchronous communication between architecture agents and HISs. It requests the appropriate content from a legacy HIS, and checks if this content has messages related to the end-user requestors. The archetypes designed in this research and the message exchange between heterogeneous healthcare information systems is discussed in Chapters 5 and 6.

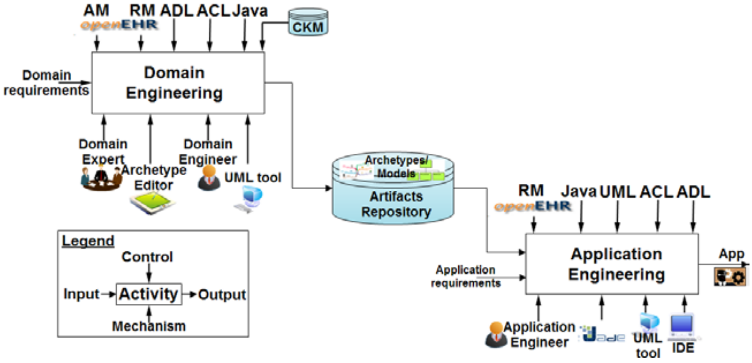
Our architecture includes static agents, which provide the necessary resources to the mobile agents, which in turn move through the architecture environment in order to achieve their goals and communicate with other agents. Agents interact with each other by using the Agent Communication Language (ACL) to share information and knowledge in the healthcare environment for which our architecture has been designed. The *agent* package contains static and mobile agents, each one endowed with specialized capabilities and goals for helping the healthcare professionals check the availability of resources, and obtain information and detect abnormalities about their patients. The static agents provide resources to the mobile agents, allowing them to move in the architectural environment in order to achieve their goals, and to communicate with other agents through an asynchronous communication channel. Agent modeling process is discussed in Chapter 7.

4.4 Methodology

Based on our reusable architecture, we defined a methodology to develop interoperable applications to exchange context-aware messages in pervasive healthcare environments. This methodology supports the development of the reusable artifacts, which can be integrated with legacy systems to overcome the weaknesses typically found in these systems, thereby adding value to legacy systems.

Figure 4-19 gives an overview of our methodology using the *Structured Analysis and Design Technique* (SADT) [188]. The semantics of the *activity* arrows in *Figure 4-19* is: *Input* represents entries to be transformed by the activity; *Output* represents results produced by the activity; *Control* represents means that influence in activity execution but are not consumed (i.e., languages, standards); and *Mechanism* represents means for executing the activity (i.e., machine, computer, person or tool). *Figure 4-19* shows that our methodology comprises two phases: *Domain Engineering* (DE), and *Application Engineering* (AE) [189, 190]. We envisaged these two phases because in the DE phase, the healthcare models are defined based on *open*EHR archetypes specification, and the agents are modeled in terms of their tasks, workflows and behaviors. These models are used to support application modeling, by generating the code that handles the agents combined with the *open*EHR archetypes. In AE phase, new applications are developed by reusing the artifacts obtained during the DE phase. In our methodology, we are prescribing tools and languages, but similar tools and languages could be used.

Figure 4-19
Methodology Overview



4.4.1 Domain Engineering

The DE phase involves the *Domain Specification* (DS), *Domain Design* (DD), and *Domain Implementation* (DI) activities, as shown in *Figure 4-20*.

Domain Specification

In the DS activity, the healthcare domain requirements are elicited, specified, analyzed and represented in models that express the knowledge about this domain. Examples of healthcare domain requirements are requesting laboratory results, making an appointment and checking resources available. *Domain Experts* are the stakeholders who have adequate clinical knowledge to ensure that the archetype models under development are in alignment with the clinical user requirements. For modeling clinical concepts, all items to be represented should be known. Any clinician or administrative support personnel working in the healthcare environment can perform initial gathering of the information used in clinical practice. In order to support the premise 'an archetype is defined as maximal data set for generic clinical concept' [191], item categories and their value ranges have to be identified from existing sources, such as datasets, healthcare information systems, electronic-based and paper-based publications and healthcare standards. These categories have to be combined into coherent and clinically meaningful concepts. In this activity, the *Domain Expert* analyzes the previous documentation system in order to identify all relevant items, which are represented as concepts. The goal of this activity is to define a hierarchical structure that classifies these items depending on their clinical meaning. From this hierarchical structure, the concepts are examined to identify repeated items and structures. The *Domain Expert* retrieves the relevant archetypes from the CKM repository, and specifies new archetypes using the Archetype Editor[26] and guided by the *open*EHR AM specifications in ADL. The output of the DS activity are the specified archetypes, which conform with the *open*EHR reference and archetype model and represent existing standard specifications and technical attributes, such as data constraints, occurrences and cardinalities. The *Domain Engineer* analyzes the current system in order to understand its goals, limitations, and requirements, and then describes the activities and use cases of the system, by identifying the agent types. The *Domain Engineer* models these agents with the UML tool. The UML tool and the Archetype Editor are the tools used to support the agent and archetype specifications, respectively. The outputs of the DS activity are the archetypes specification, the activity diagram for the agents' behavior specification,

[26] http://oceaninformatics.com/

and use case diagrams that represent the agents' interactions with the environment.

Domain Design

In the DD activity, the deliverables of the DS activity are used for defining the domain terminology, so that the *Domain Engineer* focuses on concepts, actions, predicates and their relationships. We adopt the following conventions to design the domain: '*concepts*' are denoted with substantives, such as patient, physician, and nurse; '*actions*' are denoted with verbal phrases, such as *MakeAppointment*, *RequestResource*, *RequestForMeeting*, and *RequestMedicalReport*; and '*predicates*' are denoted with Boolean assertions that are evaluated as true or false, such as *isBloodTypeAvailable*, *isICUBedAvailable*, and *isOperatingRoomAvailable*. In the DD activity, the agent types and their roles are identified, especially the agent behavior. The agent interactions are defined, determining *how*, *what* and *when* the various agents communicate. The DD outputs are the sequence diagrams that model the interactions between agents, and the class diagrams that represent the internal structure, behavior, and relationship among the agent types.

Domain Implementation

The DI activity uses the deliverables of the DD activity for identifying and creating the software agents that are assigned to the actors, with their tasks and behaviors defined according to their roles in the domain. The *Domain Engineer* uses the IDE for implementing the software agents, which are the DI outputs. At the end of the DI activity, tests are performed to determine whether it is necessary to iterate on the previous DE activities or whether the DE phase can be concluded.

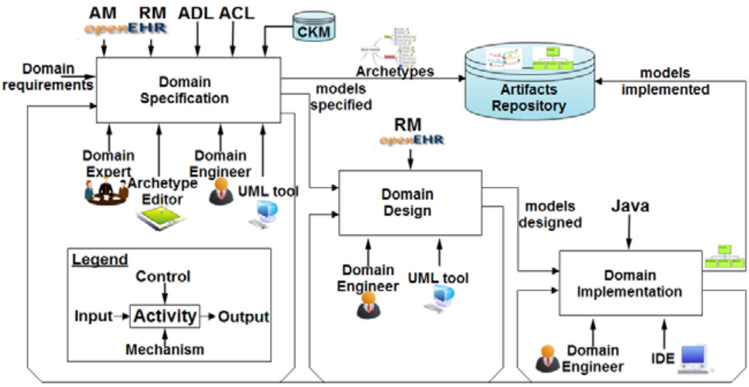


Figure 4-20 Domain Engineering Phase

The main goals of the *Domain Engineering* phase are to specify the archetypes based on the clinical concept domain and to extract the knowledge necessary for instantiating a particular agent in *Application Engineering* phase from the generic models. This knowledge corresponds to the agent behaviors, interactions with other agents and some other knowledge specific to a particular agent. We defined a set of ontologies that allow actors of the domain to communicate with each other in a language they understand.

4.4.2 Application Engineering

Application Engineering is a process in which a specific application is developed making use of the domain knowledge obtained during the *Domain Engineering* phase. The *Application Engineer* proceeds by analyzing the legacy systems functions, procedures and terminologies, and additional user requirements for the target application, selecting on appropriate domain model and completing the application development by reusing software components in a bottom-up manner. The AE phase involves the *Application Analysis* (AA), *Application Design* (AD), and *Application Implementation* (AI) activities, as shown in *Figure 4-21*.

Application Analysis

In the AA activity, the *Application Engineer* analyses the legacy applications and data, which have been probably implemented using languages, platforms and techniques older than the technologies considered in this work. Existing systems designed using structured data are analyzed so that potentially reusable components can be identified, such as source code, documentation, designs, requirements and whole subsystems. Afterwards, the *Application Engineer* analyses the additional requirements for the target application, and works with end-users in order to define the functionality that most resemble or accommodate what end-users really need. To identify the stakeholders and the application business processes, the *Application Engineer* uses UML use case diagrams, which are the AA outputs for documenting the functional requirements and for describing the application behaviors.

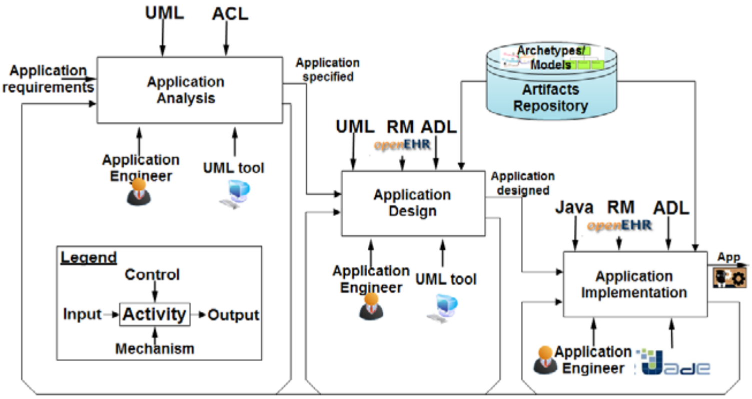
Application Design

The AD activity aims at identifying the potential reusable components of the domain problem, including the *open*EHR archetypes and the entities and relationships between the agents. For achieving this goal, the specifications are refined to model the application by considering hardware and software platforms, such as, e.g., Java EE and JADE. Using the deliverables of the AA activity, the *Application Engineer* models concepts as instances of the models built in the DE phase, and selects the classes and behaviors that are relevant to the application domain. The AD outputs are the class diagrams that describe the application structure, showing the relationship between the classes and their behaviors.

Application Implementation

In the AI activity, the application is implemented, including its communication interface, by integrating archetypes to *open*EHR Extracts. The application is implemented according to the model specifications produced before, and the IDE is the main tool to support this activity. The *Application Engineer* implements web service client interface in order to access and consume the remote web service interface provided by the Message Generator. Moreover, the other components that make up the application are developed, such as the user integration interfaces and data persistence services according to data access object (DAO) and data transfer object (DTO) patterns [192]. These components make use of a standard SQL relational database incorporating a Java-based Hibernate layer in order to associate table data to class instances. This offers efficient access to concrete clinical information, such as the results of specific laboratory tests and electrocardiograms. We applied JADE for the agents implementation, which facilitates the development of a multi-agent system and provides a standard implementation of the FIPA-ACL communication language.

Figure 4-21 Application Engineering Phase



At the end of the AI activity, tests are performed to determine whether it is necessary to iterate on the previous AE activities or whether the AE phase can be concluded. Testing enhances the integrity and reliability of the developed application, by detecting unintended deviations in the design and errors in the system, which can be caused by communication gaps between the *Application Engineer* and *Domain Experts*, and inadequate requirements gathering.

4.4.3 Discussion

In this section, we discussed a methodology through which a *Domain Expert* and a *Domain Engineer* can design personalized models, such as archetypes and agents models, respectively. Our methodology allows an *Application Engineer* to reuse the components of our reusable architecture, meant to be generally applicable to different legacy healthcare information systems, from which applications can be built to be used in realistic healthcare scenarios.

We adopted *Domain Engineer* and *Application Engineer* phases to define our methodology. The advantage of this separation is that two concerns are kept apart, (1) to build reusable and flexible domain artifacts and (2) to build specific applications by reusing available artifacts, adding in this way value to the legacy systems of the related domain. At the end of each phase, system tests to verify that it meets its requirements are performed to determine whether it is necessary to iterate on the previous activities or whether the phase can be concluded. In the *Domain Engineering* phase, the reusable domain artifacts generated, such as specified archetypes and modeled agents are stored in an Artifacts Repository to be reused in the next phase. The *Application Engineering* is a process in which a specific application is developed making use of the domain knowledge obtained during the *Domain Engineering* phase. Application artifacts comprise all development artifacts of a specific application, including the configured and tested application code.

The archetypes designed in this research and the message exchange between heterogeneous healthcare information systems is discussed in Chapters 5 and 6, respectively. Agent modeling process is discussed in Chapter 7. In Chapter 8, we discuss the application development, based on three case studies, according to our methodology in order to design applications that allowed us to shift from the legacy healthcare system that relies on informal offline communication between actors to an agent-based system.

Chapter 5

Archetypes

Healthcare professionals need to share patient information in order to support longitudinal continuing care and to follow multi-professional care pathways. The possibility of sharing EHR is therefore central to the pervasive healthcare model. Therefore, there is also consensus that the records that are only human readable are not sufficient to support this shared case: patient safety management and the pursuit of evidence-based care require computable information that can be linked to and queried by alerting healthcare systems. Efficient message exchange requires that the clinical information within EHRs are represented and organized consistently across healthcare providers, which characterizes the *semantic interoperability*. The *open*EHR dual model is a proposed standard solution to support the EHR communication through the use of archetypes.

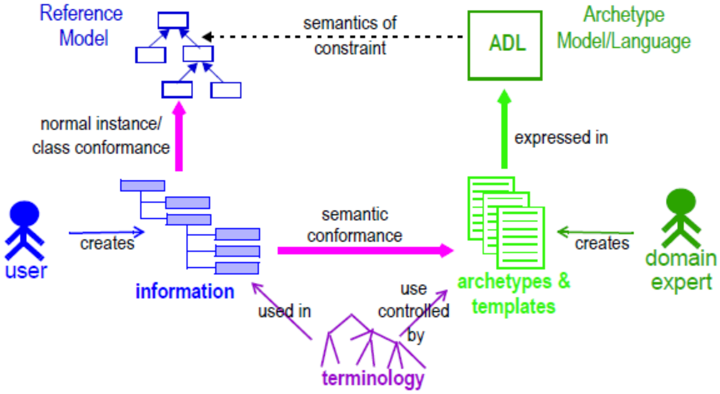
This chapter discusses how the domain experts may develop the archetypes and how to reuse the existing archetype. This chapter is organized as follows: Section 5.1 describes the concepts of archetypes and templates applied to multi-level modeling of healthcare systems. Section 5.2 gives an overview of the archetype development process, in which the domain experts are guided to perform their tasks. Section 5.3 presents the archetypes modeled and reused by domain experts in the cardiology domain.

5.1 Concepts

Multi-level modeling of healthcare information system helps separate the tasks of the *Domain Engineer* (software engineer) from the tasks of the *Domain Expert* (healthcare professional). According to *open*EHR approach, the two-level modeling (or multi-level modeling), distinguishes between a *reference model* (information level) that is used to represent the generic properties of health record data, and an *archetype model* (knowledge level) used to represent the specific information that needs to be documented within health records in each clinical domain. The main benefit of this separation is the possibility to construct a domain specialist model (knowledge model) that is independent from the activities of the developers, who focus on the reference model. In healthcare information systems, software development can proceed separately from domain modeling and information modeling, so that if new concept models are introduced, the software does not need to be redesigned, coded, tested and deployed [144].

Archetypes and *templates* play a key role in semantic interoperability in the *open*EHR dual model, as shown in *Figure 5-22*. *Archetypes* (ADL 1.4) define what is maximally documented in the world about a specific health record entity. *Templates* define what in a specific context is stored, retrieved, presented, exchanged and archived at a specific point in time [193].

*Figure 5-22 open*EHR Dual Model (Source: Beale, 2007)



5.1.1 Archetypes

Clinical archetypes play an important role in determining *how* clinical information is represented and organized inside EHRs and *when* they are communicated between systems. Archetypes often also influence the way in which clinical information is managed within individual EHR systems, how end-users enter information and how information is presented. An archetype defines a data structure, including optionality and multiplicity, data value constraints, and relevant bindings to natural language and terminology systems [194], being essential to achieve semantic interoperability between healthcare systems.

Archetypes can be divided into classes according to their content, and examples are shown in *Table 5-2* [145]:

- *composition-archetypes*: These archetypes correspond to common clinical documents or events and typical examples include *Prescription* and *Operative Notes*.
- *section-archetypes*: These organizational archetypes correspond to document headings commonly used to provide a structure in which to place the *Entry* and *Cluster* archetypes, which contain most detailed clinical content.
- *admin-entry-archetypes*: These archetypes are used for recording administrative events, such as admission, discharge, consent.
- *entry*: These descriptive archetypes are divided into the categories*:*
    - *observation-archetypes* (e.g. 'blood pressure'),
    - *evaluation-archetypes* (e.g. 'adverse reaction'),
    - *instruction-archetypes* (e.g. 'medication order'),
    - *action-archetypes* (e.g. 'procedure').
- *structure-archetypes*: These archetypes are used to model a structure (e.g., a tree or list) of items and reuse this structure in other archetypes.
- *cluster-archetypes*: These archetypes are used to model a cluster of items and to embed this cluster in other archetypes.
- *element-archetypes*: These archetypes are used to model a single item with the intention of reuse.
- *demographic-archetypes*: These archetypes are used to model demographic data and identities, addresses, personal relationships and roles.

Table 5-2 Archetypes classes, ID and descriptions

Archetypes classes	Archetype ID and Description
Composition	***openEHR-EHR-COMPOSITION.prescription.v1*** - Set of medication orders communicated to pharmacy
Section	***openEHR-EHR-SECTION.conclusion.v1*** - Section to record conclusions of an encounter with a patient
Admin-Entry	***openEHR-EHR-ADMIN_ENTRY.admission.v1*** - Used for admitted patient only. It signals the beginning of a patient's stay in a health care facility
Observation	***openEHR-EHR-OBSERVATION.apgar.v1*** - Clinical score derived from assessment of respiratory effort, heart rate, reflex irritability, muscle tone and skin color
Evaluation	***openEHR-EHR-EVALUATION.risk.v1*** - Evaluation to indicate that there is, or is not, a significant risk of this subject of care having, now or in the future, a condition
Instruction	***openEHR-EHR-INSTRUCTION.transfusion.v1*** - The instructions for giving a transfusion

Action	***openEHR-EHR-ACTION.procedure.v1*** - A clinical activity that has been carried out for therapeutic or diagnostic purposes
Item-Tree (Structure)	***openEHR-EHR-ITEM_TREE.follow_up.v1*** - Record details about a follow up
Cluster	***openEHR-EHR-CLUSTER.device.v1*** - Any physical object, product, instrument, apparatus, appliance, material or other item used directly or indirectly in the provision of healthcare, but is not used primarily through therapeutic or metabolic means
Element	***openEHR-EHR-ELEMENT.menstrual_cycle_day.v1*** - Number of days since onset of last normal menstrual period
Demographic-Cluster	***openEHR-DEMOGRAPHIC-CLUSTER.person_identifier.v1*** - Used in demographic services to register data about a person identifier
Demographic-Address	***openEHR-DEMOGRAPHIC-ADDRESS.address.v1*** - Used to communicate with, identify service provision requirements and trends, and to provide direct services
Demographic-Organization	***openEHR-DEMOGRAPHIC-ORGANISATION.organisation.v1*** - Representation of an organization's demographic data
Demographic-Person	***openEHR-DEMOGRAPHIC-PERSON.person-patient.v1*** - Representation of a patient's demographic data
Demographic-Role	***openEHR-DEMOGRAPHIC-ROLE.healthcare_provider_organisation.v1*** - Representation of an organization's demographic data in the role of healthcare provider

Archetypes require detailed descriptions of the clinical concepts such as rules, measurement ranges, data types, presentation formats, most appropriate data representation (terminology, codes), ontology, data to be included to assist decision making, data integrity constraints and allowed units with associated numeric ranges, such as, for example, the Apgar score [195] and aspects related to drug administration. The content in each archetype should be able to be interpreted in isolation and be as complete as possible to suit multiple sectors, purposes and priorities [196].

Archetypes support the binding of data nodes, names and values to concepts from external terminologies, such as SNOMED-CT. They are also multi-lingual, and support all languages that can be expressed in Unicode. *Term binding* is an ADL construct used to associate a language-independent label (e.g., at0005) with a specific *term* from a specific *terminology*. This allows for the name of a data point or archetype node to be identified as being the same as a specific *term* in a *terminology* [145].

5.1.2 Templates

An *open*EHR template is defined as a specialized *open*EHR archetype. *open*EHR *archetypes* provide flexibility and adaptability and define what makes sense and *templates* define what makes sense in a specific context. Templates provide the means of reusing archetypes in a local context without jeopardizing the archetypes. Using archetypes, healthcare providers can define and re-define at any moment templates that are needed in their work process at each specific point in time. Systems based on archetypes and templates facilitate customization and rapid adaptation to meet new clinical requirements. Templates are designed based on what is desired to be in a screen form or report, what archetypes are already available, and the local usage of terminology [144].

Since a template is a specialized archetype, it cannot change the semantics of archetypes they specialize, since it obeys the rules of the archetypes it specializes. Accordingly, all data created due to the use of templates are guaranteed to conform to the referenced archetypes, as well as to the underlying *open*EHR reference model. Templates may only specialize *existing nodes* defined in the flat parent, defined either in another template or in an archetype. Therefore, templates cannot add new data nodes. If new data nodes are required in the template context, appropriate specialized archetypes should be defined prior to their use in the final template. Templates are necessary to:

- combine items from different archetypes for several purposes;
- assign default values to items;
- link archetypes to clinical terminologies such as SNOMED CT; and
- otherwise constrain archetypes, such as, for example by deleting optional items of an archetype that are not required in the local context.

Although templates are expected to be expressed in ADL, templates form a third layer above the archetypes and the reference model that enables developers to create future-proof systems with a Graphical User Interface (GUI) that is flexible enough to present on the screen the information that the templates require. A template can be defined as a large structure where various archetypes are aggregated. It can also simplify the archetype structure by removing elements or nodes that are unnecessary, and setting default values or further constraints specifically adapted to a local implementation [144]. Templates are closely related to archetypes because they are able to set constraints, establish terminology bindings and fulfill slots where other archetypes can be attached.

Template Definition

*open*EHR templates are defined in the template package as Operational Templates. They differ from archetypes in that their root object is an *ARCHETYPE_ROOT,* and the Operational Template can add the ontology structures from the constituent archetypes [144]. An *open*EHR template is a specification that defines a tree of one or more archetypes, each constraining instances of various reference model types, such as COMPOSITION, SECTION and ENTRY subtypes. Thus, while there are likely to be archetypes for such things as *biochemistry results* (an OBSERVATION archetype) and *SOAP headings* (a SECTION archetype), templates are used to combine archetypes to form whole COMPOSITIONs in the EHR, such as for *discharge summary* and *antenatal exam*. Templates usually correspond closely to screen forms, printed reports, and in general, complete application-level lumps of information to be captured or exchanged; they may therefore be used to define message content. Templates are generally developed and used locally, while archetypes are usually widely used.

Archetype Redefinition

The most common archetype redefinition used in a template definition is the removal of unwanted nodes and sub-trees. This is because most archetypes are designed as *maximal data sets* (requiring universal concept consensus), and include a large set of data point definitions that would ever be used in a particular context. Two types of removal are often required: removal of one or more object children of a container attribute, and removal of a whole attribute (container or single-valued) [191]. *Figure 5-23* gives an example of the first situation, in which the *OBSERVATION.blood_pressure* archetype originally defines the following data points: systolic blood pressure, diastolic blood pressure, mean arterial pressure (MAP) and pulse pressure. In any particular clinical context, the only meaningful combinations that can be used are *systolic and diastolic* (typical general practice, routine hospital bedside use), *mean arterial pressure* (used as the perfusion pressure by anesthetists) or *pulse pressure* (difference between systolic and diastolic, used in many modern monitoring machines). Thus, the template can define which of the above combinations is allowed in the final data. This is done by defining exclusion constraints of the unwanted nodes by setting of the occurrences constraint to {0}, indicating that no occurrences are allowed.

Figure 5-23 Object exclusion (Source: Beale, 2007)

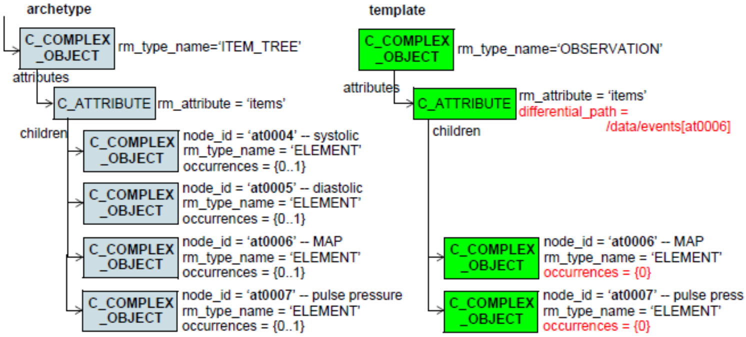
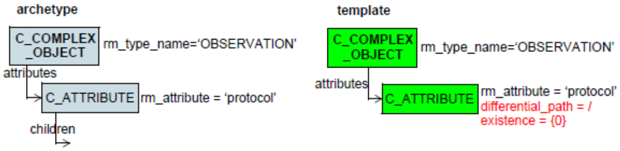


Figure 5-24 shows an example of the second situation with the removal of the *OBSERVATION.protocol* attribute from the blood pressure archetype by a template for which no protocol information (instrument type and measurement location on body) is needed.

Figure 5-24 Attribute exclusion (Source: Beale, 2007)



Operational Template

An *Operational Template* is the final usable form of a template and its constituent archetypes and overlays, and has only two differences from a flat archetype definition: (1) the root object is a *C_ARCHETYPE_ROOT* rather than a *C_COMPLEX_OBJECT*, and (2) ontology structures from the constituent archetypes are added. The first difference is because in a flattened archetype all archetype root points (including the top one) are replaced by *C_ARCHETYPE_ROOT* objects, which carry the relevant archetype identifiers. The second difference is because the flattening process usually involves more than one archetype, due to slot filling, meaning that the operational template has to explicitly include the flattened ontologies of all component archetypes in addition to its own ontology.

The operational template has been designed to function as a self-standing computable structure, containing all archetype and reference model elements relevant to runtime use, while resolving or removing design-time elements such as internal references. The resulting structure is essentially like a single large archetype, containing individual archetype identifiers at all the archetype root points, and also carrying the sum of all the archetype ontologies, enabling the at-codes (e.g., at0005) from each source archetype to be resolved at runtime.

5.2 Archetype Development Process

The cardiology domain knowledge is complex and related to several areas of healthcare. Its knowledge concepts can be represented in multiple ways using many terminologies or coding systems, such as, for example, SNOMED-CT[27] and ICD-10[28]. *Domain Experts* need to be able to document their application knowledge in a manner that enables domain engineers to understand the problems to be solved so that EHRs meet operational clinical practice needs. Furthermore, hospital information systems that support this domain need to be able to exchange data, information and knowledge without loss of meaning, which characterizes *semantic interoperability*. According to Booch and Jacobson [197, 198] an 'object model reduces communication problems between analysts and users'. Object-orientation has been widely applied to requirements analysis and design. With the *open*EHR release 1.0, a common model is available to solve some of the problems related to accessing information and knowledge by improving semantic interoperability between EHR systems.

Archetypes are designed by *Domain Experts*, and often require significant study of a subject area, such as *cardiology* and *obstetrics*. The development process may occur at a national or international level, and requires peer review and testing in real systems, according with the semantic value of archetypes, namely as reusable models of content. Consequently, from the point of view of any given site of deployment, archetypes are most likely to have been developed elsewhere, and to reside in a recognized, quality assured repository.

Archetype development must be coordinated through *Domain Knowledge Governance,* which is defined as 'comprising all tasks related to establishing or influencing formal and informal organizational mechanisms and structures in order to systematically influence the building, dissemination, and maintenance of knowledge within and between domains' [199]. Archetypes that reflect the knowledge of specific clinical knowledge domains can only be defined and maintained by the domain professionals.

We have developed cardiology domain archetypes by taking into consideration the national and international guidelines, such as *Cardiology Brazilian Society Guidelines*[29], *Guidelines for Percutaneous Coronary*

[27] http://www.ihtsdo.org/snomed-ct/

[28] http://www.who.int/classifications/icd/en/

[29] http://publicacoes.cardiol.br/

Intervention[30], *American College of Cardiology*[31], *European Society of Cardiology*[32] and *British Cardiovascular Society*[33]. The healthcare professionals involved in the cardiology network have collaborated to ensure that the outcomes that were consistent with these guidelines could be achieved. The aim of this collaboration was to develop a process for capturing clinical knowledge relevant to the cardiology domain in a manner that could facilitate clinical information exchange by many stakeholders in accordance with the *open*EHR archetypes.

We defined the archetype development process based on the integration of two existing processes for capturing expert clinical knowledge: (1) the process adopted for the development of clinical practice guidelines [200], and (2) a consensus-based standards development process [201, 202]. Archetypes need to be evidence-based and be adopted as a formal representation of clinical knowledge. The clinical concepts represented within archetypes need to be represented in a standard way and appropriately linked to clinical terminologies.

In the archetype development process, healthcare professionals participate in order to define the structure of clinical concepts as they are used in EHR systems and the relationships between them. This process involves many stakeholders with different backgrounds who collaborate to achieve a consensus in the definition of these clinical concepts.

In this research, we followed the *Domain Engineering* approach [203, 204] as a systematic process to capture the clinical concepts of a domain knowledge and to design the archetypes for general reuse. This is an iterative process to achieve the continuous refinement of the artifacts. Organizations plan their governance, organizational setting and archetype requirements. Using the *Structured Analysis and Design Technique* (SADT) diagram notation, *Figure 5-25* shows an overview of the archetypes development process, which consists of the following collaborative processes: *Archetype Analysis, Archetype Design* and *Archetype Implementation*.

[30] http://sbhci.org.br/profissional/

[31] http://www.cardiosource.org/

[32] http://www.escardio.org/

[33] http://www.bcis.org.uk/

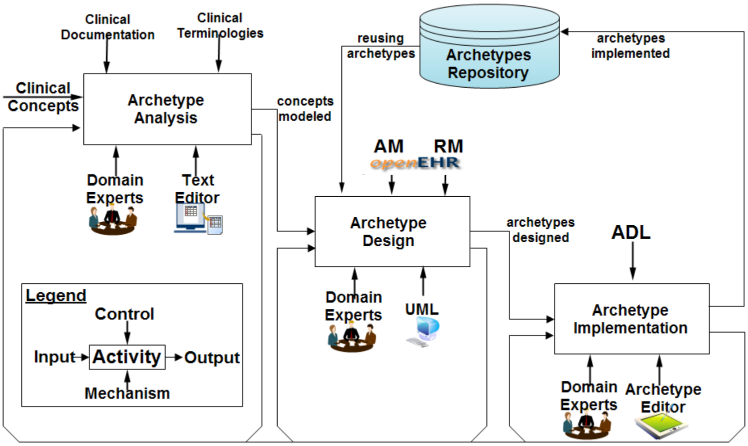


Figure 5-25 Archetype Development Process

5.2.1 Archetype Analysis

In the *Archetype Analysis* activity, the clinical concepts requirements are elicited, specified and analyzed in a brainstorming session with the *Domain Experts*. The *Domain Experts* are all potential stakeholders and have the necessary clinical knowledge to ensure that the archetype models under development are aligned with the clinical user requirements. For modeling clinical concepts all items to be documented should be known, such as the arterial blood pressure measurement details (e.g., person position, normal range and unit of measurement). Any clinician or administrative support personnel working within the healthcare environment can gather initial information that is used in clinical practice. Because an archetype should be 'a maximal data set for generic clinical concept', documented item categories and their value ranges have to be identified from existing sources, such as datasets, healthcare information systems, electronic-based and paper-based publications, healthcare standards and other knowledge from experts. The items have to be combined into coherent and clinically meaningful concepts.

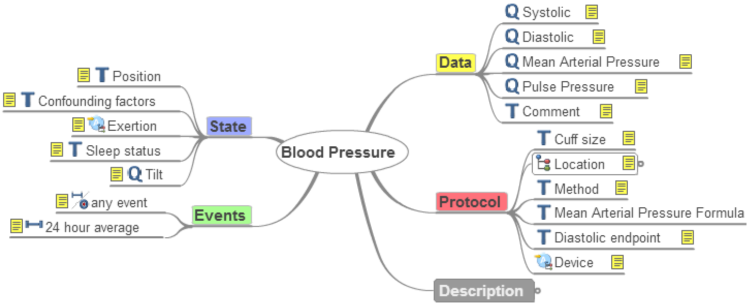
In this activity, the *Domain Expert* analyzes the previous documentation system in order to determine all items, which are documented as concepts. The goal of this activity is the definition of a hierarchical structure that classifies these items depending on their clinical meaning. From this hierarchical structure, the concepts are examined to identify repeated items and structures.

Based on the clinical concepts specified, the *Domain Expert* analyzes the applicable descriptive terminology and the list of terms and domain database tables for data element validation. Additional requirements might be needed in order to satisfy local legislations, or different translations could be needed depending on the number of the languages of the region where the archetypes are applied. The activity outputs are the concepts modeled that document the clinical concepts identified and their relationships.

5.2.2 Archetype Design

In the *Archetype Design* activity, the clinical concept specifications are refined using techniques such as *mind maps*. A mind map is a visual representation of hierarchical information that includes a central idea surrounded by connected branches of associated topics in order to better understand of relationship and connections between ideas and concepts. The clinical concepts are organized by focusing on the identification of data elements, treatment protocol, state, allowable events, pathway steps, and concepts needing terminology, as shown in *Figure 5-26*.

Figure 5-26 Blood Pressure Mind map



During this phase, items and structures are identified and duplication can hence be avoided. The *Domain Expert* selects the suitable reference model class related to each concept being modeled, such as *COMPOSITION*, *SECTION* structures, and various types of *ENTRY,* like *OBSERVATION*, *EVALUATION*, *ACTION*, *INSTRUCTION*, and *ADMIN-ENTRY*. Given that archetypes define constraints over the reference model classes, the domain experts are free to choose what reference model classes match the concepts that need to be modeled. The *Domain Experts* are helped by decision algorithm to select the appropriate classes in the *open*EHR reference model. *Figure 5-27* shows the decision algorithm to help them decide which archetype class to use to represent a concept [137, 196]. Depending on the kind of concept required the *Domain Experts* identify whether the concept is part of a document, a functionality, a standalone concept, a multiple or single element.

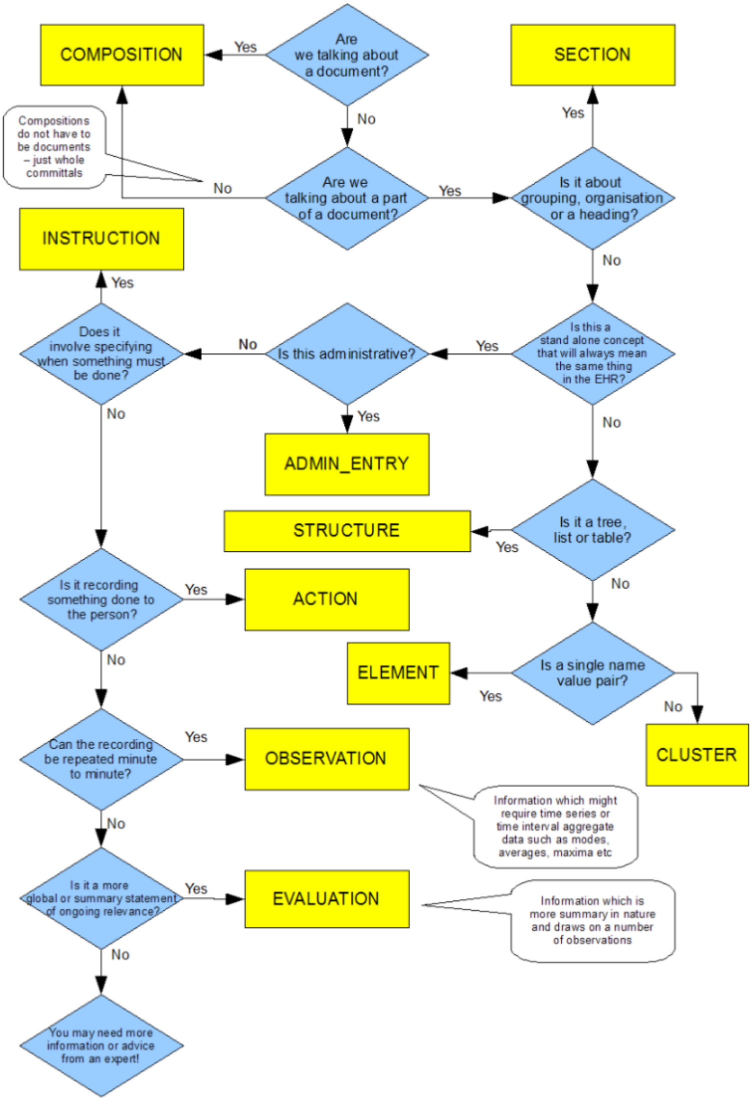

COMPOSITION
Yes
Are we talking about a document?
SECTION
No
Yes
Compositions do not have to be documents – just whole committals
No
Are we talking about a part of a document?
Yes
Is it about grouping, organisation or a heading?
INSTRUCTION
No
Yes
Does it involve specifying when something must be done?
No
Is this administrative?
Yes
Is this a stand alone concept that will always mean the same thing in the EHR?
No
Yes
No
ADMIN_ENTRY
STRUCTURE
Yes
Is it a tree, list or table?
Is it recording something done to the person?
Yes
ACTION
No
No
ELEMENT
Is a single name value pair?
Yes
No
Yes
Can the recording be repeated minute to minute?
OBSERVATION
CLUSTER
Information which might require time series or time interval aggregate data such as modes, averages, maxima etc
No
Is it a more global or summary statement of ongoing relevance?
Yes
EVALUATION
Information which is more summary in nature and draws on a number of observations
No
You may need more information or advice from an expert!

*Figure 5-27 open*EHR decision algorithm (Source: Garde *et al.*, 2009)

According to their clinical practice experience, *Domain Experts* can choose the suitable *ENTRY* class to represent a concept. The choice of these types is based on the clinical problem-solving process [205, 206], as shown in *Figure 5-28*.

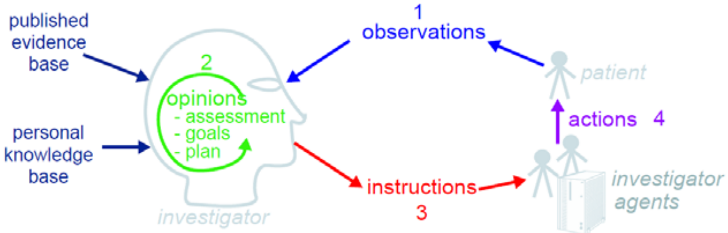


Figure 5-28 Relationship between information types and the investigation process (Source: Beale, 2008)

Figure 5-28 shows the cycle of information creation as an iterative problem solving process typical not just of clinical medicine but of science in general. The *system* to be observed as whole is made up of two parts: *patient system* and *clinical investigator system*. The later consists of healthcare professionals and is responsible for understanding the state of the *patient system* and delivering care to it. A problem is solved by making observations, forming opinions and prescribing actions (instructions) for next steps, which may be further investigation or may be a set of interventions designed to solve the problem, and finally executing the instructions (actions). The *open*EHR Entry model does not impose a process model, it only provides the possible types of information that might occur.

Figure 5-29 shows the concept of a *device*, which we use to demonstrate how the development of hierarchical structures can avoid modeling duplication. The *items* of the concept *device* can be modeled within a *CLUSTER* archetype.

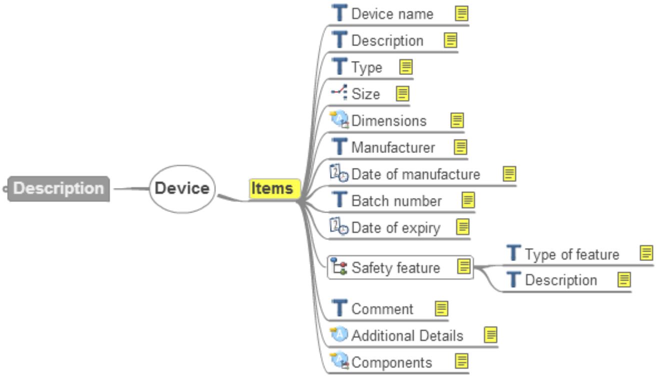


Figure 5-29 Mind Map for *Device* concept

Devices archetypes can be included in an *ECG Recording* archetype to record information details about the electrocardiograph device used to record an ECG (*ECG Device*) and to view the ECG outputs (*ECG Viewer*), as shown in *Figure 5-30*.

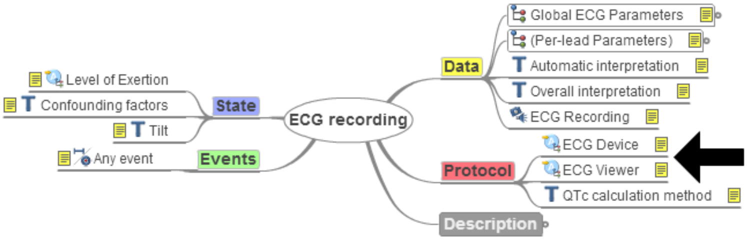


Figure 5-30 Mind map for ECG Recording

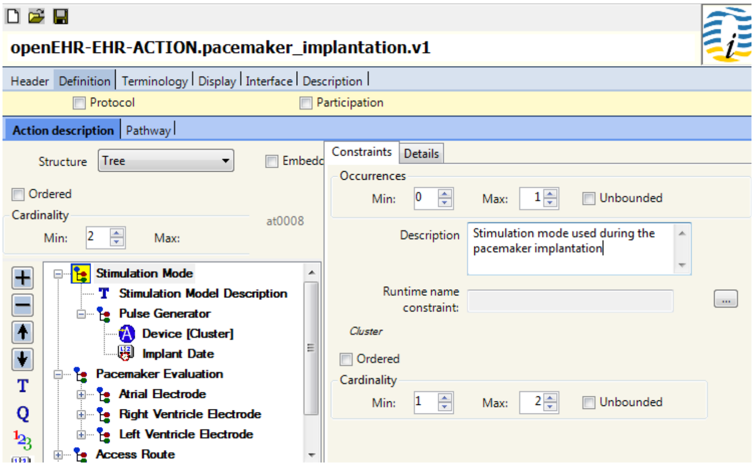
To ensure interoperability, already available archetypes are taken into account and the *open*EHR knowledge repository (CKM) can help the *Domain Expert* identify archetypes that model required items or whole concepts. Experienced clinical modelers are needed to conduct a review of existing archetypes in order to determine where definitions is already available and decide whether new archetypes need to be created. Knowledge of the current available archetypes and their attributes can help identify possible useful archetypes. If there are no appropriate archetypes for modeling the concepts we need to represent, then new archetypes have to be designed. The *Domain Expert* identifies the relationships to other archetypes or knowledge artifacts such as terminology subsets. As archetypes should be maximum data sets, most of the reusable archetypes contain more items than needed. These items can be excluded from the documentation via templates (see Section 5.1.2). When particular items are missing in an existing archetype, this archetype can be specialized and extended by adding the missing items. An example is the archetype *provider-address*, which specializes the archetype *address,* by adding new attributes related to the healthcare provider address.

5.2.3 Archetype Implementation

In the *Archetype Implementation* activity, the archetypes are implemented in ADL language using tool like the Archetype Editor[34] tool. Using Archetype Editor, the *Domain Expert* can create new archetypes based on the reference model classes. The designed archetypes can be combined within larger composite archetypes or templates. The *Domain Expert* performs a collaborative review of the metadata, content and translations of the archetypes, complementing the archetype modeling with the inclusion of terms lists, domain tables and terminologies, which were identified in prior activities. Term bindings can be added at any stage of archetype development. The *Domain Expert* can link a data point in the archetype with a code in a terminology, such as LOINC, SNOMED-CT and ICD10 with conditions if necessary. The *Domain Expert* can manage the implemented

[34] http://www.oceaninformatics.com/

archetypes using Archetype Editor, which includes regression testing functionality, full archetype parsing, validation, flattening and serialization to XML, ADL, dADL, JSON and YAML. *Figure 5-31* shows the *Pacemaker Implantation* archetype implemented using the Archetype Editor tool. The implemented archetypes must conform with *open*EHR reference and archetype model, and represent existing standard specifications or data sets and technical attributes, such as data constraints, occurrence/cardinality and null flavors. Since most of the existing archetypes are in English, they must be translated for use in other languages if necessary.



openEHR-EHR-ACTION.pacemaker_implantation.v1

□ Protocol □ Participation

Structure: Tree

□ Ordered

Cardinality Min: 2 Max:

at0008

Occurrences Min: 0 Max: 1 □ Unbounded

Description: Stimulation mode used during the pacemaker implantation

Runtime name constraint:

Cluster

□ Ordered

Cardinality Min: 1 Max: 2 □ Unbounded

- Stimulation Mode
  - Stimulation Model Description
  - Pulse Generator
    - Device [Cluster]
    - Implant Date
- Pacemaker Evaluation
  - Atrial Electrode
  - Right Ventricle Electrode
  - Left Ventricle Electrode
- Access Route

Figure 5-31 Pacemaker Implantation archetype implemented in the Archetype Editor

5.3 Application

We applied the steps discussed before together with the *Domain Experts* from *Cardiology Healthcare Network* (Marília - São Paulo, Brazil) in order to design the archetypes for the scenarios that resulted in 17 archetypes as shown in *Table 5-3*. The archetypes were developed by collaborative requirements meetings including two IT professionals, three cardiologists, two laboratory professionals, three nurses and two administrative professionals, to determine issues around content related to each expert area. IT professionals supported the healthcare professionals to perform *open*EHR modeling. Meeting occurred progressively through model development in proportion to the complexity of clinical concepts and availability of healthcare professionals. Terminologies played a key role within archetypes development, and the healthcare professionals could support the availability of suitable codes within the terminologies identified. Contents that required structured values utilized terminology subsets or were modeled explicitly within the archetypes as an internal code set.

Table 5-3 Archetypes designed

Archetypes classes	Archetype ID and Description
Action	***openEHR-EHR-ACTION.pacemaker_implantation.v1*** – Register clinical information about the pacemaker implantation
Evaluation	***openEHR-EHR-EVALUATION.pacemaker_evaluation.v1*** - Evaluation about a pacemaker implanted in a patient
Action	***openEHR-EHR-ACTION.classical_surgery.v1*** - Used to register an arrhythmia treatment and aneurysm repair
Action	***openEHR-EHR-ACTION.coronary_surgery.v1*** - Used to myocardial revascularization for coronary artery disease
Action	***openEHR-EHR-ACTION.vascular_surgery.v1*** - Used to diseases or disorders of the cardiovascular system
Action	***openEHR-EHR-ACTION.congenity_surgery.v1*** - Used to defect in the structure of the heart and great vessels which is present at birth
Action	***openEHR-EHR-ACTION.coronary_angioplasty.v1*** - Used to describe an open narrow or blocked coronary arteries
Action	***openEHR-EHR-ACTION.cardiac_catheterism.v1*** - Used to register the diagnose and treat some heart conditions
Evaluation	***openEHR-EHR-EVALUATION.echocardiography.v1*** - Used to evaluate the sonogram of heart
Evaluation	***openEHR-EHR-EVALUATION.stress_test.v1*** – Evaluate the information about how the heart works during physical stress
Evaluation	***openEHR-EHR-EVALUATION.ergometric_test.v1*** – Evaluate the identification or exclusion of ischemic heart disease
Observation	***openEHR-EHR-OBSERVATION.holter_monitor.v1*** – Used to continuously monitoring the heart's rhythms
Evaluation	***openEHR-EHR-EVALUATION.transesophageal_echo.v1*** – Assess the heart's function of patients
Evaluation	***openEHR-EHR-EVALUATION.doppler_echo.v1*** – Assess the flow of blood through the heart's chambers and valves

Evaluation	***openEHR-EHR-EVALUATION.carotid_scan.v1*** – Vascular ultrasound to assess the blood flow of the heart arteries
Evaluation	***openEHR-EHR-EVALUATION.tilt_table_test.v1*** – Evaluate the cause of unexplained syncope
Evaluation	***openEHR-EHR-EVALUATION.transesophageal_atrial_stimulation.v1*** – Evaluate the supraventricular arrhythmias

The archetypes were validated by the *Domain Experts* of *Cardiology Healthcare Network* (Marília - São Paulo, Brazil). The archetypes were published within the hospital network in order to receive comments from the health community and vendors who participated in the bidding process for healthcare integration. Through these initiatives, we could confirm that all concepts and its constraints could be correctly formalized in ADL code. The clinical team now wishes to submit these archetypes to the *open*EHR CKM community in order to contribute with the international archetype development process related to the cardiology domain archetypes. We reused some CKM archetypes in our architecture, such as *Device*, *Device Details, Prescription, Medication List, Medication Administration, Medication Action, Medication Order, Follow up Instructions, Laboratory Result Annotation, Laboratory Test, Blood Pressure, Blood Matching, Body Temperature, ECG Recording, Patient Admission, Clinical Synopsis* and *Demographic Models.*

Chapter 6

Message Generator

Network technologies have improved over the past decades, so that more powerful computer and communication capabilities have become available. It has fostered development of distributed software systems. Much of the complexity of building this software systems can be alleviated by the use of highly flexible, efficient and secure middleware. Middleware is infrastructure software that resides between the applications and the underlying operating systems, networks and hardware, specifically intended to provide a more appropriate platform for building and operating distributed systems [207].

This chapter is organized as follows: Section 6.1 presents the approach that we used to develop our middleware namely the Message Generator. Section 6.2 explains the Extract and Request structures according to the *open*EHR Extract Information Model. Section 6.3 gives an overview of the implementation of our Message Generator. Section 6.4 explains the components implemented in the Message Generator, and gives an example in the Cardiology Domain.

6.1 *open*EHR XML Schemas

Our proposed middleware namely Message Generator was designed to bridge the gap between healthcare applications and provide reusable services to distributed healthcare systems by allowing the semantic interoperability of heterogeneous healthcare information systems.

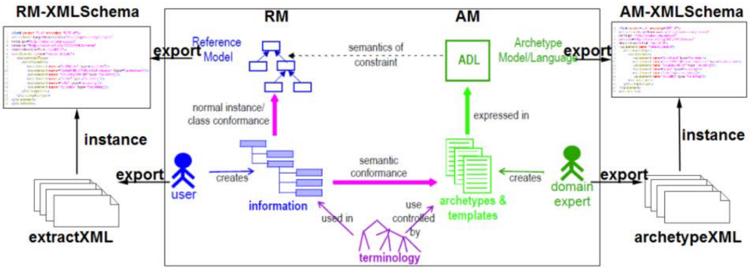
We developed facilities to perform message exchange between heterogeneous EHR systems in pervasive healthcare environments. We followed the *open*EHR Extract Information Model (IM)[35] that describes the

[35] http://www.openehr.org/programs/specification/releases/1.0.2 (in development)

several ways in which an EHR extract that contains a total or partial EHR can be built to support interoperability between EHR systems.

We designed a Message Generator Schema based on *open*EHR Extract Information Model. Since the *open*EHR standard consists of a RM part for delivering the container with the needed EHR information, and an AM part for expressing clinical content, the Message Generator Schema has also a Reference Model Schema (*RM-XMLSchema*) for representing the constraints in RM, and an Archetype Model Schema (*AM-XMLSchema*) for representing the clinical archetypes. *RM-XMLSchema* is the concrete model from which the software can be developed. *AM-XMLSchema* represents the concrete metamodels of a domain concept, which are expected to be understandable for a domain expert. *Figure 6-32* depicts an overview of the Message Generator schema.

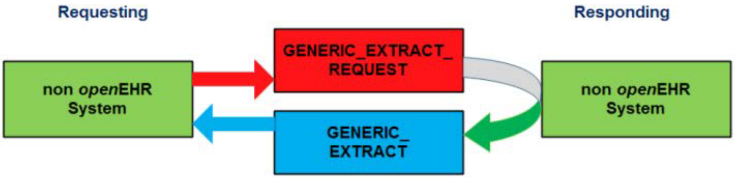
Figure 6-32 Message Generator schema



6.2 Request and Extract

In the *open*EHR Extract IM, the notions of *Request* and *Extract* (the reply) are clearly distinguished. An *Extract* may include a copy of the *Request* and an indication of its content. The common semantics of *Requests* and *Extracts* is modeled in a generic way, using a number of specialized *Request* and *Extract* types that are based on common classes. Different *Extract* concrete types are employed to satisfy particular groups of requirements, instead of applying one kind of *Extract* to perform all possible tasks. *Figure 6-33* illustrates the communication scenario of our Message Generator, where non-*open*EHR systems can exchange health information using concrete *Request* and *Extract* types defined by the *open*EHR Extract IM.

Figure 6-33 Communication scenario based on *open*EHR Request-Extract



In our Message Generator, the *Requests* and *Extracts* are implemented as types in a *Web service* environment, to support the semantic interoperability among distributed HIS. A *Request* contains a detailed specification of the information from the records of one or more subjects. A *subject* may be a patient EHR or a Person record in a demographic system. An *Extract* contains *chapters*, which contains *folders*, which contain *content items*, which in turn contain all the requested content that has been possible to retrieve. Archetypes and templates for the *Extract/chapter* are used for managing the content within *folders* in each *chapter*, and across *chapters*.

6.2.1 EXTRACT Structure

Figure 6-34 shows the *open*EHR Extract class diagram, in which the *EXTRACT_REQUEST* class has an *update_spec* attribute that specifies how the *Request* is to be processed by the server, and an *extract_spec* attribute that indicates the required information from the target healthcare information system. The *EXTRACT* class has a *request_id* attribute, an optional *participations* list, an optional *specification*, and a set of *chapters* that contain the retrieved content. The *participations* attribute denotes the subjects of the *EXTRACT_PARTICIPATION*, which are termed *parties*, since they are instances of the *PARTY* class. *Participations* are responsible for creating the *Extract*. The specification attribute has the same form of the *extract_spec* of the *EXTRACT_REQUEST*, indicating the actual *Extract* contents. An *Extract* content is enclosed within the *chapters* attribute in the form of one or more instances of the *EXTRACT_CHAPTER* class or of its subtype *EXTRACT_ENTITY_CHAPTER*. Within an *EXTRACT_CHAPTER*, the *items* attribute contains a *folder* structure composed of *EXTRACT_FOLDER* objects, which in turn contain the content *items*. The *folder* structure can be used to separate the parts of a health record, or to group items according to some other scheme (e.g., an *episode*). The *EXTRACT_ENTITY_CHAPTER* class represents a *chapter*, which is used to carry content associated with a single entity (e.g., a *patient*). An *Extract* may contain data related to multiple patients, which means that it may contain multiple *chapters*.

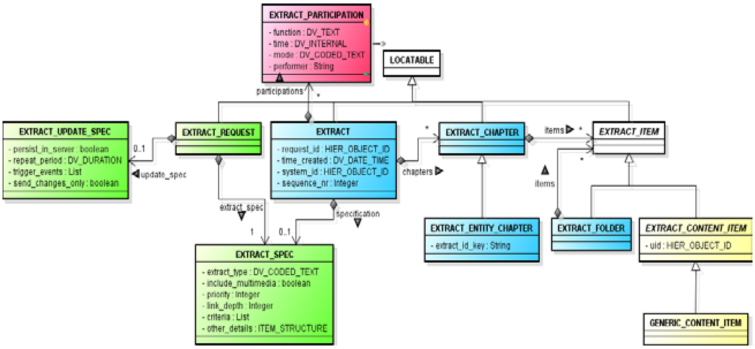


*Figure 6-34 open*EHR Extract class diagram

Extracts usually include clinical information and demographic data related to *participations* of the subject and other *parties* in the documented activities in the clinical information. Within an *Extract*, *participations* are represented in two places: in the *EXTRACT* class, *participations* are used to provide information about parties related to the *Extract*; and in the *GENERIC_CONTENT_ITEM* class, which represents the EHR actual content, the original structures of *participations* are copied faithfully, including any in-line *performer* information. *GENERIC_CONTENT_ITEM* defines metadata that can be populated with information from legacy systems, such as patient and healthcare professional demographic contents.

6.2.2 EXTRACT-REQUEST Structure

We designed the *open*EHR Extract IM in our Message Generator as a shared EHR system, aggregating data from various sources in a standardized way in order to allow the message exchange among heterogeneous healthcare information systems. The logical structure of the source data is completely preserved within the *Extract*, since only the reference is rewritten to the *UID* values used in the *Extract*. *Figure 6-35* illustrates the typical structure of the *EXTRACT_REQUEST* (a) and the *EXTRACT* (b). The *Extract* is organized in *chapters* that contain ‘EMR’ and ‘demographic’ information, and the *chapters* contain *folders* that enclose ‘patients’ and ‘providers’ data. The *content items* contain the requested content related to ‘COMPOSITION’, ‘PERSON’, and ‘ORGANISATION’ instances. The *participations* represent the demographic content of the *Extract*, identifying the ‘requester’ and ‘consultant’.

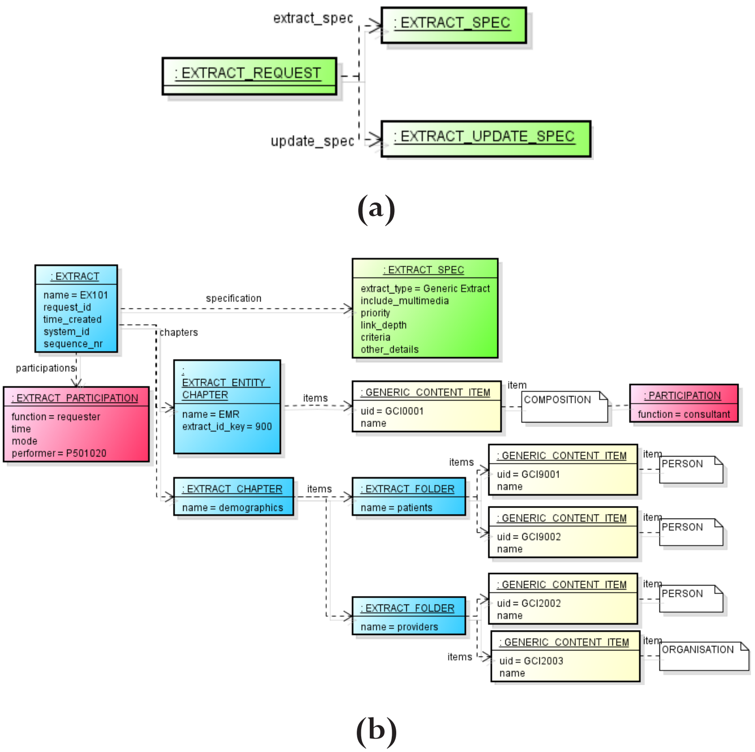


Figure 6-35 openEHR Request and Extract structure example

6.3 Implementation Overview

The Cardiology Department of Hospital Santa Casa defined a project to develop a Cardiology Healthcare Network for several healthcare providers in Marília (São Paulo, Brazil) in order to allow the integration of heterogeneous healthcare information systems in the Cardiology Healthcare Network. We designed a Message Generator aimed at managing patients' demographic and clinical information for message exchange within the cardiology network.

The Message Generator proposed for this research was developed to support different clinical specialties related to the cardiology domain in order to handle each one of them in healthcare environment. Each healthcare provider should use their own healthcare information system and have a local datacenter independent of each other. The Message Generator is a messaging infrastructure compliant with the *open*EHR specifications, aimed at ensuring the interoperability of the EHR systems of the participants of the Cardiology Healthcare Network. The decision to use these specifications was made mainly because their focus on EHR systems interoperability. The *open*EHR approach prescribes the use of archetypes to represent knowledge and includes four basic services: *clinical data services, demographic data services, archetype services* and *terminology services*.

The decision to use the archetypes repository in the Message Generator was because of the flexibility that these artifacts can offer for medical knowledge reuse. Additionally, our approach enabled the cardiology network team to take direct responsibility for any evolutionary changes in the concepts specific to the healthcare domain, and for any new clinical data representations, while software engineers would bear the technological responsibility of maintaining the archetypes processing infrastructure and managing the servers that host the Message Generator repository, which stores the archetypes, templates and terminologies applied in our solution.

The Message Generator was built in accordance with the Service-Oriented Architecture. Demographic and clinical data services were specifically designed to allow the exchange of patients' clinical data between EHR systems, and the Message Generator has been designed as a middleware. These clinical data are called *open*EHR extracts, which were developed based on the *open*EHR reference model (Section 3.3) and complemented by a set of archetypes that define the central repository concepts (Chapter 5).

6.3.1 *open*EHR Java Implementation

The Message Generator has been built on top of the *open*EHR Java Implementation[36] for implementing new classes in the RM, and the eXtensible Markup Language (XML) for encoding the EHR Extracts to be exchanged. The *open*EHR Java Implementation consists of the following software components:

(a) *openehr-rm* implements all RM specifications, such as Data Types, Data Structures, Common, Support, EHR Extract, and Demographics. It maps *open*EHR data types onto Java native data types (e.g., Java String), and implements other higher level data types (e.g., DV_TEXT, which is a text item that contains any amount of valid characters arranged as words or sentences). It also implements path-based queries for finding leaf nodes in large object trees with single crafted paths. All constructors of the RM classes have annotations for its parameters, so the invocation of an object can be done automatically. This feature is used by *rm-builder* to construct RM objects from archetypes;

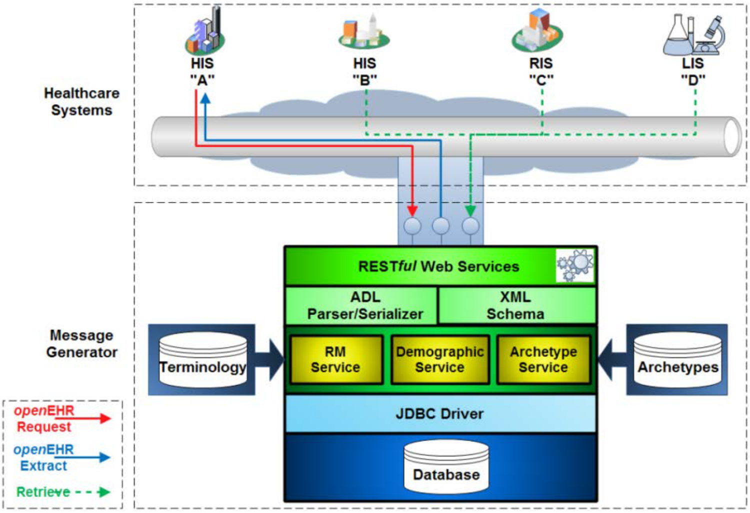
[36] http://www.openehr.org/programs/specification/releases/currentbaseline

(b) *openehr-aom* and *openehr-ap* provide in-memory representation of archetypes, which are usually authored and transmitted in ADL format;
(c) *rm-builder*, which is grounded on the *openehr-rm*, *openehr-aom* and *openehr-ap* components, provides support to create and validate archetype-based objects, and guarantees that the archetype semantics is faithfully reflected when constructing and validating RM objects;
(d) *adl-parser* provides support for parsing ADL descriptions by transforming ADL archetypes in textual format to the in-memory AM format. The *adl-parser* is the entry point for the archetype in any EHR system, so it is critical for the proper operations of these systems; and
(e) *adl-serializer* provides support for the conversion of archetypes from the in-memory AM format to the ADL textual format. It is often used before the storage and transmission of an authored archetype.

6.4 Components – Cardiology Domain Example

RM and AM constraints are implemented using XML schemas, and in the Message Generator, we employed the adapted XML Schemas prescribed in the *open*EHR specifications. We designed the Message Generator as a software application that offers and uses web services, and we built REST*ful* Web services [208] that transmit data directly on top of the HTTP protocol. *Figure 6-36* shows the general structure of our solution.

Figure 6-36 Message Generator structure for interoperable EHR



The Message Generator has *Database* and *JDBC Driver* components, which employ the MSSQL-Connector driver to access through JDBC the MSSQL databases that store the clinical information. Based on the requested information received by Web services, the connection between the Message Generator and distributed HISs is established through JDBC. The *Database* is responsible for storing clinical information in the Message Generator repository based on the archetype nodes.

RM Service, Archetype Service and *Demographic Service* are the services provided by the Message Generator, which employs the three main back-end services used in the *open*EHR Java Implementation: the *RM Service (openehr-rm* and *rm-builder)*, for understanding the classes of the *open*EHR RM; the *Archetype Service (openehr-aom)*, for understanding the format and the syntax of the archetypes; and the *Demographic Service (openehr-ap)*, for managing the identifiers of all parties involved in the medical procedure. These services offer internal functions, such as *getComposition*, *getParticipation*, *getContentItem* and *setExtract*, which allow the manipulation of the records gathered into the Message Generator. These services have been implemented by using XMLBeans[37], which is a technology for generating Java classes from XML schemas. The generated Java classes can be used to parse or generate XML documents.

ADL Parser/Serializer component (*adl-parser* and *adl-serializer*) consists of the parser and serializer tools that are included in the *open*EHR Java Implementation for processing archetype instances. The XML Schema component represents the Reference Model Schema (*RM-XMLSchema*) for representing the constraints defined in the *open*EHR RM, and an Archetype Model Schema (*AM-XMLSchema*) for representing the *open*EHR archetypes, as shown in *Figure 6-32*.

The *REST*ful *Web services* component provides a set of services for allowing the query and retrieval of clinical information and demographic data from EHRs. This component has interfaces to communicate with the client applications using REST*ful* Web services. The main Web service interfaces we developed are *openEHRRequest*, *openEHRExtract* and *Retrieve*, which expose the internal capabilities of the Message Generator.

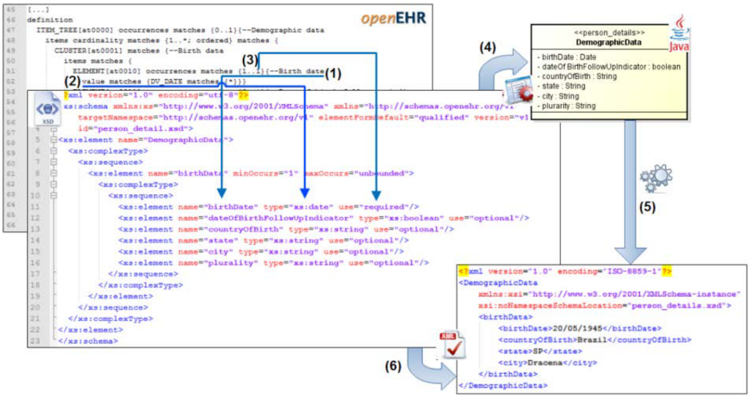
Figure 6-36 shows HIS 'A', which uses the *openEHRRequest* service to request clinical information about an EHR. The Message Generator processes the *Request* that contains the detailed specification of the required content, such as, e.g., ECG Report (from HIS 'B'), Radiology Report (from RIS 'C'), and Report for Clinical Laboratory (from LIS 'D'). All the clinical information is queried and retrieved, and the Message Generator serializes one *Extract* for each HIS that generates messages to be exchanged. The serialized *Extracts* are transformed in XML documents that are

[37] http://xmlbeans.apache.org/

forwarded back to HIS ‘A’. Using our Message Generator, any healthcare system can request clinical information and demographic data through the *openEHRRequest* interface.

Figure 6-37 shows the XMLBeans code (*extractXML*) generated according to the ‘*person_details*’ *archetype* instance: *birthDate* (1) is an attribute that was generated based on the RM; *xs:date* (2) is the type of this attribute; *required* value (3) is a mandatory attribute; the *DemographicData* Java beans class (4) corresponds to the *person_details* XML Schema; the XML instance (5), containing demographic information based on the Java beans instance, is generated in conformance with the XML Schema (6).

Figure 6-37 Generated XML Code



```
[...]
definition
  ITEM_TREE[at0000] occurrences matches {0..1}{--Demographic data
    items cardinality matches {1..*; ordered} matches {
      CLUSTER[at0001] matches {--Birth data
        items matches {
          ELEMENT[at0010] occurrences matches {1..1}{--Birth date
        value matches {DV_DATE matches {*}}}
```

```
?xml version="1.0" encoding="utf-8"?>
xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.openehr.org/v1
   targetNamespace="http://schemas.openehr.org/v1" elementFormDefault="qualified" version="v1
   id="person_detail.xsd">
<xs:element name="DemographicData">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="birthData" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="birthDate" type="xs:date" use="required"/>
            <xs:element name="dateOfBirthFollowUpIndicator" type="xs:boolean" use="optional"/>
            <xs:element name="countryOfBirth" type="xs:string" use="optional"/>
            <xs:element name="state" type="xs:string" use="optional"/>
            <xs:element name="city" type="xs:string" use="optional"/>
            <xs:element name="plurality" type="xs:string" use="optional"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

```
<<person_details>>
DemographicData
- birthDate : Date
- dateOf BirthFollowUpIndicator : boolean
- countryOfBirth : String
- state : String
- city : String
- plurarity : String
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DemographicData
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="person_details.xsd">
    <birthData>
        <birthDate>20/05/1945</birthDate>
        <countryOfBirth>Brazil</countryOfBirth>
        <state>SP</state>
        <city>Dracena</city>
    </birthData>
</DemographicData>
```

Due to the available XML technology (e.g., DOM and XPath), the output handling and the code generation have been straightforward to implement. The generated code, which represents an *open*EHR Extract, contains data about patients. These data can be exchanged using REST*ful* Web services as representational state amongst the components of a distributed HIS. The separation between the Message Generator and the legacy system allows the latter to be agnostic with respect to *open*EHR. This allow these systems to access the Message Generator by referring to the provided interfaces, avoiding in this way the explicit use of healthcare standards and also modifications of these systems.

Chapter 7

Multi-Agent System Design

Software agents have been proposed to deal with many different kinds of problems in the healthcare domain [209-214]. They can maintain the autonomy of the collaborating participants, integrate disparate operating environments, enable human-computer interfaces to adapt to medical information and users requirements, and coordinate distributed data, such as patients records held in different departments within the healthcare providers. The cooperation and coordination abilities of two or more agents can be combined through the use of well-defined communication rules for building a Multi-Agent System to cope with complex tasks in healthcare environment [215]. In this work, we advocate the use of agent technologies to automate part of the healthcare professionals' daily tasks giving support to the decision-making steps and to share clinical knowledge related to their patients.

This chapter is organized as follows: Section 7.1 explains how agents communicate and dynamic relationship between agents in our architecture. Section 7.2 presents the ontology modeling process in our work. Section 7.3 gives an overview and presents the agent modeling process that is supported to guide the domain engineer tasks. Section 7.4 presents the main agents implemented in this work.

7.1 Agent Technology

In the abstract communication model of FIPA, communication among agents occurs through the exchange of asynchronous messages that correspond to communicative acts, such as *request*, *inform, agree* and *confirm* [216]. In order for a pair of agents to understand each other, a basic requirement is that they speak the same language and talk about the same things. This is usually achieved by means of an ontology, namely an *interaction ontology*. In addition to being able to exchange messages, agents need to share a vocabulary of predicates, functions and constants with an ontology [217]. A number of ontology languages have been developed. An ontology for a given domain is a set of schemas that define the structure of the *predicates*, *agent actions* and *concepts* (basically their names and their slots) that are pertinent to that domain [218, 219].

7.1.1 Inter-agent Communication

The use of the FIPA platform also provides a standard agent communication language (FIPA-ACL). An ACL Message consists of an outer message structure, providing information, such as the sender and the receiver of the message, and a message content, expressed in some language understandable to both the sending and receiving agents. The FIPA ACL specifications define a message structure, standard communicative acts and interaction protocols. In addition, the language used for all inter-agent communication is FIPA SL language.

A FIPA ACL message contains a set of one or more message elements. Precisely which elements are needed for effective agent communication will vary according to the situation; the only element that is mandatory in all ACL messages is the *performative*, although it is expected that most ACL messages will also contain *sender*, *receiver* and *content* elements. The set of FIPA ACL message parameters is shown in *Table 7-4*:

Table 7-4 FIPA ACL Message Elements

Element	Description	Category of Elements
Performative	Denotes the type of the communicative act of the ACL message	Type of communicative acts
Sender	Denotes the identity of the sender of the message, i.e., the name of the agent of the communicative act.	Participant in communication
Receiver	Denotes the identity of the intended recipients of the message.	
Reply-to	This element indicates that subsequent messages in this conversation thread are to be directed to the agent named in the reply-to element, instead of to the agent named in the sender element.	
Content	Denotes the content of the message; equivalently denotes the object of the action.	Content of message

Language	Denotes the language in which the content element is expressed.	Description of Content
Encoding	Denotes the specific encoding of the content language expression.	
Ontology	Denotes the ontology(s) used to give a meaning to the symbols in the content expression.	
Protocol	Denotes the interaction protocol that the sending agent is employing with this ACL message.	Control of conversation
Conversation-ID	Introduces an expression (a conversation identifier) which is used to identify the ongoing sequence of communicative acts that together form a conversation.	
Reply-With	Introduces an expression that will be used by the responding agent to identify this message.	
In-Reply-To	Denotes an expression that references an earlier action to which this message is a reply.	
Reply-By	Denotes a time and/or date expression, which indicates the latest time by which the sending agent would like to have received a reply.	

The FIPA ACL specifies the following *Performatives* [220]*:*

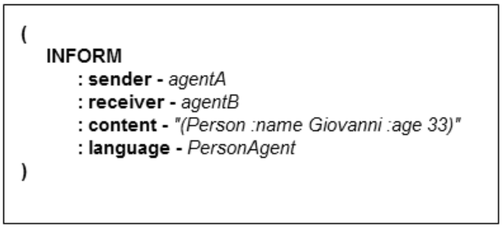
- *ACCEPT-PROPOSAL* - The action of accepting a previously submitted propose to perform an action;
- *AGREE* - The action of agreeing to perform a requested action made by another agent. Agent will carry it out;
- *CANCEL* - Agent wants to cancel a previous request;
- *CFP*- Agent issues a call for proposals. It contains the actions to be carried out and any other terms of the agreement;
- *CONFIRM* - The sender confirms to the receiver the truth of the content. The sender initially believed that the receiver was unsure about it;
- *DISCONFIRM* - The sender confirms to the receiver the falsity of the content;
- *FAILURE* - Tell the other agent that a previously requested action failed;

- *INFORM* - Tell another agent something. The sender must believe in the truth of the statement. Most used performative;
- *INFORM-IF* - Used as content of request to ask another agent to tell us is a statement is true or false;
- *INFORM-REF* - Like inform-if but asks for the value of the expression;
- *NOT-UNDERSTOOD* - Sent when the agent did not understand the message;
- *PROPAGATE* - Asks another agent so forward this same propagate message to others;
- *PROPOSE* - Used as a response to a cfp. Agent proposes a deal;
- *PROXY* - The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them;
- *QUERY-IF* - The action of asking another agent whether or not a given proposition is true;
- *QUERY-REF* - The action of asking another agent for the object referred to by an referential expression;
- *REFUSE* - The action of refusing to perform a given action, and explaining the reason for the refusal;
- *REJECT-PROPOSAL* - The action of rejecting a proposal to perform some action during a negotiation;
- *REQUEST* - The sender requests the receiver to perform some action. Usually to request the receiver to perform another communicative act;
- *REQUEST-WHEN* - The sender wants the receiver to perform some action when some given proposition becomes true;
- *REQUEST-WHENEVER* - The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again; and
- *SUBSCRIBE* - The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes.

Each information exchange between agents is wrapped inside an ACL message before being sent to the receiver, who performs the response with the additional overhead of checking semantics to determine whether the received message content can be interpreted, that is whether it is meaningful [221]. This procedure is quite challenging in open applications, where not all agents' messages are expected to carry content that is semantically correct and consistent. A meaningful content conforms to a set of predefined rules of an *ontology*. With a properly defined ontology, a programmer can leverage the power of automatic message content validation provided by a FIPA compliant agent framework, such as JADE and JADEX. They provide tools for generating code from ontology documents, thus reducing the development effort, and enabling the binding of the message contents to concepts defined in an ontology at runtime [222].

When an agent ***A*** communicates with another agent ***B***, a certain amount of information ***I*** is transferred from ***A*** to ***B*** in the form of an FIPA-ACL message. Inside the ACL message, ***I*** is represented as a content expression consistent with a proper content language (e.g. FIPA-SL) and encoded in a proper format (e.g. a string of characters) [223]. Both ***A*** and ***B*** have their own (possibly different) ways of internally representing ***I***. Taking into account that the way an agent internally represents a piece information must facilitate the handling of that piece of information, it is quite clear that the representation used in an FIPA-ACL content expression is not suitable for the inside of an agent [224]. For example, the information that '*there is a person whose name is Giovanni and who is 33 years old*' in an ACL content expression could be represented as the *string*: *(Person :name Giovanni :age 33)*, as shown in *Figure 7-38* ACL Message as String of characters.

Figure 7-38 ACL Message as String of characters



```
(
    INFORM
        : sender - agentA
        : receiver - agentB
        : content - "(Person :name Giovanni :age 33)"
        : language - PersonAgent
)
```

Storing this information inside an agent simply as a *string* variable is not suitable to handle the query to get the age of Giovanni, which would require the string to be parsed at each query. Considering software agents written in Java (as in the case of JADE agents), information can conveniently be represented inside an agent as Java objects. For example, representing the above information about '*Giovanni*' as an instance of a specific class *Person* initialized with name = 'Giovanni' and age = 33 would allow to handle it easily if the class is defined as a Java Bean as follows:

```
public class Person {
    private String name;
    private int age;
    public Person() {}
    public String getName() {return name;}
    public void setName(String name) {this.name = name;}
    public int getAge() {return age;}
    public void setAge(int age) {this.age = age;}
}
```

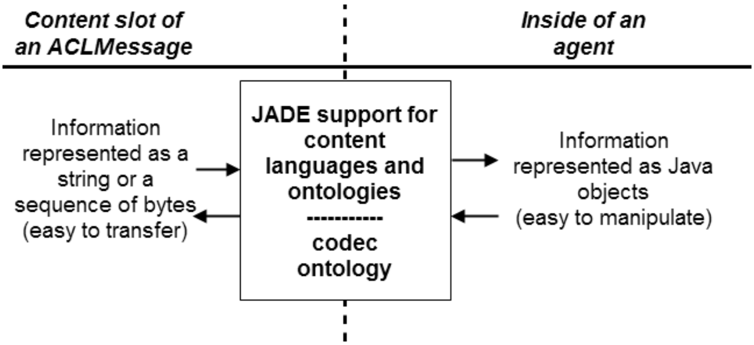
However, if on the one hand information handling inside an agent is facilitated by this class definition, on the other hand each time agent ***A*** sends a piece of information ***I*** to agent ***B***:

- ***A*** needs to convert its internal representation of ***I*** to the corresponding ACL content expression representation, and ***B*** needs to perform the opposite conversion, and
- ***B*** should also perform a number of semantic checks to verify that ***I*** is a meaningful piece of information, namely that it complies with the rules of the ontology by means of which both ***A*** and ***B*** ascribe a proper meaning to ***I***. For instance, ***B*** should verify whether the age of Giovanni is actually an integer value.

7.1.2 Agent Platform

In JADE, the conversion and validation operations are performed by a dedicated content manager object, thus allowing developers to manipulate information within their agents as Java objects without efforts [225], as shown in *Figure 7-39*. The content manager object class provides all the necessary methods for content conversion, but in reality delegates all the work done to special *ontology* and content *codec* object instances [226]. While the *codec* performs syntactic translations with direct support for two content languages, the *ontology* validates all information from a semantic perspective, which requires all elements of a content expression to be known and classified. With this information, the ontology serves as a dictionary or vocabulary for inter-agent communication.

Figure 7-39 Conversion Supported by JADE


Content slot of an ACLMessage
Inside of an agent
Information represented as a string or a sequence of bytes (easy to transfer)
JADE support for content languages and ontologies
----
codec
ontology
Information represented as Java objects (easy to manipulate)

While an ontology is typically specific to a given domain, content languages are domain independent. Therefore, unlike ontologies, which normally must be defined in an *ad hoc* way for the domain addressed by an agent application, a content language can be selected among those already available [227].

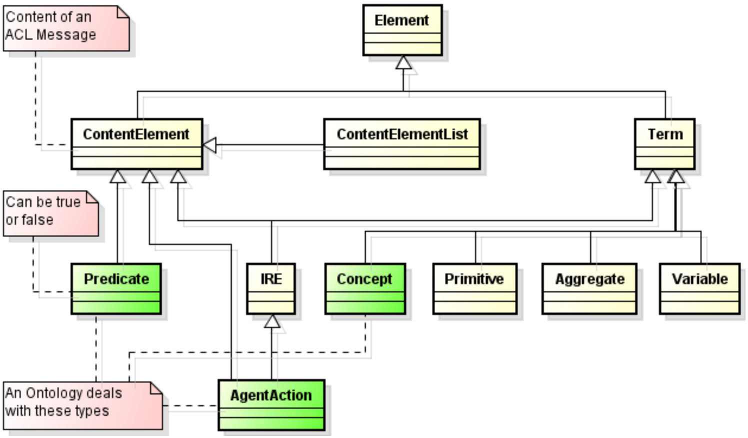
7.1.3 Agent-based Semantic Language

A Message content ontology allows agents to describe facts, beliefs, hypotheses and predications about a domain. Ontologies range in abstraction to general concepts to concepts that are restricted to some specific knowledge domain. Concepts at general levels are called upper ontology [217, 228], in which a general agreement exists:

- There are *objects* in the world;
- Objects have *properties* or *attributes* that can take *values*;
- Objects can have various different *relations* with each other;
- Properties and relations can change over *time*;
- *Events* occur at different *time instants*;
- *Processes* in which objects participate can be performed over time;
- The world and its objects can be in different *states*;
- Events can *cause* other events or states changes as *effects*; and
- Objects can have *parts*.

The different ways to represent the *objects, relations, states, events,* and *processes* does not say anything about which classes of these entities exist in some specific domain of interest. The FIPA-ACL specification dictates that all messages should have semantics that conforms to its *performative* (type of action taken). *Figure 7-40* shows the content reference model, which discerns between *predicates* (status describing expressions holding a truth value) and *terms* (expressions describing entities from the world where they exist and are the subject of agent discussions) [229].

Figure 7-40 Content Reference Model



In *Figure 7-40*, *Predicates* (or *facts*) are boolean expressions that say something about the status of the world, such as *(Has (Person :name John)*

(BloodType :type A :factorRH +)) stating that '*the person John has the blood type A Rh factor positive*'. *Predicates* can be meaningfully used, for instance, as the sole content of an INFORM or QUERY-IF message, while would make no sense if used as the content of a REQUEST message.

Terms (or *entities*) can be further categorized in six types, as shown in *Figure 7-40*:

- *Concept*: entities with a complex structure that can be defined in terms of slots, such as *(Person :name John :age 33)*. Concepts typically make no sense if used directly as the content of an ACL message. In general, they are referenced inside predicates and other concepts, such as *(Physician :specialty Cardiologist :name (Person :name Peter))*;
- *AgentAction*: special concept that indicates an action that can be performed by some agent, such as *(Implant-Pacemaker (Physician :specialty Cardiologist) (Person :name John))*. It is useful to treat agent actions separately, since they are meaningful contents of certain types of ACL messages, such as a REQUEST;
- *Primitive*: expressions that indicate atomic information types such as String and Integers;
- *Aggregate*: groups of other entities, such as *(sequence (Person :name John) (Person :name Peter))*;
- *IRE (Identifying Reference Expressions)*: expressions using a predicate as the identification criterion, such as *(all ?x (Has ?x (BloodType :type A factorRH +)))*, identifying 'all the elements X for which the predicate *'(Has ?x (BloodType :type A factorRH +))'* is true, i.e., all the persons who have A positive blood type. These expressions are typically used in queries;
- *Variable*: expressions that indicate a generic element not known beforehand and they are typically used in queries.

7.1.4 Agent-based Modeling

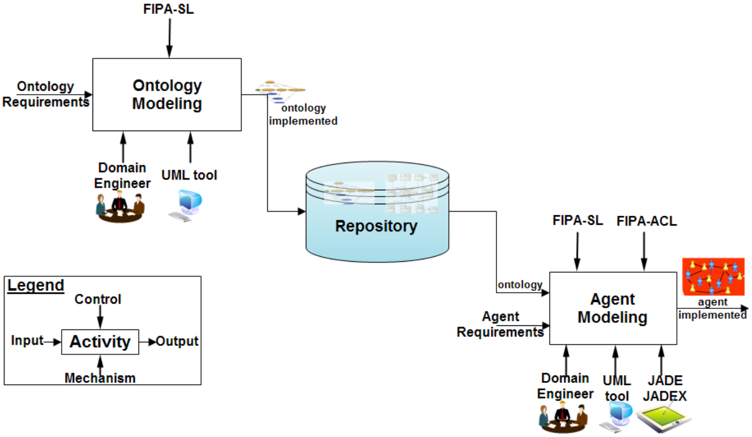
Traditional modeling approaches treat employees, projects, products, customers, and partners of a company either as averaged quantities or as passive entities or resources of a process. The process-centric approach focuses on the processes themselves, rather than on people, documents, contents or other components [230]. Process-centric approach can capture organizational dynamics, but they ignore that all those people, products and projects are different and have their own histories, intentions, desires, individual properties and complex relationships. Agent-based approaches address these limitations as they suggest that the modeler directly focus on individual objects in and around the organization, their individual behaviors and their interactions [231]. The agent-based model is actually a set of interacting active objects that reflect objects and relationships in the real world and thus is a natural step to understand and manage the complexity of today's business and social systems [232].

The specification of agent behaviors consists of two elements [233]: (1) a proper representation of *agent actions,* which are the elements at the basis of agent behavior, because they can cause modifications in their environment or in other agents that constitute the agent-based modeling; (2) the mechanisms that effectively select the actions to be carried out, according to the perceptions and internal state of the agent.

Different approaches have been proposed in order to describe specific agent behaviors, and they are generally classified into *deliberative* and *reactive* [234]. *Reactive* agents perform their actions as a consequence of the perception of stimuli coming either from other agents or from the environment. Generally, the behavioral specification of this kind of agent is a set of *condition-action* rules, with the addition of a selection strategy for choosing an action to be carried out whenever more rules could be activated. *Cognitive* models, defined within the planning context, provide a symbolic and explicit representation of the world within agents, and their decisions are based on logic reasoning and symbol manipulation. In our architecture, we applied a hybrid approach for agent modeling, in which are combined *reactive* and *cognitive* agents.

In this work, we followed the *Knowledge Engineering* approach [235] as a systematic process to design knowledge-based systems, which are structured according to our understanding of how human reasoning and logic works. Using the *Structured Analysis and Design Technique (SADT)* diagram notation, *Figure 7-41* shows an overview of the agent modeling process, which consists of the following phases: *Ontology Modeling (OM) and Agent Modeling (AM).* In OM, the domain ontology models are defined based on semantic requirements needed for the message exchange between the agents, ensuring that the vocabulary is well understood by the agents. These models are used to support agent modeling, by generating the code that handles the message exchange between the agents in order to support their interoperability and cooperation. In AM, agents are developed by reusing the artifacts obtained during the OM phase.

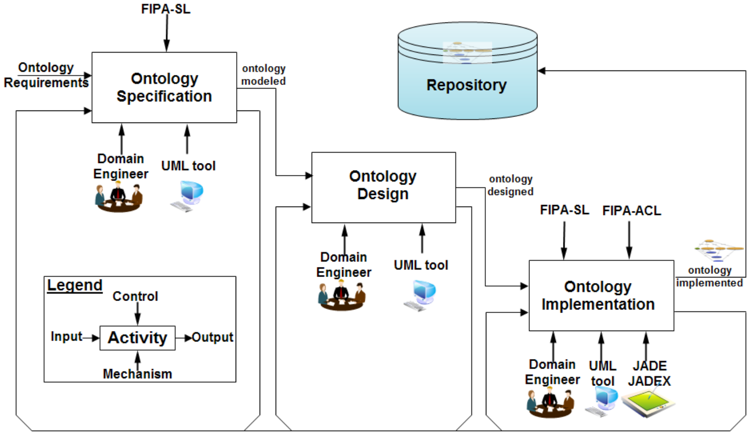
Figure 7-41 Agent-based Modeling



7.2 Ontology Modeling

Figure 7-42 shows an overview of the ontology modeling process, which consists of the following activities: *Ontology Specification* (OS), *Ontology Design* (OD), and *Ontology Implementation* (OI). The role of the ontologies in the agent-based modeling process is to facilitate the construction of a domain model and the message exchange between the agents by providing a vocabulary of terms and relations of the domain.

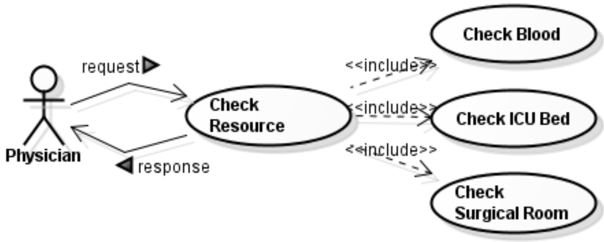
Figure 7-42 Ontology Modeling Overview



Ontology Specification

In the OS activity, the domain ontology requirements are elicited, analyzed and represented in models that express the knowledge about the domain. The OS main actor is the *Domain Engineer*, who focuses on the functionality of the system during the requirements specification and early analysis phase. The *Domain Engineer* employs use case models to capture the functional requirements related to the domain ontology by focusing on usage situations. The *Domain Engineer* analyzes the domain requirements with the help of UML tool and guided by the FIPA-SL specifications in order to understand the goals, limitations, and requirements of the domain, and then describes the use cases by identifying a set of terms and their relations. The outputs of the OS activity are the use case diagrams, in which the *Domain Engineer* specifies new schemas in order to identify the types of *concepts*, *agent actions* and *predicates* that are pertinent to the addressed domain. *Figure 7-43* shows an excerpt of the diagram that provides a high-level view of the use case in which the actor physician requests for resources in a healthcare environment.

Figure 7-43 Ontology Specification Model – Request for Resources



Ontology Design

In the OD activity, the deliverables of the OS activity are used for defining the domain terminology, in which the *Domain Engineer* focuses on *concepts, agent actions, predicates* and their *relationships* in order to design the domain ontologies. The *Domain Engineer* uses the UML tools to define the classes and their relationships for all types of *predicate*, *agent action* and *concept* in the domain. We adopted the following conventions to define the domain terminology: 'concepts' are denoted with substantives, such as *patient, physician, blood type* and *resource*; 'actions' are denoted with verbal phrases, such as *MakeAppointment, RequestResource, RequestForMeeting, BloodMatch* and *RequestMedicalReport*; and 'predicates' are denoted with Boolean assertions that are evaluated as true or false, such as *hasBloodType, hasICUBed* and *hasOperatingRoom*. *Figure 7-44* shows the ontology class diagram, which describes the structure of the concept *BloodType*, action *BloodMatch* and predicate *HasBloodType*. These are used in the content of messages exchanged by agents in this domain. We adorned the classes with stereotypes <<*concept*>>, <<*action*>> and <<*predicate*>> to indicate that they have additional semantic beyond a UML class.

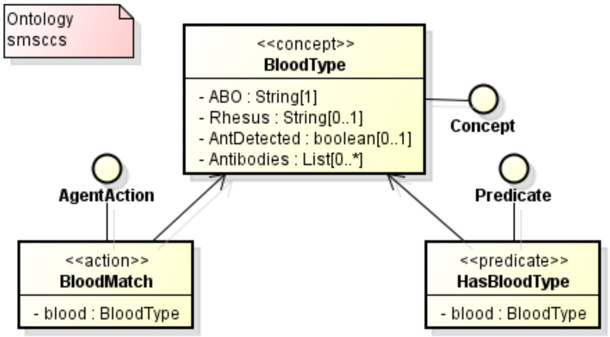
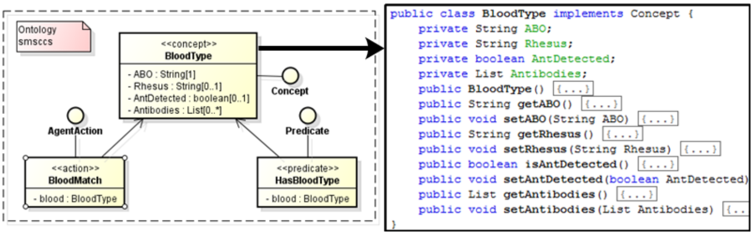


Figure 7-44 Class Diagram representing a simple Ontology for *Requesting Blood Type*

Ontology Implementation

In the OI activity, deliverables of the OD activity are used to create the Java code skeletons. The *Domain Engineer* uses the IDE and JADE/JADEX tools for implementing the classes and their relationships, which are the OI outputs. *Figure 7-45* shows the Java code produced in this activity, such as the *BloodType* (concept), *BloodMatch* (action) and *HasBloodType* (predicate) classes, which are implemented according to the ontology class diagram for *Requesting Blood Type* designed in the OD activity. Since the ontology schemas are based on FIPA-SL, the models created using an ontology model can be considered as a refinement performed to define the messages to be exchanged between the agents.



```
public class BloodType implements Concept {
    private String ABO;
    private String Rhesus;
    private boolean AntDetected;
    private List Antibodies;
    public BloodType() {...}
    public String getABO() {...}
    public void setABO(String ABO) {...}
    public String getRhesus() {...}
    public void setRhesus(String Rhesus) {...}
    public boolean isAntDetected() {...}
    public void setAntDetected(boolean AntDetected)
    public List getAntibodies() {...}
    public void setAntibodies(List Antibodies) {..
}
```

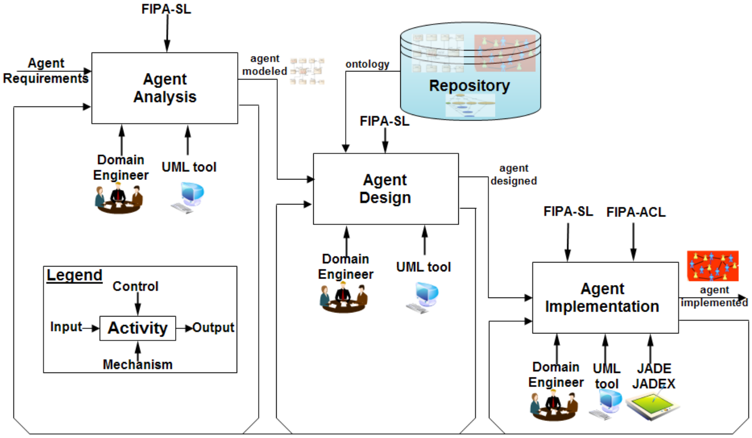
Figure 7-45 Excerpt of Java Code for Ontology *Requesting Blood Type*

The main goal of the *Ontology Modeling* phase is to extract the knowledge necessary for instantiating a particular agent in the *Agent Modeling* phase from the generic models. This knowledge corresponds to the ontology schemas. We defined a set of ontologies that allow actors of the domain to communicate with each other in a language they understand.

7.3 Agent Modeling

Using the Structured Analysis and Design Technique (SADT) diagram notation, *Figure 7-46* shows an overview of the agent-based modeling process, which consists of the following steps: *Agent Analysis* (AA), *Agent Design* (AD) and *Agent Implementation* (AI). In our research, we considered our multi-agents system as a set of interrelated agents that communicate with each other in order to fulfill the agents' goals, using the ontologies implemented in the *Ontology Modeling* phase for ensuring semantic interoperability between agents.

Figure 7-46 Agent Modeling Overview



Agent Analysis

In the *AA* activity, the *Domain Engineer* analyzes the legacy system in order to understand its goals, limitations and requirements, and then describes the activities and use cases of the system, by identifying the abstract models, which represent the agent types. The *Domain Engineer* can model these agents with UML tool. The outputs of the AA activity are the activity diagrams for the agents' behavior specification. *Figure 7-47* shows an excerpt of the activity diagram that emphasizes the sequence and conditions for coordinating agent behaviors for requesting resources.

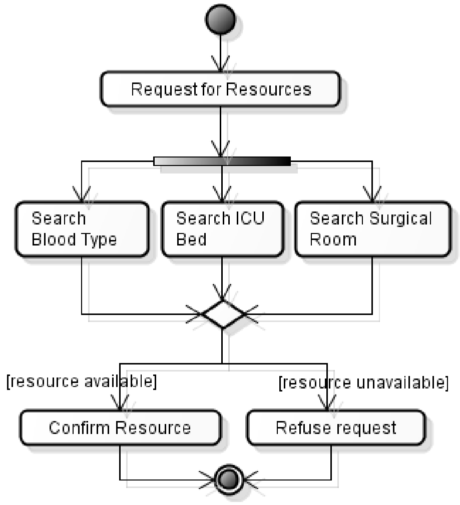


Figure 7-47 Agent Activity Diagram - *Request for Resources*

Agent Design

In the AD activity, the agent types and their roles are identified, especially the agent behavior. For example, an agent of type *PhysicianAgent* can play the roles of surgeon or anesthesiologist. The agent interactions are defined to determine how, what and when the various agents communicate. The agents are designed by reusing the ontologies implemented in the *Ontology Modeling* phase for ensuring semantic interoperability between agents. The AD outputs are the sequence diagrams that model the interactions between agents, and the class diagrams that represent the internal structure, behavior and relationship among the agent types.

Figure 7-48 shows the sequence diagram for requesting resources that focuses on the negotiation between the agents. For instance, a message is annotated with the *cfp* (call for proposal) FIPA-performative to indicate that it should be multicast from an Initiator (*ResourceAgent*) to *n* Participants (*IntensiveUnitAgent, SurgicalCenterAgent* and *BloodBankAgent*). In this example, the *BloodBankAgent* participant sends a refusal message to this call for proposal, while *SurgicalCenterAgent* and *IntensiveUnitAgent* send proposals. The *proposal* can be accepted, which has been the case for *IntensiveUnitAgent* or rejected, which has been the case for *SurgicalCenterAgent*.

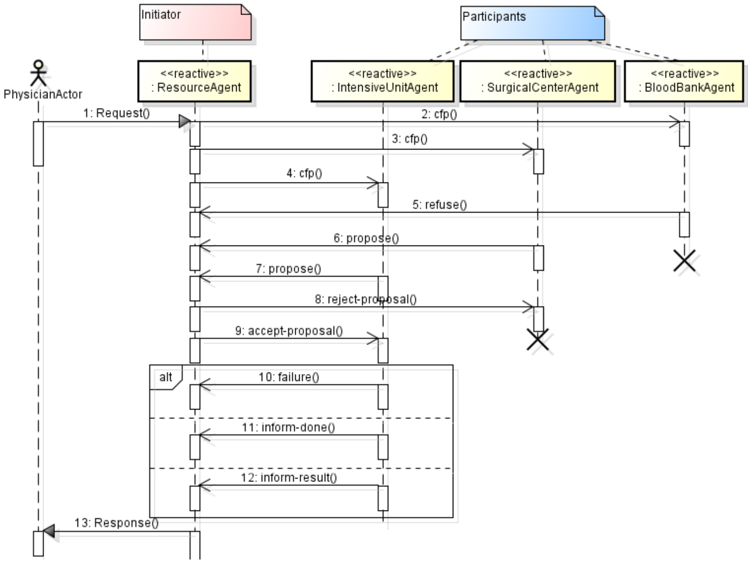


Figure 7-48 Sequence Diagram for *Requesting Resources*

Figure 7-49 shows the class diagram for the classes involved in requesting resources. *PatientAgent* is responsible for the continuous monitoring of the evolution of a patient, and can send and receive messages to and from a *PhysicianAgent*. *PhysicianAgent* is a mobile agent endowed with intentionality that helps the medical staff monitor the tasks performed during a workday, and obtain information about patients and the availability of resources without requiring the intervention of healthcare professionals. For example, information related to bedridden patients is obtained via *ResourceAgent*. *ResourceAgent* is a static agent that runs on a remote server and is responsible for mediating the access to resources related to *HISAgent*. *ResourceCTR* provides interfaces to *PatientAgent* and *PhysicianAgent*, which are the main agents that communicate with *ResourceAgent* for obtaining clinical information. *PhysycianBehaviour* and *ResourceBehaviour* represent the behaviors executed by the agents *PhysicianAgent* and *ResourceAgent*, respectively. These behaviors include checking of received ACL message performatives (i.e., *inform*, *request* and *cfp*), requesting for information (i.e., *requestResource*, *requestForMeeting* and *RequestMedicalReport*) and sending reply ACL messages to the requestor agents (i.e., *prepareResponse* and *prepareResultNotification*).

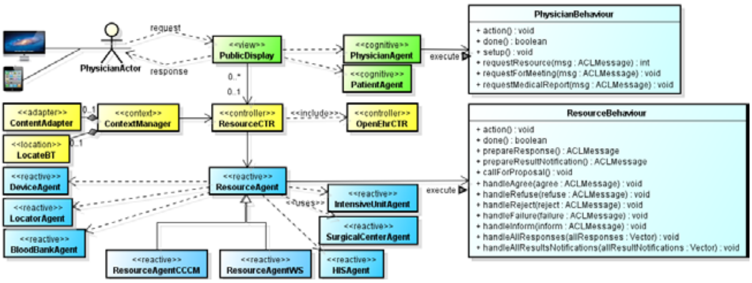
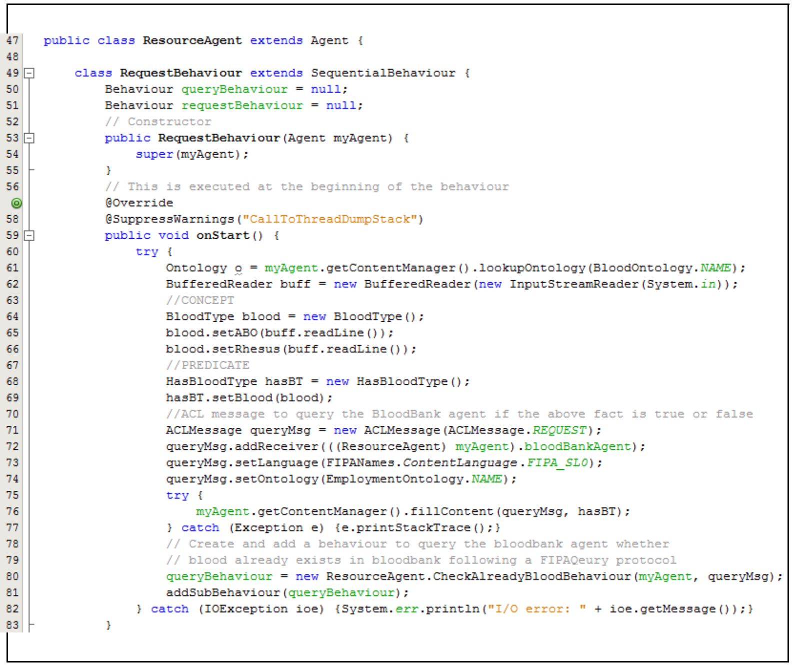


Figure 7-49 Class Diagram – Agents and Behaviors

Agent Implementation

In the AI activity, deliverables of the AD activity are used to identify and implement the software agents that are assigned to the actors, so that their tasks and behaviors are defined according to their roles in the domain. The *Domain Engineer* uses the IDE for implementing the software agents, which are the AI outputs. We applied JADE for the agents' implementation, which facilitates the development of a multi-agent system and provides a standard implementation of the FIPA-ACL communication language. The agent and behavior classes that were refined and designed on AD activity are implemented, and the agents communicate through JADE. *Figure 7-50* shows the *ResourceAgent* behavior implemented in Java according to the *Requesting Resource* sequence diagram obtained in the AD activity, using the ontology for *Requesting Blood Type* obtained in the OM phase. The *BloodType* concept (Line 63-66) and *HasBloodType* predicate (Line 67-69) are instantiated, and the request *ACLMessage* (Line 71-77) is instantiated to query the *BloodBankAgent* for the availability of blood of the requested blood type (Line 80-81).



```java
public class ResourceAgent extends Agent {

    class RequestBehaviour extends SequentialBehaviour {
        Behaviour queryBehaviour = null;
        Behaviour requestBehaviour = null;
        // Constructor
        public RequestBehaviour(Agent myAgent) {
            super(myAgent);
        }
        // This is executed at the beginning of the behaviour
        @Override
        @SuppressWarnings("CallToThreadDumpStack")
        public void onStart() {
            try {
                Ontology o = myAgent.getContentManager().lookupOntology(BloodOntology.NAME);
                BufferedReader buff = new BufferedReader(new InputStreamReader(System.in));
                //CONCEPT
                BloodType blood = new BloodType();
                blood.setABO(buff.readLine());
                blood.setRhesus(buff.readLine());
                //PREDICATE
                HasBloodType hasBT = new HasBloodType();
                hasBT.setBlood(blood);
                //ACL message to query the BloodBank agent if the above fact is true or false
                ACLMessage queryMsg = new ACLMessage(ACLMessage.REQUEST);
                queryMsg.addReceiver(((ResourceAgent) myAgent).bloodBankAgent);
                queryMsg.setLanguage(FIPANames.ContentLanguage.FIPA_SL0);
                queryMsg.setOntology(EmploymentOntology.NAME);
                try {
                    myAgent.getContentManager().fillContent(queryMsg, hasBT);
                } catch (Exception e) {e.printStackTrace();}
                // Create and add a behaviour to query the bloodbank agent whether
                // blood already exists in bloodbank following a FIPAQeury protocol
                queryBehaviour = new ResourceAgent.CheckAlreadyBloodBehaviour(myAgent, queryMsg);
                addSubBehaviour(queryBehaviour);
            } catch (IOException ioe) {System.err.println("I/O error: " + ioe.getMessage());}
        }
```

Figure 7-50
ResourceAgent Behavior – Request for Resources

7.4 *Agent* Package

In our architecture presented on Chapter 4, the *agent* package contains static and mobile agents, each one endowed with specialized capabilities and goals for helping the healthcare professionals check the availability of resources, and obtain information and detect abnormalities on their patients [236].

7.4.1 Main Components

The static agents provide resources to the mobile agents, allowing them to move in the architectural environment in order to achieve their goals and to communicate with other agents through an asynchronous communication channel. *Figure 7-51* depicts part of the component diagram of the *agent* package, showing their dependencies. In *Figure 7-51* the agents endowed with intentionality are decorated with the cognitive stereotype, while the agents that only display reactive behavior are decorated with the reactive stereotype. The agents were modeled using our *Agent Modeling* approach centered on the intentional characteristics of agents.

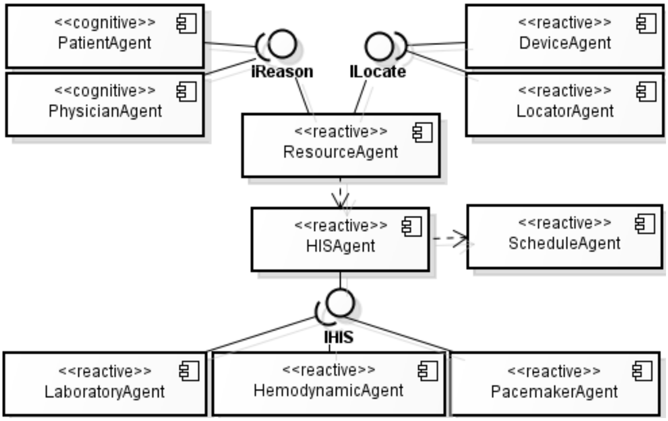


Figure 7-51
Components of *agent* Package

HISAgent is a static agent, which has the ability of analyzing several EHRs and retrieving information from them, which have been requested by a *ResourceAgent*. The main task of *HISAgent* is to ensure the interoperability between various healthcare information systems through the message exchange between agents. These messages contain an EHR extract, which is defined based on the constraints imposed by the archetype. This EHR extract is structured according to the *open*EHR RM. *Figure 7-51* shows the *IHIS* interface, which is provided by *HISAgent* and required by *LaboratoryAgent*, *HemodynamicAgent* and *PacemakerAgent*. These components define models for information retrieval from an EHR.

ResourceAgent is a static agent that runs on a remote server and is responsible for mediating the access to resources related to *HISAgent*. *ResourceAgent* provides the interface *IReason* to *PatientAgent* and *PhysicianAgent*, which are the main agents that communicate with *ResourceAgent* for obtaining information related to the EHR.

PhysicianAgent is a mobile agent endowed with intentionality that helps the medical staff monitor the tasks performed during a workday, and obtain information about patients and the availability of resources without requiring the intervention of healthcare professionals. For example, information related to bedridden patients is obtained via *ResourceAgent*. Any medical staff member can use her mobile device to trigger the *PhysicianAgent*, which is responsible for achieving the goal established in accordance with the plans, allowing the medical staff to deal with any emergency. A *PhysicianAgent* can be dispatched by the network from its origin container to another container, and when it migrates it keeps its intentionality according to its beliefs for achieving its goals. After achieving its goals, the *PhysicianAgent* returns to its origin bringing a message that can be a string value, or a serialized Java object containing an EHR extract related to an *open*EHR archetype, or an *Ontology* object containing the description of the concepts used by the agents (e.g., in FIPA-SL).

PatientAgent is a static intelligent agent, since it has beliefs, desires and intentions and is capable of applying plans to pursue its intentions in the environment where it resides. A *PatientAgent* is responsible for the continuous monitoring of the evolution of a patient, and can send and receive messages to and from a *PhysicianAgent*.

7.4.2 Ontology Example

Bluetooth Access Points (BAPs) were employed in the hospital and clinics in order to handle the location of people and devices. *DeviceAgent* and *LocatorAgent* are responsible for determining the patients' and healthcare professionals' locations. Based on the information provided by these agents, a *PhysicianAgent* can move through the platform in pursuit of its goals.

Figure 7-52 shows an ontology example used for device location that enables *DeviceAgent* and *LocatorAgent* to identify the owner of a given device. Several types of ontology schemes can be defined by applying our *Ontology Modeling* approach using the stereotypes *concept, predicate* and *action*. For example, *Device* and *Person* are concepts, the predicate *Owns* indicates that a device has a given owner, and the action *Locate* indicates that an agent must locate the owner of a device. *Predicates* and *actions* are exchanged in the messages instead of concepts. *Concepts* are assumed to correspond to unary predicates, i.e., they consist of simple sets of elements.

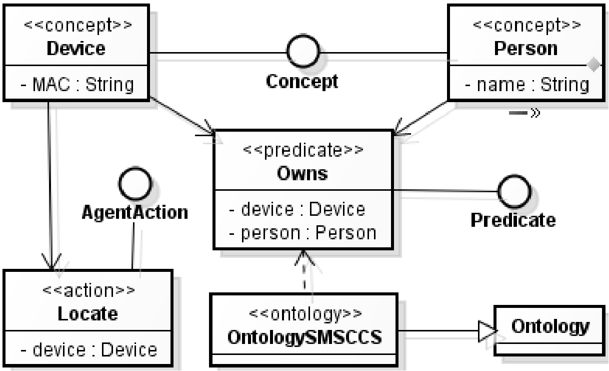
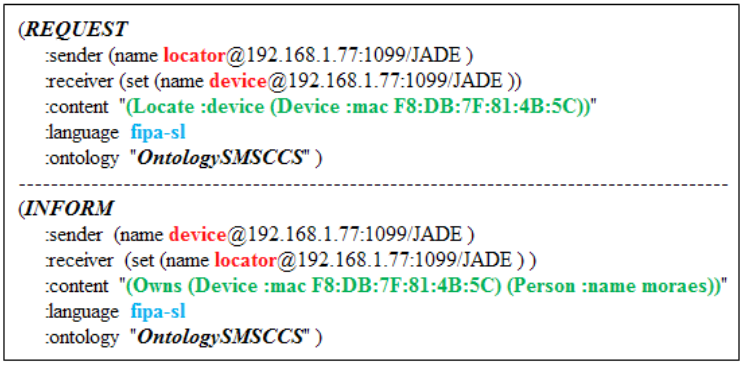


Figure 7-52 Example of the Ontology for Device Location

Agents need to communicate in order to achieve their goals effectively. Communication extends the agents' perception, by allowing them to benefit from the information and expertise of other agents. *DeviceAgent* and

LocatorAgent communicate by exchanging messages written in a language that supports the semantics of the message content within the specified domain. Therefore, these agents are able to understand the messages they exchange because they share the necessary domain knowledge (ontology) and the language they use has a well-defined semantics, in accordance with this knowledge.

Figure 7-53 shows two FIPA-ACL messages indicated by the performative *REQUEST* and *INFORM*. In *REQUEST* message, the *locator agent* has sent a request message to the *device agent*. The message is written in FIPA-SL (Semantic Language) and its content is related to *OntologySMSCCS*. The message indicates that the MAC address F8:DB:7F:81:4B:5C of a device was located by the access point. In *INFORM* message, the device agent has sent an informative message to the locator agent. The message indicates that the owner of the device with MAC address F8:DB:7F:81:4B:5C is the person called 'moraes'.



```
(REQUEST
   :sender (name locator@192.168.1.77:1099/JADE )
   :receiver (set (name device@192.168.1.77:1099/JADE ))
   :content "(Locate :device (Device :mac F8:DB:7F:81:4B:5C))"
   :language fipa-sl
   :ontology "OntologySMSCCS" )
----------------------------------------------------------------------------------
(INFORM
   :sender (name device@192.168.1.77:1099/JADE )
   :receiver (set (name locator@192.168.1.77:1099/JADE ) )
   :content "(Owns (Device :mac F8:DB:7F:81:4B:5C) (Person :name moraes))"
   :language fipa-sl
   :ontology "OntologySMSCCS" )
```

Figure 7-53 ACL Messages Example

Each agent implemented in our architecture has specialized capabilities and goals in order to perform tasks for the benefit of pervasive healthcare. The agents allow the healthcare professionals to detect abnormalities in their patients, to check the availability of resources within the healthcare environment, and to obtain information about patients. This allows healthcare professionals and patients to use any device, anywhere and at anytime.

Chapter 8

Case Studies and Evaluation

This chapter describes the validation field of our reusable architecture and message generator and its architectural components discussed throughout Chapters 4 to 7, by means of case studies where the architecture has been instantiated and tested.

This chapter is further structured as follow: Section 8.1 presents the participants of the usage scenarios. Section 8.2 gives an overview of the usage scenarios: *Deliver of Laboratory Analysis Results, Pacemaker Evaluation,* and *Medical Staff Meeting for Cardiac Surgery*. Section 8.3 discusses the results of the case studies, evaluating the applications developed for these case studies based on our reusable architecture.

8.1 Settings and Participants

We have conducted a case study at three cardiology clinics, one analysis laboratory, and the cardiology sector of the Santa Casa hospital, all of them located in the city of Marília (São Paulo, Brazil), where we defined the following usage scenarios with the help of ICT and healthcare professionals for capturing the requirements and applying our results: *Delivery of Laboratory Result, Pacemaker Evaluation,* and *Medical Staff Meeting*. These three scenarios are distinct in some aspects, such as end-users, deployment settings and goals. These scenarios have some similarities as they require the exchange clinical information in a distributed healthcare environment within the Cardiology Healthcare Network.

We started by applying observation methods [237], such as direct observation and work methods analysis, in order to determine the interactions, the participants and their roles on these interactions, to understand how healthcare professionals communicate with each other, and to compute the time they normally spent on the activities of these scenarios. Based on these observations, we instantiated the components of our reusable architecture, in order to design three communication systems for the defined usage scenarios. The acceptance of these integrated technologies in these communication systems was evaluated by two groups of participants, involving a limited but relevant set of people: *healthcare professionals* (physicians, medical students and nurses) and *patients*. The users have used the applications in their daily activities from 1st January to 30th July 2012, and after that they were asked to fill in a structured questionnaire. We decided to group all healthcare professionals together because they work as a team in these scenarios. We assigned fictitious names to them in this thesis to make sure the participants remain anonymous. *Table 8-5* depicts the number of participants for each scenario during the evaluation period.

Table 8-5 Scenarios participants

Participants Scenarios	Physicians	Medical Students	Nurses	Patients	Total
Scenario 1: Delivery of Laboratory Result	3 cardiology 2 laboratory	4 cardiology 3 laboratory	3 cardiology 3 laboratory	235	253
Scenario 2: Pacemaker Evaluation	3 cardiology	2 cardiology	2 cardiology	95	102
Scenario 3: Medical Staff Meeting	6 cardiology 3 hemodynamic 13 clinical 4 intensivist	8 cardiology 3 hemodynamic 5 clinical 2 intensivist	5 cardiology 2 blood bank 4 intensivist 2 surgical center	122	179
Total of participants	34	27	21	452	534

To cope with ethical issues, this research was submitted to and approved by the Medical Research Ethics Committee of the Federal University of São Paulo (UNIFESP). Before agreeing to voluntarily participate in the study, all participants received written and oral information about the study. Anonymity and confidentiality were guaranteed to all participants.

8.2 Usage Scenarios

The usage scenarios we have considered in our case study are in the Cardiology Domain due to its relevance and its suitability to the objectives of our message exchange support in pervasive healthcare environments. In our scenarios, we reused the hardware and software platforms then available in the Cardiology Healthcare Network in the city of Marília (São Paulo, Brazil).

8.2.1 Delivery of Laboratory Analysis Results

Scenario Description

Life Institute (LI) is a clinical analysis laboratory in the Marília area that has a unit in the Santa Casa hospital. LI receives and processes daily several requests for clinical analysis, which are stored with their results in the LI Laboratory Information System (LIS-LI) database. Results of requests from Santa Casa should be directly sent to the medical staff of this hospital.

Stakeholders

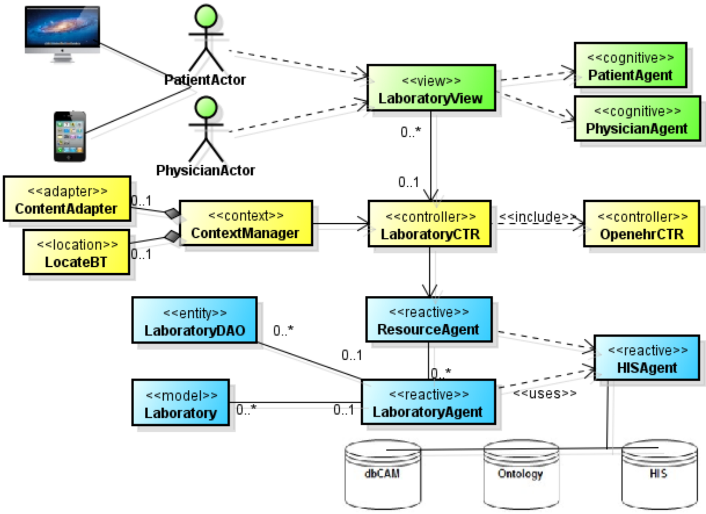
In this scenario, Dr. Ray, responsible for this LI unit, has finished the analysis requested by Dr. Call, a cardiac surgeon from the Cardiology Center of Marília (CCCM) that is located in Santa Casa, for his patient Mr. Silva.

Solution

Figure 8-54 shows the class diagram with the architecture components for this scenario, where *PhysicianAgent, LaboratoryAgent, PatientAgent,* and *HISAgent* represent Dr. Call, Dr. Ray, Mr. Silva and LIS-LI respectively.

The following main interactions between the agents occur in this scenario: (1) once the analysis result is available, *PatientAgent* receives a notification about this result in his mobile device; (2) *HISAgent* retrieves the information from the LIS-LI, and delivers it to *LaboratoryAgent*, which in turn sends a message containing this result to *ResourceAgent*; (3) using contextual information, *LaboratoryCTR* modifies this message, by adapting its content to the capabilities of the requesting device, and by serializing the *open*EHR Extract containing this result; and (4) this modified message, which is represented in ACL, is enveloped by *ResourceAgent* and delivered to the *PhysicianAgent*, who finally receives the analysis result in his mobile device.

Figure 8-54 Delivery of Laboratory Results



8.2.2 Pacemaker Evaluation

Scenario Description

A pacemaker is a medical device to regulate the heartbeat, and people carrying pacemakers should be checked at regular intervals. The Cardiology Clinic of Marília (CRTB) provides on-going follow-up care for patients with pacemakers, and has a HIS (CRTBSys) to keep track of the care provided to its patients. Pacemaker evaluation is often scheduled via the phone. During the evaluation, the physician usually spends a lot of time searching for clinical information on legacy systems related to a patient's pacemaker implantation.

Stakeholders

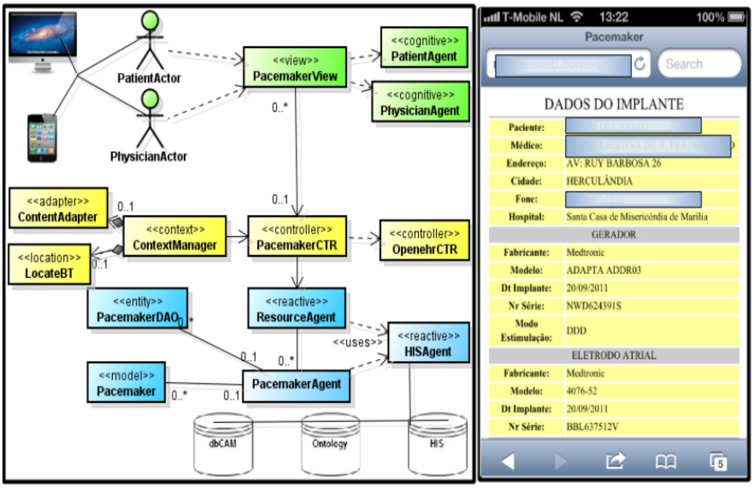
In this scenario, Dr. Call has implanted a pacemaker on patient Mr. Martins, who must have a pacemaker evaluation.

Solution

Figure 8-55 (a) shows the class diagram with the architecture components for this scenario, where *PhysicianAgent, PatientAgent,* and *HISAgent* represent Dr. Call, Mr. Martins, and CRTBSys respectively. *Figure 8-55* (b) shows the user interface of Dr. Call's device with information related to Mr Martins's pacemaker implantation.

The following main interactions between the agents occur in this scenario: (1) *PatientAgent* receives a notification in his mobile device setting the evaluation appointment; (2) once information about Mr. Martins's pacemaker implantation is required, *HISAgent* retrieves it from the CRTBSys, and delivers it to *PacemakerAgent*, which in turn sends a message containing this information to *ResourceAgent*; (3) using contextual information, *PacemakerCTR* modifies this message by adapting its content to the capabilities of the requesting device, and by serializing the *open*EHR Extract containing this information; and (4) this modified message, which is represented in ACL, is enveloped by *ResourceAgent* and delivered to the *PhysicianAgent* that finally receives the information about Mr. Martins's pacemaker implantation in his device.

Figure 8-55 Pacemaker Evaluation



(a) Class Diagram

DADOS DO IMPLANTE

Paciente:	[redacted]
Médico:	[redacted]
Endereço:	AV: RUY BARBOSA 26
Cidade:	HERCULÂNDIA
Fone:	[redacted]
Hospital:	Santa Casa de Misericórdia de Marília
GERADOR	
Fabricante:	Medtronic
Modelo:	ADAPTA ADDR03
Dt Implante:	20/09/2011
Nr Série:	NWD624391S
Modo Estimulação:	DDD
ELETRODO ATRIAL	
Fabricante:	Medtronic
Modelo:	4076-52
Dt Implante:	20/09/2011
Nr Série:	BBL637512V

(b) User Interface

8.2.3 Medical Staff Meeting for Cardiac Surgery

Scenario Description

Cardiac surgery is one of the best examples of teamwork in healthcare, and is performed by highly trained staff: a cardiovascular surgeon, who leads the team; an assistant surgeon, who follows the instructions of the cardiovascular surgeon; a cardiovascular anesthesiologist, who administers the drugs to keep patients asleep during surgery; a perfusionist, who operates the cardiopulmonary bypass machine, and cardiovascular nurses, who are specially trained to assist during cardiac surgery. Before going to a cardiac surgery of a certain patient, the team must have a meeting in order to prepare this surgery.

Stakeholders

Dr. Call is a cardiac surgeon and Dr. Day is an assisting surgeon, both from CCCM; Dr. John is an anesthesiologist from Santa Casa; Dr. Marden and Dr. Peter are physicians from the Hemodynamic Department of Marília; Mrs. Aline is a perfusionist; and Mrs. Elienne is a nurse. This Heart Team is planning to perform a cardiac surgery on patient Mr. Silva.

Solution

Figure 8-56 shows the class diagram with the architecture components for this scenario, where Caregiver encompasses all stakeholders involved in the cardiac surgery.

Figure 8-56 Staff Meeting for Cardiac Surgery

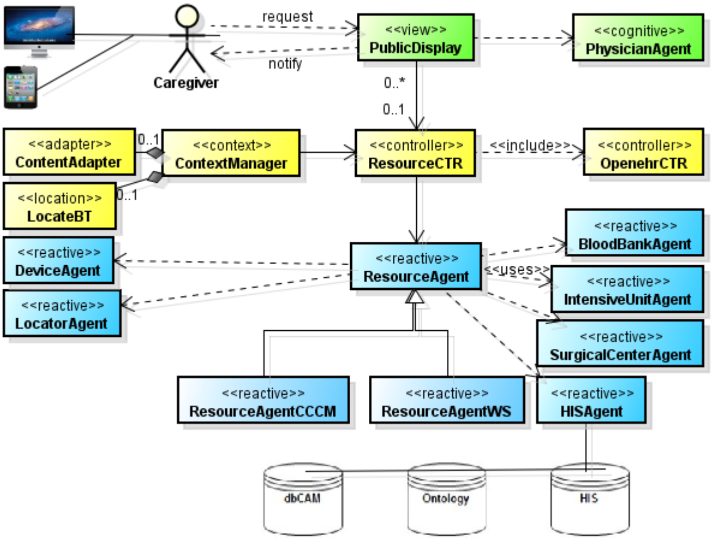
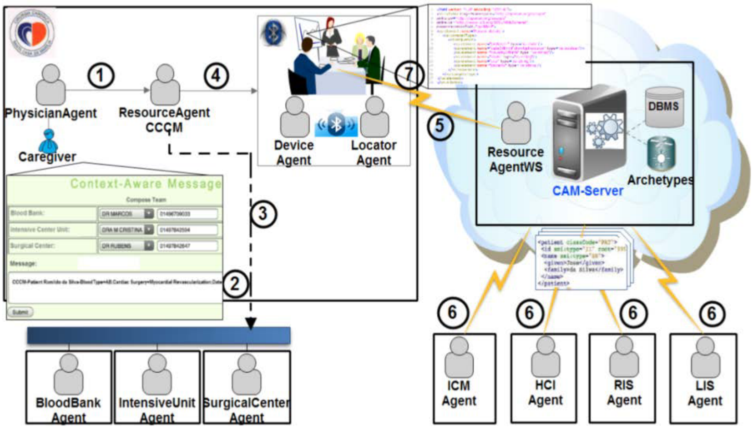


Figure 8-57 shows the following main interactions between the agents that are deployed in this scenario: (1) *PhysicianAgent* requests the resources for the surgery to the *ResourceAgentCCCM*; (2) *ResourceAgentCCCM* requests the correct blood type to the *BloodBankAgent*, a bed to the *IntensiveUnitAgent*, and a room to the *SurgicalCenterAgent*; (3) *ResourceAgentCCCM* checks constantly the availability of the requested resources; (4) once these resources are available, *ResourceAgentCCCM* notifies each staff member to set a date for the meeting, by sending a message to their mobile devices; (5) in the context-aware meeting room, the Heart Team members are located by *DeviceAgent* and *LocatorAgent*, and as soon as the surgeon registers the patient's name, the *ResourceAgentCCCM* requests information related to the patient's EHR to the *ResourceAgentWS*; (6) *ResourceAgentWS* receives the

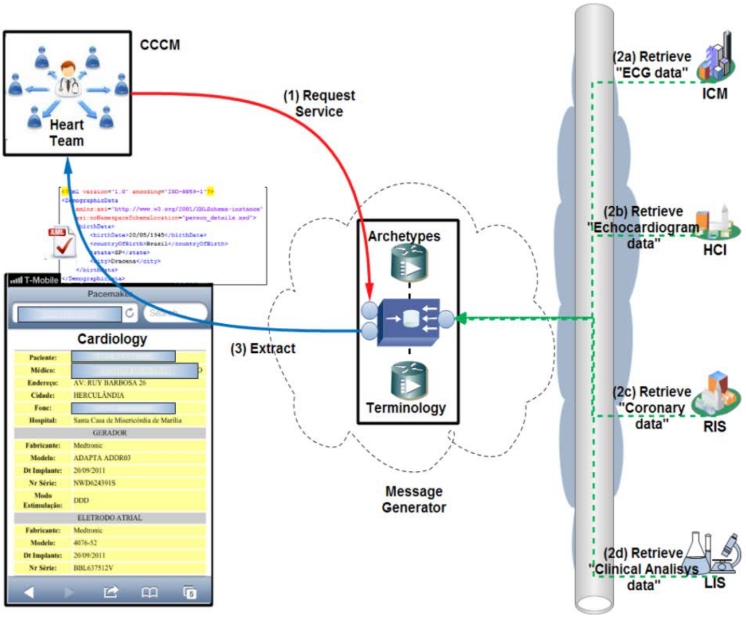
messages that contain the requested information from various healthcare providers, and *ResourceCTR* modifies each one of these messages by serializing the *open*EHR Extract; and (7) these modified messages, which are represented in ACL, are enveloped by *ResourceAgentWS* and delivered to *ResourceAgentCCCM*, and the information related to the patient's EHR is finally displayed on a screen for the Heart Team.

Figure 8-57 Agents Interactions for the Staff Meeting



The message generator supports the data transfer among EHR systems and healthcare stakeholders, allowing interoperable healthcare environments to be built. Although the message generator has been used in all three scenarios, this last scenario is the most appropriate to show its usefulness, since the information related to the patient's EHR is distributed among several healthcare providers: Cardiology Institute of Marília (ICM), Heart Center Institute (HCI), Radiology Information System (RIS), and Life Institute (LI).

Figure 8-58 Message Exchange Facilities



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DemographicData
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="person_details.xsd">
    <birthData>
        <birthDate>20/05/1945</birthDate>
        <countryOfBirth>Brasil</countryOfBirth>
        <state>SP</state>
        <city>Dracena</city>
    </birthData>
</DemographicData>
```

Cardiology

Paciente:	[redacted]
Médico:	[redacted]
Endereço:	AV: RUY BARBOSA 26
Cidade:	HERCULÂNDIA
Fone:	[redacted]
Hospital:	Santa Casa de Misericórdia de Marília
GERADOR	
Fabricante:	Medtronic
Modelo:	ADAPTA ADDR03
Dt Implante:	20/09/2011
Nr Série:	NWD624391S
Modo Estimulação:	DDD
ELETRODO ATRIAL	
Fabricante:	Medtronic
Modelo:	4076-52
Dt Implante:	20/09/2011
Nr Série:	BBL637512V

The message generator provides services via its *openEHRRequest, openEHRExtract* and *Retrieve* interfaces. *Figure 8-58* shows Dr. Call requesting the information related to the patient's EHR in order to plan the pacemaker implantation (1). The message generator processes the *Request* containing the required contents, such as ECG data (from ICM), Echocardiogram data (from HCI), Coronary data (from RIS), and Clinical Analysis data (from LIS). This clinical information is queried and retrieved (2a-d), and the message generator serializes four *open*EHR Extracts, which are transformed into XML documents and forwarded to the requester (3).

8.3 Evaluation

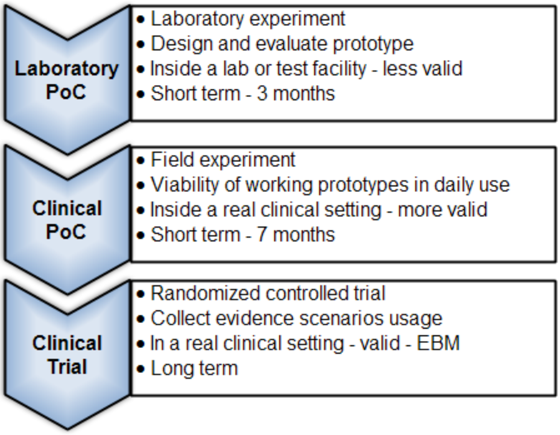
Below we describe the proof-of-concept we built to evaluate our message exchange support. After that, we applied Technology Acceptance Model (TAM) to interpret the data we obtained in this evaluation.

8.3.1 Description

In [4], a Technical Proof-of-Concept (PoC) has been proposed for new ubiquitous computing technology and this PoC has been applied to evaluate subsequently the implementation of a ubiquitous applications in a limited setup. In a pervasive healthcare environment, this approach seems to lack some rigor when investigating whether some technology addresses health-related challenges. In the healthcare domain, this kind of evaluation is usually not enough for new healthcare information technologies, and a methodology named Clinical Proof-of-Concept [77] was proposed for striking the balance between a Technical Laboratory PoC and full-scale Clinical Trials.

In a Clinical PoC, the healthcare information technology should be a working prototype that is usable (but not necessarily user-friendly) by stakeholders, works on its own, and addresses specific research questions [25]. This technology should be deployed in a real clinical setting using daily scenarios, should be used by real users for a short-term, but for a sufficient period of time [76]. A Clinical PoC has revealed that the methods used for evaluating the healthcare information technology are appropriated for judging the perceived usefulness of a given technology [13]. Furthermore, a Clinical PoC may expose implementation issues that would otherwise have not been observed under laboratory conditions. *Figure 8-59* shows the sequence of the Clinical PoC method.

Figure 8-59 Sequence of Proof-of-Concept


Laboratory PoC
• Laboratory experiment
• Design and evaluate prototype
• Inside a lab or test facility - less valid
• Short term - 3 months
Clinical PoC
• Field experiment
• Viability of working prototypes in daily use
• Inside a real clinical setting - more valid
• Short term - 7 months
Clinical Trial
• Randomized controlled trial
• Collect evidence scenarios usage
• In a real clinical setting - valid - EBM
• Long term

The evaluation of the reusable architecture we developed involved a limited but representative set of healthcare professionals and patients, as shown in *Table 8-5*. First, we applied a Technical Laboratory PoC during 3 months in the proposed scenarios to enable the use of these systems in the daily activities of these people. During these 3 months of a longer deployment period we carried out a series of interviews and laboratory studies of these scenarios. Our goals were to study issues of medical treatments, division of work and tasks, communication between healthcare professionals and patients, patient's self-understanding and the technology used by legacy systems in healthcare environments. This study conducted in the laboratory gave evidence about the improvements necessary to achieve suitable message exchange between healthcare professionals.

After this period, we applied a Clinical PoC during 7 months to collect evidences about the findings expected in these scenarios. During this period, everything was done exactly as in real medical procedures for each scenario. In the Clinical PoC, our goals were to provide objective measurements on the usefulness and ease of use of our communication systems and, at the same time, to investigate the end-users reaction concerning the intention to use the applications we developed.

For instance, in the scenario *Delivery of Laboratory Analysis Results* we tested the communication system for delivering the laboratory analysis results and evaluated if end-users were able to properly exchange analysis results over time. In this scenario, the Clinical PoC involved a technical prototype, which was executed with limited interference from the researchers.

The development of the scenarios included six domain engineers, which is not statistically significant for clinical evidence, but sufficient for establishing the viability of the technical setup and its use in realistic scenarios. By actually deploying the integrated technologies in the Cardiology Healthcare Network and asking the stakeholders to use the applications during real-world in healthcare environment, a huge amount of issues surfaced, which have not been found otherwise, such as that the end-user interface had to be improved due to the distance from the table to the screen during the meeting for a cardiac surgery.

However, this Clinical PoC setup is not sufficient to provide clinical evidence for improved patient safety inside the healthcare environment during the medical procedures. In our research, we did not intent to investigate if our communication systems would improve patient safety. This would require a randomized clinical trial over a longer period of time involving a control group, which in its turn would require a full working system ready for large-scale and long-time deployment. Providing such clinical evidence is, however, not the purpose of a Clinical Proof-of-Concept, which has been designed to investigate the feasibility of the proposed solution for further development.

In our research, the core benefit of running the Clinical PoC was the different problematic issues regarding the designed prototypes, which must be addressed before making a larger clinical trial. These issues concerned how to combine different technologies, acceptance measurement and team-based aspects that need to be addressed in combination. Most importantly though, these complex and interrelated issues in the Pervasive Healthcare environments would probably never have been found without running a Clinical PoC.

For this purpose, we employed the Technology Acceptance Model (TAM) [238] to interpret the data, since it is generally considered as an appropriate model to predict technology acceptance [239].

8.3.2 Research Model and Hypotheses

Directly measure IT contribution has been a challenge for most researchers due to its intangible benefits. IT measures have been developed, such as technology acceptance, which directly relates to IT usage. In our research, we applied the TAM model to fully understand the determinants of IT acceptance, to be able to plan effective IT implementation strategies and promote IT usage in healthcare environment. Many previous studies have adopted and expanded the TAM model, which showed empirical evidence of having high validity.

TAM theorizes that an individual intention to adopt a technology is determined by two beliefs, *Perceived Usefulness* and *Perceived Ease of Use*. *Perceived Usefulness* is defined as the degree to which an individual believes that using a particular technology would enhance his productivity, while *Perceived Ease of Use* is defined as the degree an individual believes that using a particular system would be free of effort [26]. Between these two, *Perceived Ease of Use* has a direct effect on both *Perceived Usefulness* and technology usage [240]. Therefore, TAM assumes that the *Perceived Usefulness* (PU) and *Perceived Ease of Use* (PEU) can predict the use and *Intentions to Use* (IU) of a particular technology. PU and PEU are significant factors that affect IT acceptance, and previous research has empirically found positive relationships between PEU and PU as critical factors for the use of IT in healthcare environments [241-245]. A technology perceived to be useful and perceived to be easy to use is more likely to be accepted by its users [246-248].

Subsequent research [249] refined TAM, suggesting that PU and PEU could be affected by external variables, such as, computer self-efficacy, which is an important external variable, and a positive relationship exists between higher computer self-efficacy on the one hand, and PU and PEU on the other hand. In our research, we have used the TAM model [26] minus the external variables. We adopted this research model because of the increasing interest of healthcare professionals to apply health information technology to have proper access to clinical information in healthcare environment. The research model adopted in our work is shown in *Figure 8-60*. Based upon these arguments, we have the following hypotheses:

Hypothesis 1: *Perceived Usefulness* positively affects the *Intention to Use* of the interoperable communication system.

Hypothesis 2: *Perceived Ease of Use* positively affects the *Intention to Use* of the interoperable communication system.

Hypothesis 3: *Perceived Ease of Use* positively affects the *Perceived Usefulness* of the interoperable communication system.

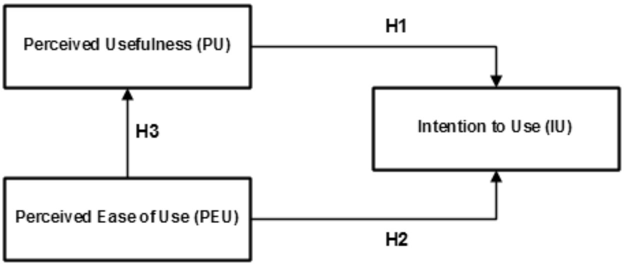


Figure 8-60 Research Model based on TAM

8.3.3 Method

The data for this study were obtained according to the settings and participants described in Section 8.1. We undertook an intensive study of the relevant literature to identity existing measures for the related variables of TAM. When developing our research model, we designed a questionnaire according to the variables defined in the TAM model. The questionnaire was tested in a pilot with five healthcare professionals involved in our project in order to guarantee the validity of the questionnaire, and we conducted face-to-face discussions with these professionals after they completed the questionnaire. Based on their feedback, we refined the survey questions. The questionnaire contains items that measured three variables of our interest, namely *Perceived Usefulness, Perceived Ease of Use* and *Intention to Use*. *Table 8-6* shows the structured questionnaire based on TAM that was sent to the healthcare professionals involved in the scenarios, and that was distributed to patients after medical procedures (see Appendix B for the complete questionnaire). Concerning familiarity with the operations of the legacy system, respondents had previous experience with the usage of this system.

Likert Scale [250] was used in our work as it is the most commonly used measure in scale design. Three-point and seven-point Likert scales have had a large popularity in community research. However, we applied a five-point Likert Scale, because a three-point Likert Scale reduces people's strongest and mildest opinion, while a five-point Likert Scale can express it ideally, and a seven-point Likert Scale causes confusion for people with poor distinguishing ability. Therefore, in our work we adopted a five-point Likert Scale with the responses rated as follows: 1 as strongly disagree, 2 as disagree, 3 as somewhat agree, 4 as agree, and 5 as strongly agree. To collect the data, respondents were asked to indicate their opinion with respect to several statements based on this five-point Likert scale. To control for bias, the questionnaires randomly intermixed items across constructs (PU, PEU, IU), and we conducted a group pre-test to ensure that the scales were appropriate. About 80% and 73% of the distributed questionnaires were duly completed by the healthcare professionals and patients, respectively.

Table 8-6 Questionnaire

Structured Questionnaire		
Perceived Ease of Use	PEU1	My interaction with the system is clear and understandable.
	PEU2	Interacting with the system does not require a lot of my mental effort.
	PEU3	I find the system is easy to use.
	PEU4	I find easy to get from the system what I want it to do.
Perceived Usefulness	PU1	Using the system improves my performance in my job.
	PU2	Using the system increases my productivity in my job.
	PU3	Using the system enhances my effectiveness in my job.
	PU4	I find the system useful in my job.
	PU5	Using the system my tasks are facilitated.
Intention to Use	IU1	Assuming I have access to the system, I intend to use it.
	IU2	Given that I have access to the system, I predict I will use the system.
	IU3	Using the system in my job is a good idea.

8.3.4 Results

Our analysis was divided in two parts: (1) the validity and reliability of the questionnaire were tested by *Cronbach's Alpha* [251]; and (2) to examine the research model and the hypotheses, the data were analyzed using *Structural Equation Modeling* (SEM) [252], which is a statistical method to analyze relationships among variables in a research model. *Table 8*-7 shows the validity tests for all scenarios, in which the internal consistency of the TAM constructs were conducted using reliability. The constructs have *Cronbach's Alpha* values close to the limit of 0.700 [251], which is considered acceptable. After that, the measurement of IT usage generated the covariance matrix; there was no evidence that the measurement model is incorrect. *Table 8*-7 demonstrates an average value of every construct variable (Appendix B). Based on this analysis of the data we concluded that healthcare information system users have responded positively in the perceived usefulness, perceived ease of use and intention to use, which indicate that the research model applied in our work was appropriated.

Table 8-7 Statistics and Reliability of Constructs

Participants	Construct	Items	Mean	Cronbach's Alpha
Healthcare professionals	PEU	4	3.01 to 4.05	0.803 to 0.815
	PU	5	3.12 to 4.70	0.851 to 0.860
	IU	3	3.18 to 4.49	0.799 to 0.854
Patients	PEU	4	3.10 to 4.33	0.850 to 0.895
	PU	5	3.53 to 4.66	0.862 to 0.923
	IU	3	3.04 to 4.16	0.764 to 0.865

In order to test these hypotheses, we have chosen Statistical Regression Analysis [252, 253] over the collected data from the users in each scenario, and the results are summarized in *Table 8-8*. For example, we tested the hypothesis in the scenario *Delivery of Laboratory Analysis Results,* where the participants are healthcare professionals. Regarding the associations between perceived usefulness and intention to use (H1), the standard coefficient of perceived usefulness and intention to use is 0.389, which confirms H1. Meanwhile, we also concluded that perceived usefulness positively influences the users' intention to use of the application built with our architecture. In addition, the standard coefficient of perceived ease of use and intention to use (H2) is 0.542, thus, H2 is supported, reflecting that perceived ease of use positively affects the users' intention to use the application built with our architecture. Concerning the relations between perceived ease of use and perceived usefulness (H3), the standard coefficient of perceived ease of use and perceived usefulness is 0.399, thus, H3 is proved, showing that perceived ease of use has a positive impact on the users' perceived usefulness of the application built with our architecture. According to the relationship between the variables, the perceived ease of use affects users' intention to use both directly and indirectly, through H3 and H2, with values at 0.155 and 0.399 (by multiplying 0.389 * 0.399, plus the path coefficient 0.399), hence, the overall influences is 0.554. Our analysis shows that perceived ease of use has the largest impact on users' intention to use indirectly, when perceived usefulness is influenced by perceived ease of use, with an overall influence of 0.554.

Table 8-8 Statistic Regression Analysis

Scenarios	Hypotheses	H1	H2	H3
	Participants	Path Coefficient	Path Coefficient	Path Coefficient
Scenario 1	Healthcare professionals	0.389	0.452	0.399
	Patients	0.280	0.372	0.433
Scenario 2	Healthcare professionals	0.383	0.322	0.397
	Patients	0.335	0.432	0.403
Scenario 3	Healthcare professionals	0.495	0.357	0.361
	Patients	0.293	0.498	0.341

According to the methodological review in prior research [254], our research is in line with the previous results. In short, end users' intention to use a system would be influenced positively and directly by their perceived usefulness and perceived ease of use, which in turn displays the importance of perceived ease of use of a given technology. According to our experience in developing healthcare information systems and introducing these systems in a healthcare environment, it is necessary to motivate users to use the system by making the operation and interface of the system simple and easy to learn.

Therefore, we could conclude that all hypotheses were confirmed in all scenarios and at all measurement points. Based on this evaluation, we concluded that the healthcare professionals found that applications built with our architecture were extremely useful for their daily tasks and were easy to use. Most patients have identified some usability benefits, such as the efficient method for exchanging messages.

Our applications built with our reusable architecture were designed to work in a large social and organizational context in a healthcare environment. A Clinical PoC showed suitability for initial investigation of the impact arising from this complex context. Especially in the meeting for a cardiac surgery, we found a significant change in the division of work and interactions between the healthcare professionals from various specialties and the patients.

However, our data analysis has some limitations: the questionnaire model is not completely free of subjectivity, since each respondent can react to it in a particular way; all healthcare professionals were grouped together and the results were generalized, despite their different specialties; *Perceived Usefulness* and *Perceived Ease of Use* were the most important factors to explain people's intention of using a given technology, however, other factors may affect this decision, such as the prior experience and their job relevance [238, 249, 255].

Chapter 9

Related Work

Many initiatives that apply ubiquitous computing technologies, healthcare standards, Software-Oriented Architecture and software agent technologies in healthcare domain have been proposed by the academic community, which demonstrates the relevance of our research. This chapter discusses some initiatives that are related to our methodological and technological support to develop interoperable applications to exchange context-aware messages in pervasive healthcare environments. This chapter also compares these initiatives with our work.

This chapter gives an overview of the related work and is further structured as follow: Section 9.1 presents some of the relevant approaches that applied ubiquitous technologies to healthcare. Section 9.2 discusses some of the related work that applied healthcare standards to allow the interoperability between heterogeneous healthcare information systems. Section 9.3 presents some work where Software-Oriented Architecture has been applied in healthcare domain. Section 9.4 presents some relevant work related to software agent technologies applied in healthcare environment. Section 9.5 presents a comparative analysis among this related work, and briefly discusses most relevant similarities and differences to our work.

9.1 Ubiquitous Computing Technologies

The three major research projects in which ubiquitous computing technologies are applied in healthcare environment are: *Activity-based Computing* at the IT University of Copenhagen, *MobileWard* at Aalborg University, Denmark and *Context-aware Communication System* at General Hospital in Ensenada, Mexico.

9.1.1 Pervasive Computing Support for Hospitals

Bardram *et al.* [256] applied Activity-based Computing (ABC) in order to organize how devices, services, and data are handled in terms of computational activities that facilitate recreating, sharing, and switching computational context on demand at whatever device is at hand. The ABC project [257] followed an iterative design process, addressing five medical themes that reflect major working areas in large hospitals: medicine administration by nurses, medicine prescription by physicians, collaboration between clinicians, medical conferences, and surgery. The ABC project developed a coherent conceptual framework for a pervasive computing platform, which supports a wide range of pervasive computing aspects, ranging from mobility to collaboration to context-aware computing.

The main goal of the ABC project is to provide a programming platform for the development and deployment of computer applications that can be used in the activity-based computing concept. Using these applications, end-users can simply carry on with the various work activities that they are engaged in, and seamlessly transfer information on these activities from one computer to another. In healthcare environment, the public displays are embedded in floors, walls, medicine cabinets, beds, and so on, as shown in *Figure 9-61*.

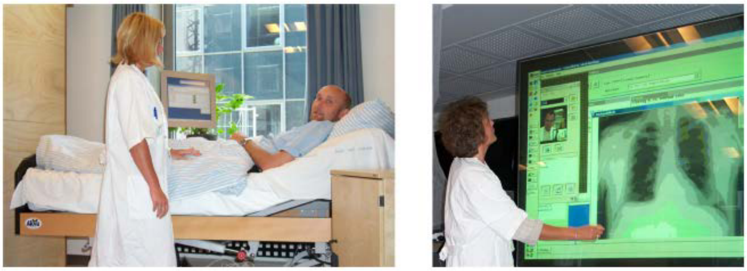
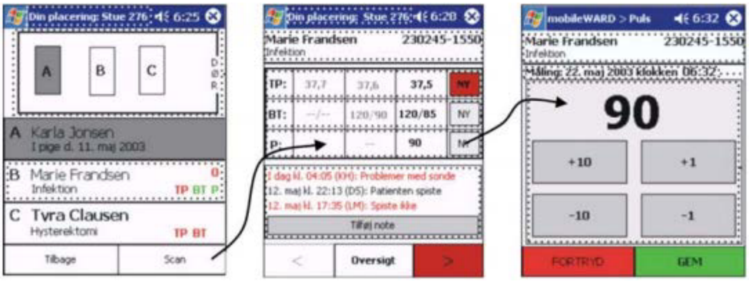


Figure 9-61 The interactive bed and public wall-display (Source: Bardram, 2009)

9.1.2 Exploring context-awareness for ubiquitous computing in the healthcare domain

Kjeldskov *et al.* [258] developed and evaluated an ubiquitous Electronic Patient Record (EPR) system in Frederikshavn (Denmark) hospital. This system, named *MobileWard,* addresses identified challenges of mobility, complexity and relation to work activities by utilizing context-awareness as a key means for supporting the nurses' interaction with EPR system. The aim of this system is to provide the nurse with information that helps her plan the scheduled morning procedure. The system presents information and functionality adapted to the location of the nurse and the time of the day. Furthermore, the system knows the status of each patient and represents already measured values and values yet to be measured with simple color codes, as shown in *Figure 9-62*.


Din placering: Stue 276 6:25

A Karla Jonsen
I pige d. 11. maj 2003

B Marie Frandsen 0
Infektion TP BT P

C Tvra Clausen
Hysterektomi TP BT

Din placering: Stue 276 6:28

Marie Frandsen 230245-1550
Infektion

TP:	37,7	37,6	37,5	NY
BT:	--/--	120/90	120/85	NY
P:		--	90	NY

I dag kl. 04:05 (KH): Problemer med sonde
12. maj kl. 22:13 (DS): Patienten spiste
12. maj kl. 17:35 (LM): Spiste ikke

mobileWARD > Puls 6:32

Marie Frandsen 230245-1550
Infektion

Måling: 22. maj 2003 klokken 06:32

90

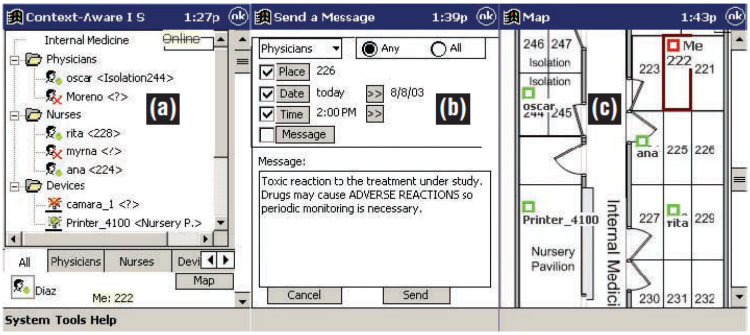
Figure 9-62 Screens displayed in the ward in relation to the tasks of measuring temperature, blood pressure, and pulse (Source: Kjeldskov *et al.,* 2007)

The results of this research showed that workers in the healthcare domain could benefit from ubiquitous computing environments, which may be improved by offering mobile and context-aware points of access. However, these studies also confirm that the design of ubiquitous computing system for the real world needs further investigations.

9.1.3 Context-aware Mobile Communication in Hospitals

A context-aware communication system is reported in [259] in which mobile devices recognize the context in which healthcare professionals perform their tasks. This communication system was developed at the IMSS General Hospital in Ensenada, Mexico, which is a public-health institution that is the core provider in a healthcare network. Muñoz *et al.* proposed an extension of the traditional Instant Messaging (IM) paradigm by using the Extensible Messaging and Presence Protocol (XMPP) for exchanging XML messages. These messages contain contextual information, including the essential information that allows the system to deliver messages, such as location, delivery timing, role reliance, and artifact location and state.

As shown in *Figure 9-63* (a), the client user notes the status of the system users *(online, busy,* and *disconnected)*, as well as the resources available in the vicinity, such as printers or public displays, their status, and the services they can provide. The client also shows the location of users and devices, if known. In *Figure 9-63* (a), this information is shown in brackets after the person's name (ward or room number). In contrast to traditional IM, the sender of a context-aware message must specify delivery context. *Figure 9-63* (b) shows the field that users fill in to write a message and specify their context. The sender specifies a physical area where the recipient must be found for the message to be delivered. To aid in this process, the system displays a map, as in *Figure 9-63* (c), which the sender can tap on to select the desired area. The sender can specify a lower bound or an upper bound to the time and date, or both.



☑ Place 226
☑ Date today >> 8/8/03
☑ Time 2:00 PM >>
□ Message

Toxic reaction to the treatment under study. Drugs may cause ADVERSE REACTIONS so periodic monitoring is necessary.

Figure 9-63 Interaction with the context-aware handheld system (Source: Muñoz *et al*, 2003)

9.2 Healthcare Standards

Standards are critical for the interoperability of electronic healthcare information systems. In this section, we discuss some standardization efforts related to healthcare standards, such as ISO 13606 and *open*EHR dual model.

9.2.1 Building a logical EHR architecture based on ISO 13606 standard and semantic web technologies

Santos *et al.* [260] proposed an architecture based on ISO 13606 and on the utilization of semantic technologies for a EHR scenario, as shown in *Figure 9-64*. The architecture was designed to allow different EHR systems to interoperate by offering an integrated service, preserving the existing semantics in the knowledge domain, and updating clinical data for each patient in a consistent way. The messages exchanged among the EHR systems are based on XML files, aiming to facilitate their utilization by EHR system suppliers. Although the repositories in the architecture are based on a reference model and on archetypes, the architecture needs to anticipate the participation of EHR systems that do not use archetypes. In [260], the ability of the ISO 13606 EHR reference model to accommodate the scenario was highlighted, together with the support provided by the use of the ontology specification languages (RDF and OWL) with respect to the maintenance of a controlled vocabulary.

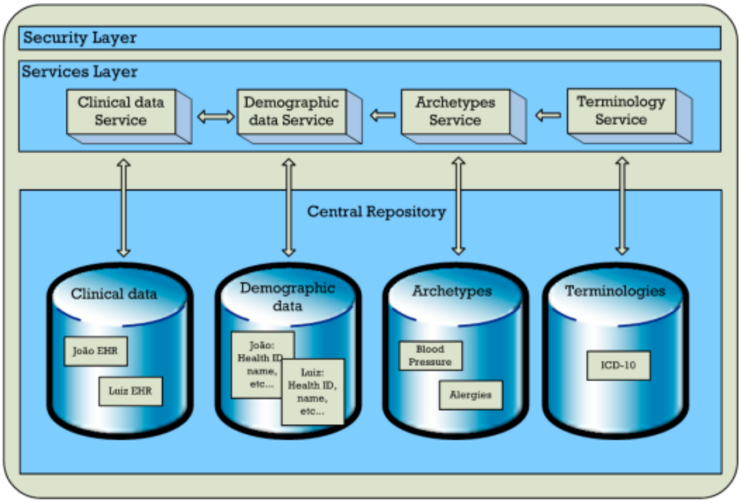


Figure 9-64 EHR system architecture (Source: Santos *et al.*, 2010)

9.2.2 Modeling healthcare authorization and claim submissions using the *open*EHR dual-model approach

Dias *et al.* [261] adopted three approaches according to *open*EHR RM for modeling healthcare authorization and claim submissions among healthcare plans and providers in Brazil. In the first approach, a set of archetypes was designed using ENTRY subclasses. In the second one, a set of archetypes was designed using exclusively ADMIN_ENTRY and CLUSTERs as their root classes. In the third approach, the *open*EHR ADMIN_ENTRY is extended with classes designed for authorization and claim submissions, and an ISM_TRANSITION attribute is added to the COMPOSITION class, as shown in *Figure 9-65*. Another set of archetypes was designed based on this model. This extended *open*EHR RM model is therefore semantically aligned with the concepts involved in authorization and claim submissions. Although this work focuses on the modeling healthcare authorization and claim submissions, and assumes that a communication architecture to simulate the exchange of administrative data between systems according to the *open*EHR approach should be available.

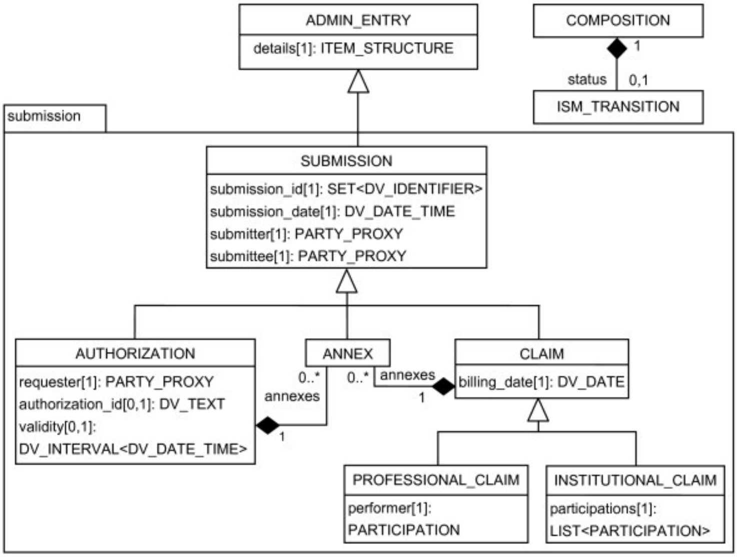
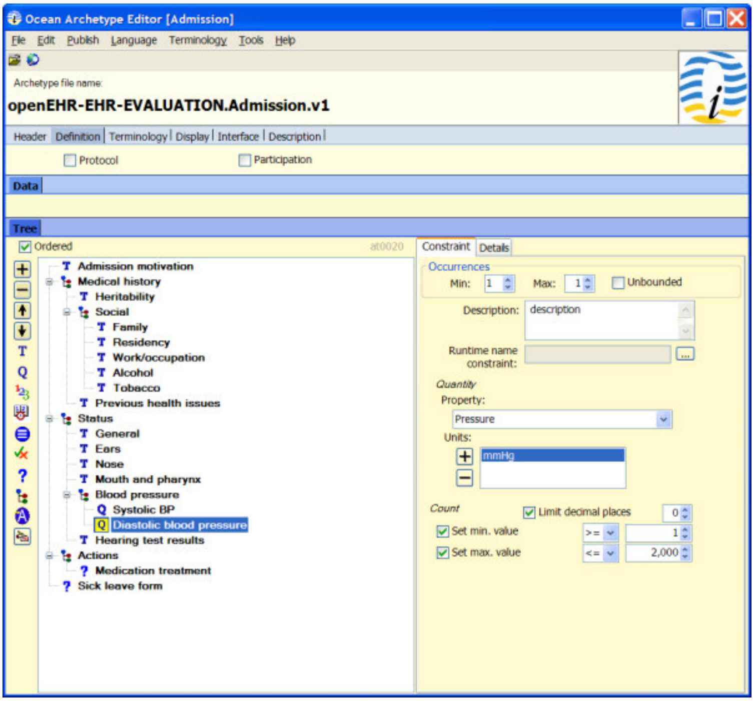


*Figure 9-65 open*EHR extended RM for authorization and claim submissions (Source: Dias *et al.,* 2011)

9.2.3 Archetype-based conversion of EHR content models: pilot experience with a regional EHR system

Chen *et al.* [262] investigated the feasibility of transforming EHR templates from an installed Cambio COSMIC system[38], which a regional EHR product deployed in Sweden and in several other countries, into *open*EHR archetypes, and inversely the feasibility of converting standard archetypes into the COSMIC template formalism. Several archetypes from the *open*EHR Clinical Knowledge Repository have been imported into COSMIC, preserving most of the structural constraints and those related to terminology. COSMIC templates from a large regional installation were successfully converted into the *open*EHR archetype format. The conversion from the COSMIC templates into archetype format preserves nearly all structural and semantic definitions of the original content models, as shown in *Figure 9-66*. The result of this work showed that models of clinical content from an existing EHR system with a large installed base covering both primary care and specialist hospital care could be consistently represented as archetypes. This result indicated the applicability of the *open*EHR archetype methodology and verified the expressiveness of the *open*EHR models. With some limitations, internationally available archetypes could be converted to legacy EHR models.

[38] http://www.cambiosys.com/



Archetype file name: openEHR-EHR-EVALUATION.Admission.v1

□ Protocol □ Participation

☑ Ordered at0020

Min: 1 Max: 1 □ Unbounded

Description: description

Runtime name constraint:

Property: Pressure

Units: mmHg

☑ Limit decimal places 0

☑ Set min. value >= 1

☑ Set max. value <= 2,000

Figure 9-66 Exported COSMIC template in the form of an archetype (Source: Chen *et al.*, 2009)

9.3 Software-Oriented Architecture

Software-Oriented Architecture has been applied in many initiatives to support application development and provide interoperability among healthcare information systems. In this section, we present approaches that applied the SOA principles.

9.3.1 Interoperability of Personal Health Records

Lähteenmäki *et al.* [263] proposed an approach to achieve interoperability between self-care systems, when they exchange non-clinical information alongside clinical data. This work is based on web services for communicating with heterogeneous systems, where SOAP messages are defined for transporting the Personal Health Record (PHR) content in structures of the so-called Health Diary Entry (HDE), allowing in this way the use of external vocabularies and ontologies in order to achieve semantic interoperability, as shown in *Figure 9-67*. The non-clinical PHR content is typically composed of information originating from the user or from a measurement device controlled by the user. A Health Diary Entry document is a collection of health diary entries of one of the following types: occurred event, goal and background entry.

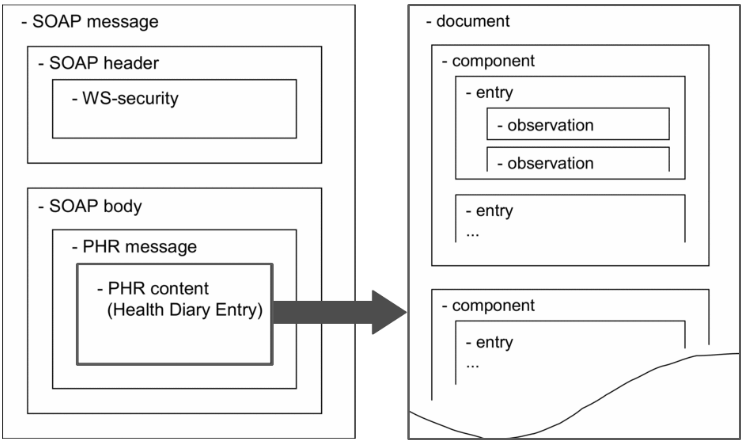


Figure 9-67 PHR message structure (Source: Lähteenmäki *et al.*, 2009)

9.3.2 A SOA Service Governance Approach to u-Healthcare System with Mobility Capability

Yu *et al.* [264] proposed an SOA-based architecture for a wireless mobile healthcare system, called *uSGHealth*. The system was divided into subsystem components developed by applying tier-based and component-based system design approaches. The system mainly focuses on SOA and Service Governance. The system components include: *SOA framework, Service Governance, Business Layer, Database Layer, Application Server,* and *Mobile clients,* as shown in *Figure 9-68*. The architecture handles client's request and responds with the desired output. When clients like patients and healthcare professionals issue requests for any kind of service, the architecture handles these requests efficiently. According to the authors, medical devices can transmit the health information through mobile devices to the distributed EHR database by using their approach.

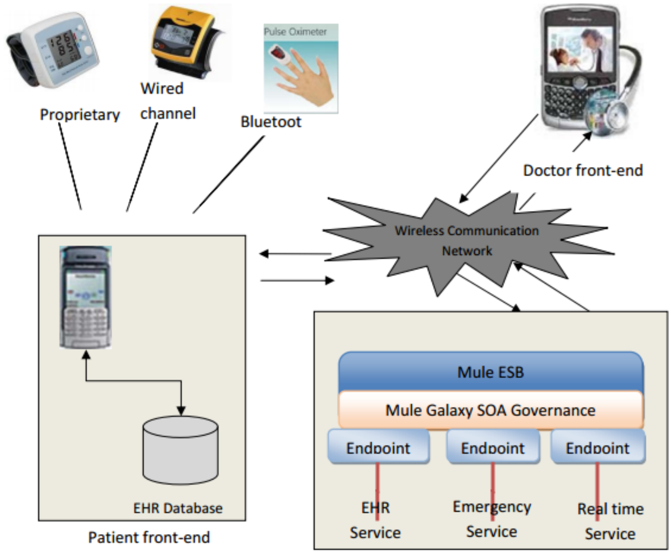


Figure 9-68 uSGHealth System Architecture (Source: Yu *et al.,* 2011)

9.3.3 REST*ful* dissemination of healthcare data in mobile digital ecosystem

Kazi *et al.* [265] proposed a mobile healthcare architecture named *Pain Information on the Go (PlnGO)*, which was developed in collaboration between Bioinformatics Research Lab and MADMUC lab at the University of Saskatchewan, in Canada. *PInGo* runs on mobile devices and has been used in healthcare training program for children who suffer from chronic pain with arthritis at Royal University Hospital, in Canada. The architecture consists of three layers: User front-end layer, Web Server layer and Cloud hosted persistent data storage layer, as shown in *Figure 9-69*. *PInGO* is a questionnaire-based electronic pain diary developed to examine and assess the muscles improvement of *Juvenile Idiopathic Arthritis* (JIA) patients. Infant patients were provided with mobile device running the application and were required to fill-in the questionnaire as they performed the exercise module over a 6-week training period. In order to propagate questionnaire data from patient's device to the healthcare professionals in a fast and secure manner, on event-driven publish/subscribe-based architectural model was adopted in the *PInGO* architecture.

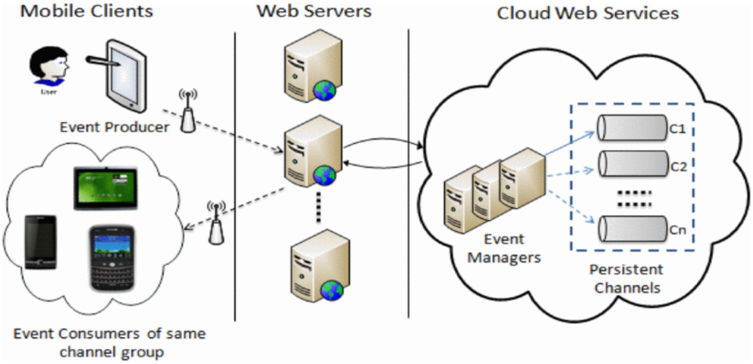


Figure 9-69 Architecture adopted for the *PInGO* app (Source: Kazi *et al.*, 2013)

9.4 Software Agent Technologies

Software agent technologies allow task coordination for message exchange between healthcare professionals and patients. In this section, we present related work in which software agent technologies have been applied in healthcare.

9.4.1 A Multi-agent System Approach to Drugs Management Using Combinatorial Auctions

Baffo *et al.* [266] adopted an approach based on the following assumption: a hospital is a complex system composed by a large number of entities and processes that have to cooperate to reach a common solution to operation and tactical problems. The MAS is supposed to consist of several decisional and operational agents that can be related to departments and wards. The work presented a model of agent cooperation for addressing the problem of departments, wards and personnel allocation in order to improve efficiency regarding drugs management, as shown in *Figure 9-70*. The application of this model with ICT solutions to drugs procurement allows the improvement of the FIFO (*First-In-First-Out*) material management strategy. The logistic department should have a strategic role inside of hospital because it can become the gathering point of all hospital information related to state and location of drugs and equipment.

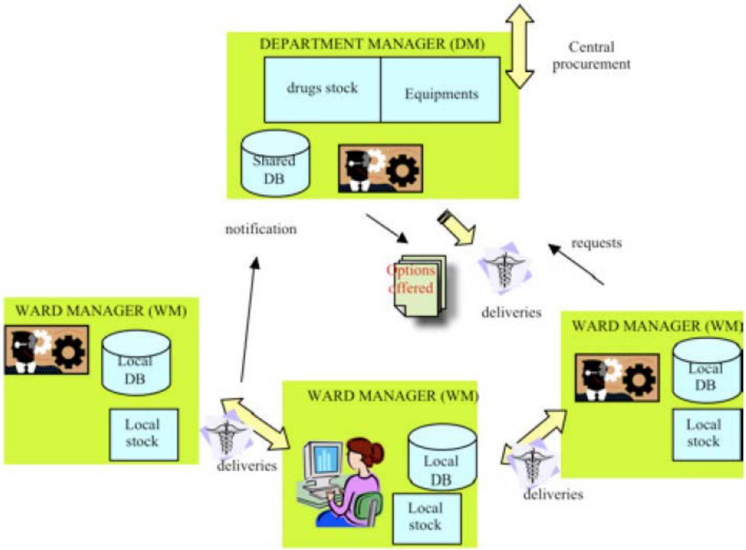


Figure 9-70 Multi Agent Collaborative System (Source: Baffo *et al.*, 2010)

9.4.2 Knowledge-driven delivery of home care services

K4Care [267, 268] is a large European Union (EU) project whose main goal is to design, implement, and validate a new Home Care Model (HCM) to help manage the home care assistance required by the increasing EU senior population. Batet *et al.* [269] proposed the knowledge-based personalization aspects that have been incorporated in the K4Care platform, as shown in *Figure 9-71*. An important characteristic of this platform is that it separates medical and organizational knowledge, which are modeled by means of several machine-readable knowledge and data sources, from the software components responsible for to the execution of the system. The *Knowledge Layer* contains the EHR, which stores clinical information about the patients using standard XML documents. The *Data Abstraction Layer* is a mediator component between the *Knowledge Layer* and the platform that allows the required data and knowledge to be transmitted in a transparent way. In the *Agent-based Layer*, the execution logic of the K4Care platform is implemented using a multi-agent system. Users can interact with their agents through a web-based application. With this approach, a knowledge-driven system has been developed, which dynamically adapts its behavior when the knowledge base issues a modification.

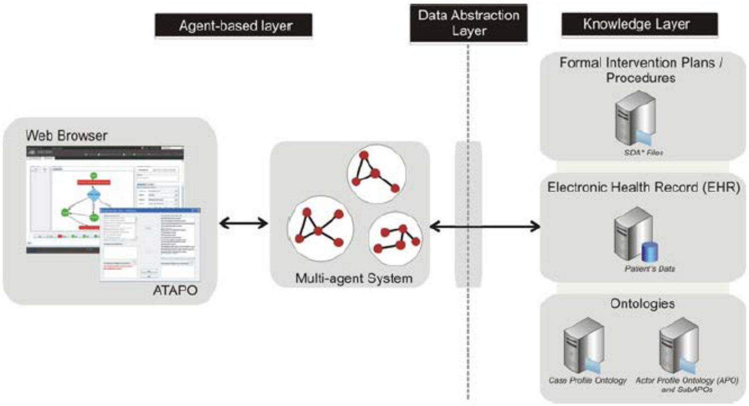


Figure 9-71 K4Care platform architecture (Source: Batet *et al.*, 2012)

9.4.3 A peer to peer agent coordination framework for IHE based cross-community health record exchange

A distributed coordination framework for dynamically exchanging EHRs between healthcare communities has been proposed by Urovi *et al.* [270]. These communities are connected by means of a *Peer to Peer* (P2P) model, and they use a multi-agent platform and a set of distributed rules to coordinate the agents in the search of specific health records. *Figure 9-72* (a) shows the agent coordination architecture used to coordinate the interactions for one single community. *Figure 9-72* (b) shows the healthcare communities connected in a P2P network, which can make use of a scalable mechanism for search queries and event notification. A community interested in finding data about a specific patient queries the P2P to find which communities hold these data, and performs a direct query to these communities to retrieve the proper information.

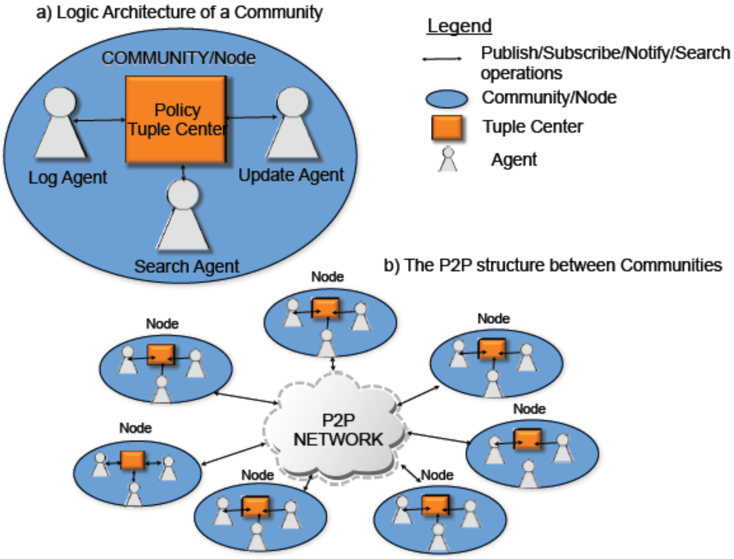

a) Logic Architecture of a Community
COMMUNITY/Node
Policy Tuple Center
Log Agent
Update Agent
Search Agent
Legend
Publish/Subscribe/Notify/Search operations
Community/Node
Tuple Center
Agent
b) The P2P structure between Communities
Node
P2P NETWORK

Figure 9-72 The logic architecture of the community nodes (Source: Urovi *et al.*, 2013)

9.5 Comparative Analysis

In the sequel we briefly discuss the most relevant similarities and differences between the work discussed in the previous sections and ours.

9.5.1 Ubiquitous Computing Technologies

Bardram *et al.* [256, 257], Kjeldskov *et al.* [258] and Muñoz *et al.* [259] proposed the use of ubiquitous computing technologies in healthcare environments for continuous wellness and real-time illness diagnosis. Similarly to our work, these ubiquitous computing technologies are applied to provide patient records available for healthcare professionals from any given location and time. However, these approaches do not apply any standard language for agents communication and SOA principles, which we have chosen to provide reusable and interoperable services. These authors also disregard healthcare standards for the exchange of clinical concepts.

9.5.2 Healthcare Standards

Santos *et al.* [260], Dias *et al.* [261] and Chen *et al.* [262] applied healthcare standards on their work to enable the interoperability among heterogeneous healthcare information systems, similarly to our work. However, they do not take advantage of agent communication language for the exchange of patient clinical information and they do not apply ubiquitous technologies to allow the health information available through any device and in any location. Unlike in the approach described in our work, these approaches do not apply SOA implementation to help solve the challenge of exchanging clinical information across the healthcare providers by providing opportunities to integrate isolated healthcare information systems.

9.5.3 Software-Oriented Architecture

Lähteenmäki *et al.* [263], Yu *et al.* [264] and Kazi *et al.* [265] designed architectures for exchanging information in healthcare environments. These architectures have some similarities with ours, since they deal with the interoperability of heterogeneous systems by using SOA technologies, but they do not use software agent technologies, which could be provided with a flexible interaction mechanism for allowing them to cooperate by exchanging healthcare content. Moreover, in these architectures healthcare standards are not employed to enforce interoperability at the message level between heterogeneous systems, which brings difficulties with sharing and reuse of legacy systems.

9.5.4 Software Agent Technologies

Baffo *et al.* [266], Batet *et al.* [269] and Urovi *et al.* [270] follow similar approaches, as they employed software agents for controlling clinical tasks and coordinating the exchange of EHR content. In these approaches, ubiquitous technologies were not applied and we did in our work. Furthermore, these approaches do not applied SOA technologies to help enable the interoperability between healthcare information systems. In these approaches, healthcare standards are not employed to provide interoperability between healthcare information systems.

Chapter 10

Conclusions and Future Work

This thesis focused on designing methodological and technical support to develop interoperable applications to exchange context-aware messages in pervasive healthcare environments. This methodology is based on the use of ubiquitous computing technologies, healthcare standards, Software-Oriented Architecture and software agents technologies, and has been validated with a case study.

In this chapter, we discuss what we learned and then propose identify some challenges as potential future research topics. In Section 10.1, we summarize the general contributions of this thesis. Section 10.2 elaborates on the most important research contributions of this thesis. Section 10.3 discusses future work based on the limitations identified in our work. Section 10.4 summarizes the chapter.

10.1 General Considerations

In our research, we observed that the current healthcare model applied in most countries is still centered on highly specialized people located in large hospitals and focused on acute cases for treatment. However, this healthcare model has become inadequate, which indicates that the centralized healthcare model needs to change into a distributed model to produce faster responses and to allow the patients to manage their own health.

We argued in this thesis that the design of the Pervasive Healthcare Model is a challenging task, which proves why the development of novel methodological and technical support to allow efficient and safe information exchange between healthcare professionals and to provide healthcare information anywhere, anytime to anyone is still not widespread.

We aimed in this thesis to define a Reusable Architecture, supported by the *open*EHR dual model approach, ubiquitous computing technologies, service-oriented architecture and software agents. Our study addressed the architectural and technological challenges of combining these technologies to achieve our goals. This thesis presented a Reusable Architecture and a Message Generator that employ these technologies for exchanging context-sensitive messages among healthcare professionals and patients. We showed that our Reusable Architecture and Message Generator meet the requirements of interoperability between heterogeneous healthcare information systems that are usually found in Pervasive Healthcare environments.

In our investigation, we learned that software agents can support healthcare professionals in their daily practices since they are able to properly perform communication tasks on behalf of these professionals. Software agents can also monitor their environment since they are able to properly ensure the fulfillment of the contextual requirements. We also found out that these agents are able to take proper decisions about the activities to be performed and when and how to communicate to perform them.

The Message Generator was designed so that it offers Web service interfaces for achieving interoperability in the healthcare domain. The use of REST*ful* Web services standardizes the mechanisms used to describe, discover, and access the resources offered by the Message Generator, in which non-*open*EHR systems can interoperate by sharing data in the form of XML messages. The message exchange with other systems is accomplished by integrating EHR extracts represented in terms of archetypes into ACL messages. We believe the Message Generator can be included in a new generation of EHR tools, contributing in this way to the *open*EHR Software Program.

We performed a case study in a realistic distributed healthcare environment, where three usage scenarios were defined. Using the Reusable Architecture and the Message Generator, three communication systems were built for these scenarios, and they were employed and evaluated by both healthcare professionals and patients.

We showed that the methodology proposed in this thesis truly facilitates the development of the Pervasive Healthcare Model. The proposed methodology preserves the investment in the legacy systems and allows developers to add new features to them, aiming to fill the automation gap and satisfying the need for increased user mobility, while providing smart assistance to the end-users.

10.2 Research Contributions

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

RQ1: *What are the current problems found in the context of Pervasive Healthcare environments?*
To answer this question, we had answered several sub-questions as follows:

RQ1.1: *Which stakeholders and with which goals can be identified in the context of healthcare environments?*
In Chapter 2, we acquired more knowledge about the stakeholders and their goals in the healthcare domain. We identified several stakeholders, such as physicians, nurses, assistants and patients, and the hospital environment in which the stakeholders use the legacy healthcare information systems. Healthcare environment involves a huge amount of stakeholders who participate in the treatment of patients, so that each of these stakeholders has different goals and exchanges clinical information in order to cooperate with each other.

RQ1.2: *What are the problematic phenomena perceived by these stakeholders?*
In Chapter 2, we identified the problematic phenomena perceived by the stakeholders in the context of complex healthcare environments. Information in the healthcare domain is quite complex because it covers different types of data, such as clinical data, laboratory data and patient administration.

We have analyzed the challenges in Pervasive Healthcare environment reported in Chapter 2. With this analysis, we have understood that achieving a completely paperless environment within a healthcare environment is an unrealistic expectation. Maturity models should inspire continuous solutions focused on the strategic removal of paper from the healthcare organizations.

Numerous solutions have been proposed to achieve interoperability in the healthcare, with different degrees of success. However, the effective and meaningful information exchange of healthcare information across healthcare information systems is yet to be achieved. We have proposed the use of healthcare standard to improve patient care by allowing interoperability between heterogeneous systems. In the healthcare domain, information is expected to be exchanged in the form of standardized reports.

RQ1.3: *What are the criteria and requirements that influence the stakeholders' goals?*
Based on the problem investigation we formulated the criteria and requirements for our solution, which contribute to the stakeholders' goals. In Chapter 4, we identified functional and non-functional requirements that our methodological and technical support should provide. The requirements were identified by interviewing stakeholders and the literature survey in this area.

RQ2: *How to develop a reusable architecture for Pervasive Healthcare environments that supports the requirements imposed by stakeholders?*
To answer this question, we need to answer several sub-questions as follows:

RQ2.1: *What are the available solutions for message exchange in Pervasive Healthcare environment?*
To answer this question, we performed a case study to identify the potential problems and challenges typically found in Pervasive Healthcare environment, in Chapter 2.

In Chapter 3, we have conducted a literature investigation to identify to what extent available ubiquitous computing technologies, healthcare standards, service-oriented approaches and software agent technologies have been addressed in Pervasive Healthcare environments. With this investigation, we have gained the necessary knowledge to be able to use and extend the most suitable solutions and technologies for allowing the communication between legacy systems in Pervasive Healthcare environments. Based on the identified existing solutions, we concluded that there was a need for adapting them to the objectives and scope of our work.

RQ2.2*: How to design a reusable architecture for message exchange in Pervasive Healthcare environments?*
To define the architectural design of our Reusable Architecture, we have identified a set of requirements for the Pervasive Healthcare environment, presented in Chapter 4 (Section 4.1). We classified the requirements according to three applications scenarios identified by the healthcare professionals and a software engineer in the cardiology domain: *delivery of laboratory analysis results, schedule of an appointment for pacemaker evaluation,* and *support of a medical staff meeting to prepare for a cardiac surgery*. The requirements imposed by the stakeholders determined how the reusable architecture should be designed in order to fulfill the message exchange in Pervasive Healthcare environments.

We have also defined that the Reusable Architecture should be supported by the technologies discussed in Chapter 3 in order to provide a technical solution in the healthcare domain.

In Chapter 5, we introduced an archetype-based modeling process for the *Domain Experts* to support the message exchange of our Reusable Architecture. In this context, clinical archetypes have played an important role in determining how clinical information is represented and organized inside EHRs when they are exchanged between systems.

In Chapter 6, we presented our Message Generator, which acts as middleware to bridge the gap between healthcare applications, and provides reusable services to distributed healthcare systems by allowing the semantic interoperability of heterogeneous healthcare information systems. We have developed facilities to perform message exchange between heterogeneous EHR systems in Pervasive Healthcare environments.

In Chapter 7, we have defined the Agent-based Modeling process to allow the *Domain Engineer* to design the software agents applied in our Reusable Architecture. In our work, the agent models were implemented as ontologies with the advantages of including a machine-readable and human-readable model, reasoning capabilities and facilitated configuration of the models. The software agents implemented in our architecture provide an appropriated mechanism for developing heterogeneous and complex distributed systems due to their ability to cope with distribution and interoperability.

RQ2.3: *What are the typical components of an architecture for Pervasive Healthcare environments?*
In Chapter 4, we presented the components available in our Reusable Architecture and how these components interact with each other. The biggest challenge for the components in our architecture is to support the mobility and collaboration among healthcare professionals when they perform clinical tasks. The components were organized according to the MVC pattern (Model-View-Controller), to separate the business logic from the presentation logic for the sake of flexibility and reuse.

In Chapter 5 (Section 5.3), we have presented the archetypes that were validated by the *Domain Experts* of a Cardiology Healthcare Network in Marília (São Paulo, Brazil). The archetypes were published within the hospital network in order to become available for all participants in the bidding process for healthcare integration. We have also reused some archetypes available in the CKM repository in our architecture.

In Chapter 6 (Section 6.4), we have designed and implemented the components of our Message Generator as a software application that offers and uses REST*ful* Web services to exchange messages directly on top of HTTP.

In Chapter 7 (Section 7.3), we have introduced the *agent* package of our Reusable Architecture, which contains static and mobile agents, each one endowed with specialized capabilities and goals for helping the healthcare professionals check the availability of resources, and obtain information and detect abnormalities on their patients.

RQ3: *What is the effect of using the architecture on application development based on Pervasive Healthcare environments?*
By answering these sub-questions, we had a complete design of our Reusable Architecture and then we explained the implementation of our case studies as a proof of concept in Chapter 8.

RQ3.1: *What is the effect of applying our Reusable Architecture when it is instantiated in different applications scenarios in a Pervasive Healthcare environment?*
Our Reusable Architecture was validated in a realistic Pervasive Healthcare environment. In Chapter 8, we reported our three different application scenarios in the cardiology domain and presented their results. The results demonstrated that our Reusable Architecture can be instantiated in different scenarios in a Pervasive Healthcare environment.

RQ3.2: *How to validate the perceived usefulness and ease of use of the applications designed based on our Reusable Architecture?*
The evaluation of our Reusable Architecture was mainly performed during the case studies presented in Chapter 8. In these case studies, we applied a Technical Laboratory PoC during 3 months in the proposed scenarios to enable the use of these systems in the daily activities of these people. After this period, we applied a Clinical PoC during 6 months to collect evidences from these scenarios, employing the Technology Acceptance Model (TAM), since it is generally accepted as an appropriate model to explain technology acceptance in healthcare. The evaluation involved a limited but representative set of healthcare professionals and patients.

RQ3.3: *What is the level of acceptance of the applications designed based on our Reusable Architecture by the stakeholders?*
According to the results of the evaluation, we have concluded that the perceived usefulness and ease of use of the applications designed based on our Reusable Architecture met the stakeholders' expectations.

Healthcare professionals reported that these applications were extremely useful for daily tasks and they were easy to use. The healthcare professionals were satisfied with the fact that they can gather the patient's clinical information when they perform clinical tasks. These healthcare professionals have gained confidence in the software agents that work on their behalf, in spite of the time taken by this process. Most patients have identified some usability benefits, such as the efficient method for exchanging messages.

RQ3.4: *How can our Reusable Architecture be used to support other applications in the healthcare domain in Pervasive Healthcare environment?*
In Chapter 4, we have discussed a methodology through which a *Domain Expert* and a *Domain Engineer* can design personalized models, such as archetypes and agents models, respectively. Our methodology allows an *Application Engineer* to reuse the components of our Reusable Architecture, meant to be generally applicable to different legacy healthcare information systems, from which applications can be built to be used in realistic healthcare scenarios. In Chapter 8, we have discussed how our methodology can be applied in order to design applications that allow us to shift from legacy healthcare information systems that rely on informal offline communication between actors to an agent-based system.

This methodology can be also employed in other application domains, since archetypes and software agents can be designed specifically to the target domain, and common components can be encapsulated and normalized, facilitating their reuse.

New archetypes can be designed for other applications or domains according to the Archetype Modeling process discussed in Chapter 5. The message exchange between heterogeneous healthcare information systems can be implemented by reusing the web services provided by the Message Generator discussed in Chapter 6. The software agents can be designed by following the Agent-based Modeling process discussed in Chapter 7.

RQ3.5: *How does our Reusable Architecture compare with similar solutions?*
In Chapter 9, we have discussed the body of knowledge with respect to the approaches and technologies that have been applied in Pervasive Healthcare environments in order to allow the message exchange between heterogeneous healthcare information systems. We then identified similarities, assets and differences with our work that have helped us position our Reusable Architecture and Message Generator with respect to the related available solutions.

10.3 Future Research

In further work, we are especially interested in evaluating the performance of our methodology, especially its scalability, which is a crucial non-functional requirement for realistic applications and for the simultaneous support of multiple scenarios. We argued that the methodology is reusable, and we want to confirm this claim in other domains. Therefore, we aim to find domains with different properties and challenges so that the true potential and limitations of the methodology can be analyzed.

In our work, we identified some limitations related to the privacy and security of our Message Generator that could have potential impact if we increase the number of healthcare providers exchanging healthcare information. In future work, we intent to address the privacy and security issues that arise when we integrate these technologies into a legacy healthcare system. Other issues that must be addressed include ensuring the confidentiality and security of message exchanges and healthcare information. Regarding the Message Generator, we intend to reuse an infrastructure security layer to enforce role-based access control and to deploy the architecture and its services in a cloud computing environment.

As the learned from developing our agent-based modeling process, one of the possible solutions is to apply agent technologies to handle the coordination of business processes. In future work, we could provide a mapping from the Business Process Modeling Notation (BPMN) [271, 272], which is a graphical language used to represent business process, to agent concepts in the healthcare domain. We believe that by applying BPMN as the source language for this mapping, our methodology can offer some benefits. For example, the *Domain Experts* and *Domain Engineers* would have a suitable and intuitive graphical notation in order to define process models at different levels of abstraction in the healthcare domain. We also believe that by applying software agents as a target model, we can offer the capabilities of the powerful execution environment typically found in Multi-Agent System platforms.

In our work, we applied *open*EHR specifications in order to allow more efficient sharing of clinical information between heterogeneous healthcare information systems. However, the XML Schema implementation of the *open*EHR specifications is under development and we needed to adapt the schemas in order to validate our archetype models. Multilevel modeling has been proven in software as a viable solution for semantic interoperability, without imposing any specific programming languages or persistence models. The Multilevel Healthcare Information Modeling (MLHIM)[39] specifications have adopted the XML Schema Definition 1.1 as

[39] http://mlhim.org/

the basis for its reference implementation, since XML technologies are consistent across all platforms and operating systems, with tools available for all mainstream programming languages. An interesting topic for further work is to support our Message Generator for MLHIM specifications, in the scope of a Brazilian project namely National Institute of Science and Technology for Medicine Assisted by Scientific Computing (INCT-MACC) to which we contribute. The MLHIM Domain Models do not require top-down consensus to be implemented and they are interoperable. The MLHIM specifications are implemented in XML technologies, and therefore they can count on global technical support and a number of third-part validation tools [141].

Our data analysis has some limitations: the questionnaire model is not completely free of subjectivity, since each respondent can react to it in a particular way; all healthcare professionals were grouped together and the results were generalized, despite their different specialties; Perceived Usefulness and Perceived Ease of Use were the most important factors to explain people's intention of using a given technology, however, other factors may affect this decision, such as the prior experience and their job relevance. We are also planning to extend TAM in order to evaluate the effect of others variables, such as, user experience, job relevance and output quality, to moderate the effects of perceived usefulness and perceived ease of use on the intention to use.

10.4 Summary

What was already known at the start of our work:

- The current Healthcare model is changing to a patient-centric model;
- Healthcare standards are being developed to support interoperability among HISs;
- Efficient information exchange among healthcare professionals and patients is absolutely necessary.

What this study added to the common knowledge:

- Ubiquitous Computing technologies, mainly the context awareness, are useful and suitable in Pervasive Healthcare models for allowing patients to access healthcare information anywhere and at anytime, and how they can be applied;
- The *open*EHR dual model can be used in the design of a Reusable Architecture and a Message Generator that provide services in order to enable interoperability between heterogeneous HISs;

- The use of software agents to build communication systems can support strong collaboration and cooperation between healthcare professionals and patients.

Appendix A

Author Publications

During the development of this thesis, the author has published various parts of this work in the following papers (listed in reverse chronological order):

University of Twente (2012 – 2014)

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Francisco do Prado, A. (2014) A Methodology based on *open*EHR Archetypes and Software Agents for Developing e-Health Applications Reusing Legacy Systems. Computers in Biology and Medicine. ISSN 0010-4825 (submitted in Dez 2013).

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Francisco do Prado, A. (2014) An Architecture for Health Information Exchange in Pervasive Healthcare Environment. In: Enterprise Information Systems. Lecture Notes in Business Information Processing 190. Springer International Publishing Switzerland, Switzerland, pp. 1-17. ISSN 1865-1348 ISBN 978-3-319-09491-5.

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Cavalini, L.T. and do Prado, A.F. Uma Arquitetura para Troca de Mensagens usando Agentes Inteligentes (An architecture for message exchange in Pervasive Healthcare based on the use of intelligent agents) In: Jornal Brasileiro de TeleSSaúde December 2013, Article. [Deposited 26 May 2014]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Cavalini, L.T. and do Prado, A.F. Uma Abordagem Para o

Desenvolvimento de Aplicações no Cuidado de Saúde Pervasivo Através do Uso de Arquétipos (An Approach to Developing Applications in the Pervasive Healthcare Environment through the use of Archetypes) In: Jornal Brasileiro de TeleSSaúde December 2013, Article. Item availability restricted. [Deposited 16 April 2014]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Cavalini, L.T. and Ferreira Pires, L. and do Prado, A.F. Uma Abordagem Para Troca De Mensagens Usando Arquétipos (An Approach for Message Exchange Using Archetypes In: Jornal Brasileiro de TeleSSaúde December 2013, Article. [Deposited 17 April 2014]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Cavalini, L.T. and do Prado, A.F. Using the dual-level modeling approach to developing applications in the Pervasive Healthcare environment In: Journal of Mobile Multimedia 30 November 2013, Article. Item availability restricted. [Deposited 26 February 2014]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Cavalini, L.T. and Ferreira Pires, L. and do Prado, A.F. An approach for message exchange using archetypes In: Proceedings of the 14th World Congress on Medical and Health Informatics, MEDINFO 2013 August 2013, Conference or Workshop Paper. [Deposited 17 December 2013]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and do Prado, A.F. Message generation facilities for interoperability in Pervasive Healthcare environments In: Proceedings of the 19th Americas Conference on Information Systems, AMCIS 2013 August 2013, Conference or Workshop Paper. [Deposited 29 September 2013]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and do Prado, A.F. Towards a reusable architecture for message exchange in Pervasive Healthcare In: Proceedings of the 15th International Conference on Enterprise Information Systems, ICEIS 2013 July 2013, Conference or Workshop Paper. [Deposited 26 February 2014]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Cavalini, L.T. and do Prado, A.F. A novel approach to developing applications in the Pervasive Healthcare environment through the use of archetypes In: 13th International Conference on Computational Science and Its Applications, ICCSA 2013 June 2013, Conference or Workshop Paper. [Deposited 03 December 2013]

Cardoso de Moraes, J.L. and Lopes de Souza, W. and Ferreira Pires, L. and Cavalini, L.T. and do Prado, A.F. A novel architecture for message exchange in Pervasive Healthcare based on the use of intelligent agents In: Proceedings of the 10th ACS/IEEE International Conference on Computer Systems and Applications, AICCSA 2013 May 2013, Conference or Workshop Paper. [Deposited 13 October 2013]

Federal University of São Carlos (2009 – 2012)

R. M. M. Braga, L. T. Cavalini, C. E. Cirilo, T. W. Cook, B. S. P. M. Correa, S. M. Freire, A. T. A. Gomes, V. M. Macedo, A. Menezes, Cardoso de Moraes, J.L., A. F. Prado, W. L. Souza, I. M. Teixeira, A. Ziviani, Model-driven development of healthcare applications, chapter in Scientific Computing Applied to Medicine and Healthcare, Raúl A. Feijóo, Artur Ziviani, Pablo J. Blanco (Eds.), LNCC/INCT-MACC, ISBN 978-85-99961-17-9, chapter 10, pp. 315 – 354, August 2012, Petrópolis (RJ).

Cardoso de Moraes, J.L., Menezes, A. L., Cirilo, C.E., Souza, W.L., Prado, A. F. Usando Arquétipos e Linguagem Específica de Domínio no Desenvolvimento de Aplicações Ubíquas para o Cuidado de Saúde Pervasivo In: XIII Congresso Brasileiro de Informática em Saúde (SBIS), ISSN 2178-2857, 2012, pp. 1-6, Curitiba (PR).

Cardoso de Moraes, J.L., Bossonaro, A.A., Menezes, A.L., Souza, W.L., PRADO, A. F. Uma Arquitetura Baseada em Agentes Inteligentes para a Troca de Mensagens Sensíveis ao Contexto para o Cuidado de Saúde Pervasivo In: Conferência IADIS Ibero-Americana WWW/Internet, 2011, ISBN 978-989-8533-03-6. IADIS Press, 2011. p.421 – 425. Rio de Janeiro (RJ).

Bossonaro, A. A.; Vilela, Mateus A.; Cardoso de Moraes, J.L.; Prado, A. F.; Araujo, Regina B. An Integrated System to Support Critical Infrastructure Security. In: 1ª Brazilian Conference on Critical Embedded Systems, EPUSP, 2011. v. 1. p. 73-78. São Carlos (SP).

Bossonaro, A. A.; Cardoso de Moraes, J.L.; Prado, A. F.; Souza, Wanderley Lopes; Araujo, Regina B. Um Sistema Ciber-Físico para Operações de Segurança de Infra-estruturas Críticas. In: Simpósio de Aplicações Operacionais em Áreas de Defesa - SIGE, 2011. v. 1. São José dos Campos (SP).

Menezes, A.L., Cirilo, C.E., Cardoso de Moraes, J.L., Prado, A. F., Souza, W.L. Using Archetypes and Domain Specific Languages on

Development of Ubiquitous Applications to Pervasive Healthcare In: Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems, 2010, ISBN 978-989-8533-03-6, DOI: 10.1109/CBMS.2010.6042677. v.01. p.395 – 400. Perth (AU).

Cardoso de Moraes, J.L.; Bossonaro, A. A.; Souza, Wanderley Lopes; Prado, A. F. Desenvolvimento de um framework de componentes para o domínio de Cardiologia e seu reuso em aplicações para dispositivos móveis, implementadas em Object Pascal. In: Congresso de Inovação com Tecnologias Open Source. I Workshop Open Source, 2010. v. 1. p. 29-38. Marília (SP).

Bossonaro, A. A.; Cardoso de Moraes, J.L.; Prado, A. F. Construção de Componentes a partir de Sistemas Legados. In: Congresso de Inovação com Tecnologias Open Source. I Workshop Open Source, 2010. v. 1. p. 49-58. Marília (SP).

Cardoso de Moraes, J.L., Prado, A. F., Souza, W.L. Ambiente de Computação Ubíqua para o Cuidado de Saúde Pervasivo (ACUCSP) In: XXIXX Congresso da Sociedade Brasileira de Computação, XXXVI Seminário Integrado de Software e Hardware (SEMISH), ISSN 2175-2761, 2009. p.217 – 231, Bento Goncalves (RS).

Appendix B

Questionnaires

During the development of this thesis, the author has applied the questionnaire based on Technology Acceptance Model (TAM) using the five points Likert Scale.

Technology Acceptance Model Questionnaire

Please fill out the following. Your comments will help us to design further improvements in our software system so your comments are very important. Thank you for your time!						
Age:						
Sex:						
Occupation:						
How do you rate your experience of using the technology?						
	LOW				HIGH	
	(1)	(2)	(3)	(4)	(5)	

Please fill in the number that represents how you feel about the software system you have been using:

1. My interaction with the system is clear and understandable.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

2. I find the system useful in my job.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

3. Using the system improves my performance in my job.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

4. Interacting with the system does not require a lot of my mental effort.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

5. Using the system increases my productivity in my job.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

6. I find easy to get from the system what I want it to do.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

7. Given that I have access to the system, I predict I will use the system.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

8. Using the system my tasks are facilitated.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

9. Assuming I have access to the system, I intend to use it.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

10. Using the system enhances my effectiveness in my job.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

11. I find the system is easy to use.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

12. Using the system in my job is a good idea.

(1) Strongly Agree	(2) Agree	(3) Neither	(4) Disagree	(5) Strongly Disagree

Healthcare Professionals – Delivery of Laboratory Analysis Results

Structured Questionnaire			Mean
Perceived Ease of Use	PEU1	My interaction with the system is clear and understandable.	3.31
	PEU2	Interacting with the system does not require a lot of my mental effort.	3.80
	PEU3	I find the system is easy to use.	3.70
	PEU4	I find easy to get from the system what I want it to do.	4.11
Perceived Usefulness	PU1	Using the system improves my performance in my job.	4.43
	PU2	Using the system increases my productivity in my job.	3.90
	PU3	Using the system enhances my effectiveness in my job.	3.93
	PU4	I find the system useful in my job.	4.24
	PU5	Using the system my tasks are facilitated.	4.45
Intention to Use	IU1	Assuming I have access to the system, I intend to use it.	4.58
	IU2	Given that I have access to the system, I predict I will use the system.	4.41
	IU3	Using the system in my job is a good idea.	4.27

Patients – Delivery of Laboratory Analysis Results

Structured Questionnaire			Mean
Perceived Ease of Use	PEU1	My interaction with the system is clear and understandable.	3.22
	PEU2	Interacting with the system does not require a lot of my mental effort.	3.10
	PEU3	I find the system is easy to use.	4.01
	PEU4	I find easy to get from the system what I want it to do.	3.93
Perceived Usefulness	PU1	Using the system improves my performance in my job.	3.51
	PU2	Using the system increases my productivity in my job.	3.66
	PU3	Using the system enhances my effectiveness in my job.	4.11
	PU4	I find the system useful in my job.	4.53
	PU5	Using the system my tasks are facilitated.	4.03
Intention to Use	IU1	Assuming I have access to the system, I intend to use it.	3.05
	IU2	Given that I have access to the system, I predict I will use the system.	3.96
	IU3	Using the system in my job is a good idea.	4.15

References

[1] Y.X. Shanshan Jiang, Annarita Giani, Ruzena Bajcsy, Robust Medical Data Delivery for Wireless Pervasive Healthcare, in: 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, 2009, pp. 802-807.

[2] B. Skinner, M. Rovere, Paying More, Getting Less: Measuring the Sustainability of Government Health Spending in Canada in, Fraser Institute, 2009, pp. 40.

[3] U. Hansmann, M.S. Nicklous, T. Stober, Pervasive Computing Handbook, Springer-Verlag New York, Inc., 2001.

[4] M. Weiser, Some Computer Science Issues in Ubiquitous Computing, Commun. ACM, 36 (1993) 75-84.

[5] E. Browne, openEHR Archetypes for HL7 CDA Documents, in, Ocean Informatics, 2008, pp. 47.

[6] M. Papazoglou, W.-J. Heuvel, Service oriented architectures: approaches, technologies and research issues, The VLDB Journal, 16 (2007) 389-415.

[7] M. Wooldridge, An Introduction to MultiAgent Systems, Wiley Publishing, 2009.

[8] F. Kart, L.E. Moser, P.M. Melliar-Smith, Building a Distributed E-Healthcare System Using SOA, IT Professional, 10 (2008) 24-30.

[9] C.E. Koop, R. Mosher, L. Kun, J. Geiling, E. Grigg, S. Long, C. Macedonia, R. Merrell, R. Satava, J. Rosen, Future delivery of health care: Cybercare, Engineering in Medicine and Biology Magazine, IEEE, 27 (2008) 29-38.

[10] J. Bisbal, D. Lawless, B. Wu, J. Grimson, Legacy information systems: Issues and directions, IEEE Software, 16 (1999) 103-111.

[11] G. Borriello, V. Stanford, C. Narayanaswami, W. Menning, Pervasive computing in healthcare, IEEE Pervasive Computing, 6 (2007) 17-19.

[12] A. Mihailidis, J.E. Bardram, Pervasive Computing in Healthcare, Taylor & Francis, 2006.

[13] J.E. Bardram, Pervasive Healthcare as a Scientific Discipline, Methods of Information in Medicine, 47 (2008) 178-185.

[14] M. Niazi, D. Wilson, D. Zowghi, A maturity model for the implementation of software process improvement: an empirical study, Journal of Systems and Software, 74 (2005) 155-172.

[15] R.N. Taylor, N. Medvidovic, E.M. Dashofy, Software Architecture: Foundations, Theory, and Practice, Wiley Publishing, 2009.

[16] R. Wieringa, Design science as nested problem solving, in: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, ACM, Philadelphia, Pennsylvania, 2009.

[17] R. Wieringa, Designing Requirements Engineering Research, in: Proceedings of the 2007 Fifth International Workshop on Comparative Evaluation in Requirements Engineering, IEEE Computer Society, 2007.

[18] K. Peffers, M. Rothenberger, B. Kuechler, R. Wieringa, A. Moralı, Technical Action Research as a Validation Method in Information Systems Design Science, in: Design Science Research in Information Systems. Advances in Theory and Practice, Springer Berlin Heidelberg, 2012, pp. 220-238.

[19] R. Wieringa, Design science methodology: principles and practice, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, ACM, Cape Town, South Africa, 2010.

[20] R. Wieringa, N. Condori-Fernandez, M. Daneva, B. Mutschler, O. Pastor, Lessons learned from evaluating a checklist for reporting experimental and observational research, in: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, ACM, Lund, Sweden, 2012.

[21] N. Cross, Engineering design methods: strategies for product design, Wiley, 2000.

[22] J. Robertson, Product design fundamentals and methods - Roozenburg,NFM, Eekels,J, Ergonomics, 39 (1996) 992-993.

[23] A.D. Hall, A Methodology for Systems-Engineering, Operations Research, 11 (1963) B31-B32.

[24] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for Information Systems Research, Journal of Management Information Systems, 24 (2007) 45-77.

[25] J.E. Bardram, Clinical Proof-of-Concept – A Evaluation Method for Pervasive Healthcare Systems, in: UbiComp '08 Workshop on Ubiquitous Systems Evaluation (USE '08), CEUR-WS, Seoul, Korea, 2008, pp. 6.

[26] F.D. Davis, Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly, 13 (1989) 319-340.

[27] HIMSS, Healthcare security and compliance guide, in, 2010.

[28] D.W. Bates, A.A. Gawande, Patient safety: Improving safety with information technology, New England Journal of Medicine, 348 (2003) 2526-2534.

[29] J.D. Kleinke, C.M. Christensen, J.H. Grossman, J. Hwang, The Innovator's Prescription: A Disruptive Solution for Health Care, Health Affairs, 28 (2009) 1223-1224.

[30] R.M. Mullner, K. Chung, Current Issues in Health Care Informatics, J. Med. Syst., 30 (2006) 1-2.

[31] J. Adler-Milstein, D.W. Bates, Paperless healthcare: Progress and challenges of an IT-enabled healthcare system, Business Horizons, 53 (2010) 119-130.

[32] P. Waterson, Y. Glenn, K. Eason, Preparing the ground for the 'paperless hospital': A case study of medical records management in a UK outpatient services department, International Journal of Medical Informatics, 81 (2012) 114-129.

[33] R.L. Nolan, Managing the computer resource: a stage hypothesis, Commun. ACM, 16 (1973) 399-405.

[34] J. Becker, R. Knackstedt, J. Poeppelbuss, Developing Maturity Models for IT Management - A Procedure Model and its Application, Business & Information Systems Engineering, 1 (2009) 213-+.

[35] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, M. Paulk, Software quality and the Capability Maturity Model, Communications of the Acm, 40 (1997) 30-40.

[36] R. van de Wetering, R. Batenburg, A PACS maturity model: a systematic meta-analytic review on maturation and evolvability of PACS in the hospital enterprise.

[37] B. Sharma, Electronic Healthcare Maturity model (eHMMM), in, Quintegra, Chenai, India, 2008, pp. 6.

[38] R. Fitterer, P. Rohner, Towards assessing the networkability of health care providers: a maturity model approach, Information Systems and E-Business Management, 8 (2010) 309-333.

[39] R. Haux, Health information systems – past, present, future, International Journal of Medical Informatics, 75 (2006) 268-281.

[40] P. Gottschalk, Maturity levels for interoperability in digital government, Government Information Quarterly, 26 (2009) 75-81.

[41] R.D. Galliers, A.R. Sutherland, Information systems management and strategy formulation: the 'stages of growth' model revisited, Information Systems Journal, 1 (1991) 89-114.

[42] M.A. Pešić, V.J. Milić, A. Anđelković, Business Process Management Maturity Model - Serbian Enterprises' Maturity Level, Serbian Association of Economists Journal, (2012) 190-198.

[43] C.P. Team, Capability Maturity Model Integration - CMMI for Development, Version 1.3, in, 2010.

[44] S.S.C. Shang, S.-F. Lin, Understanding the effectiveness of Capability Maturity Model Integration by examining the knowledge management of software development processes, Total Quality Management & Business Excellence, 20 (2009) 509-521.

[45] T. De Bruin, R. Freeze, U. Kaulkarni, M. Rosemann, Understanding the Main Phases of Developing a Maturity Assessment Model, in: Australasian Conference on Information Systems (ACIS), Australasian Chapter of the Association for Information Systems, Australia, New South Wales, Sydney, 2005.

[46] H.E. Essmann, Toward innovation capability maturity, in: Industrial Engineering, University of Stellenbosch, Stellenbosch, South Africa, 2009, pp. 219.

[47] P. Brooks, O. El-Gayar, S. Sarnikar, Towards a Business Intelligence Maturity Model for Healthcare, in: Proceedings of the 2013 46th Hawaii International Conference on System Sciences, IEEE Computer Society, 2013.

[48] C.P. Holland, B. Light, A stage maturity model for enterprise resource planning systems use, SIGMIS Database, 32 (2001) 34-45.

[49] E.P. Ambinder, Electronic Health Records, Journal of Oncology Practice, 1 (2005) 57-63.

[50] R. Hillestad, J. Bigelow, A. Bower, F. Girosi, R. Meili, R. Scoville, R. Taylor, Can Electronic Medical Record Systems Transform Health Care? Potential Health Benefits, Savings, And Costs, Health Affairs, 24 (2005) 1103-1117.

[51] D.I. Thompson, J. Osheroff, D. Classen, D.F. Sittig, A review of methods to estimate the benefits of electronic medical records in hospitals and the need for a national benefits database, Journal of healthcare information management : JHIM, 21 (2007) 62-68.

[52] W. Hersh, A. Wright, What workforce is needed to implement the health information technology agenda? Analysis from the HIMSS analytics database, AMIA Annual Symposium proceedings, (2008) 303-307.

[53] M.F. Furukawa, T.S. Raghu, B.B.M. Shao, Electronic Medical Records, Nurse Staffing, and Nurse-Sensitive Patient Outcomes: Evidence from California Hospitals, 1998-2007, Health Services Research, 45 (2010) 941-962.

[54] P.G. Shekelle, S.C. Morton, E.B. Keeler, Costs and benefits of health information technology, Evidence report/technology assessment, (2006) 1-71.

[55] C.L. Goldzweig, A. Towfigh, M. Maglione, P.G. Shekelle, Costs And Benefits Of Health Information Technology: New Trends From The Literature, Health Affairs, 28 (2009) W282-W293.

[56] J.M. Overhage, P.R. Dexter, S.M. Perkins, W.H. Cordell, J. McGoff, R. McGrath, C.J. McDonald, A Randomized, Controlled Trial of Clinical Information Shared From Another Institution, Annals of Emergency Medicine, 39 (2002) 14-23.

[57] D.G. Covell, G.C. Uman, P.R. Manning, Information Needs in Office Practice: Are They Being Met?, Annals of Internal Medicine, 103 (1985) 596-599.

[58] S.E. Ross, C.-T. Lin, The Effects of Promoting Patient Access to Medical Records: A Review, Journal of the American Medical Informatics Association, 10 (2003) 129-138.

[59] M. Simonov, L. Sammartino, M. Ancona, S. Pini, W. Cazzola, M. Frascio, Information, knowledge and interoperability for healthcare domain, First International Conference on Automated Production of Cross Media Content for Multi-channel Distribution, Proceedings, (2005) 35-42.

[60] O. Iroju, A. Soriyan, I. Gambo, J. Olaleke, Interoperability in Healthcare: Benefits, Challenges and Resolutions, International Journal of Innovation and Applied Studies, 03 (2013) 8.

[61] IEEE, Institute of Electrical and Electronics Engineers (IEEE) Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, in, IEEE, New York, 1990.

[62] M. Janssen, Y. Charalabibis, G. Kuk, T. Cresswell, Guest Editors' Introduction: E-government Interoperability, Infrastructure and Architecture: State-of-the-art and Challenges, Journal of theoretical and applied electronic commerce research, 6 (2011) I-VIII.

[63] G.A. Silver, K.R. Bellipady, J.A. Miller, K.J. Kochut, W. York, Supporting Interoperability Using the Discrete-Event Modeling Ontology (DeMO), in: Winter Simulation Conference 2009, Austin, TX, 2009, pp. 1379-1390.

[64] V.N. Stroetmann, D. Kalra, P. Lewalle, A. Rector, J.M. Rodrigues, K.A. Stroetmann, G. Surjan, B. Ustun, M. Virtanen, P.E. Zanstra, Semantic Interoperability for Better Health and Safer Healthcare, in, Deployment and Research Roadmap for Europe, 2009, pp. 34.

[65] K. Heubusch, Interoperability: what it means, why it matters, Journal of AHIMA / American Health Information Management Association, 77 (2006) 26-30; quiz 33-24.

[66] T. Beale, S. Heard, An Ontology-based Model of Clinical Information, Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics, Pts 1 and 2, 129 (2007) 760-764.

[67] A. Tolk, S. Diallo, C. Turnitsa, Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering, Systemics, Cybernetics and Informatics, 5 (2007) 65-74.

[68] P. Groen, M. Wine, Medical Semantics, Ontologies, Open Solutions and EHR Systems, in, Computer & Information Science Department Shepherd University - West Virginia, 2009.

[69] E. Ortiz, C.M. Clancy, Use of information technology to improve the quality of health care in the United States, Health Services Research, 38 (2003) XI-XXII.

[70] D.M. Lopez, B.G.M.E. Blobel, A development framework for semantically interoperable health information systems, International Journal of Medical Informatics, 78 (2009) 83-103.

[71] P. Johannesson, H. Fujita, Special issue on 'legacy systems and software change', Knowledge-Based Systems, 17 (2004) 237-238.

[72] P.P. Reid, W.D. Compton, J.H. Grossman, G. Fanjiang, Building a Better Delivery System: A New Engineering/Health Care Partnership, The National Academies Press, 2005.

[73] S.O. Zandieh, K. Yoon-Flannery, G.J. Kuperman, D.J. Langsam, D. Hyman, R. Kaushal, Challenges to EHR implementation in electronic-versus paper-based office practices, Journal of General Internal Medicine, 23 (2008) 755-761.

[74] P.Y. Mahon, D.M. Nickitas, K.M. Nokes, Faculty Perceptions of Student Documentation Skills During the Transition from Paper-Based to Electronic Health Records Systems, Journal of Nursing Education, 49 (2010) 615-621.

[75] M. Hubatsch, P. Kikeli, Z. Preg, C. Ionescu, P. Jakab, K.R. Kosa, Experiences with the Medprax model in Romania in the data acquirement from the primary health care system and its transfer from paper based patient files into electronic support, 2006.

[76] B. Arnrich, O. Mayora, J. Bardram, G. Tröster, Pervasive Healthcare. Paving the Way for a Pervasive, User-centered and Preventive Healthcare Model, Methods of Information in Medicine, 49 (2010) 67-73.

[77] J.E. Bardram, Pervasive Healthcare as a Scientific Discipline, Methods of Information in Medicine, 47 (2008) 178-186.

[78] U. Varshney, O. Mayora, Special issue on pervasive healthcare, Mobile Networks & Applications, 12 (2007) 111-112.

[79] Z. Zhu, X. Sui, F. Jiang, J. Li, J. Ye, A user-centric Pervasive Computing Services Model for medical and health-care, 2007.

[80] C. Orwat, A. Graefe, T. Faulwasser, Towards pervasive computing in health care - A literature review, BMC Medical Informatics and Decision Making, 8 (2008).

[81] P. Moderator, T. Znati, On the challenges and opportunities of pervasive and ubiquitous computing in health care, 2005.

[82] D. Hutter, G. Müller, W. Stephan, M. Ullmann, J. Bohn, F. Gärtner, H. Vogt, Dependability Issues of Pervasive Computing in a Healthcare Environment, in: Security in Pervasive Computing, Springer Berlin Heidelberg, 2004, pp. 53-70.

[83] W. Maass, U. Varshney, Design and evaluation of Ubiquitous Information Systems and use in healthcare, Decision Support Systems, 54 (2012) 597-609.

[84] J. Ye, S. Dobson, P. Nixon, An Overview of Pervasive Computing Systems, in: Ambient Intelligence with Microsystems, Springer US, 2009, pp. 3-17.

[85] J. Halloran, R. Iqbal, D. Aliakseyeu, M. Fernando, R. Cooper, A. Grzywaczewski, R. Grewal, A. James, C. Greenhalgh, Design Challenges and Solutions: Review of the 4th International Workshop on Ubiquitous Computing (iUBICOM 2009), Dataspace: the Final Frontier, Proceedings, 5588 (2009) 234-245.

[86] U. Varshney, Pervasive healthcare and wireless health monitoring, Mobile Networks & Applications, 12 (2007) 113-127.

[87] I. Korhonen, J. Parkka, M. Van Gils, Health monitoring in the home of the future, IEEE Engineering in Medicine and Biology Magazine, 22 (2003) 66-73.

[88] E. Kafeza, D.K.W. Chiu, S.C. Cheung, M. Kafeza, Alerts in mobile healthcare applications: Requirements and pilot study, IEEE Transactions on Information Technology in Biomedicine, 8 (2004) 173-181.

[89] E.C. Kyriacou, C.S. Pattichis, M.S. Pattichis, An Overview of Recent Health Care Support Systems for eEmergency and mHealth Applications, in: Annual International Conference of the IEEE-Engineering-in-Medicine-and-Biology-Society, Minneapolis, MN, 2009, pp. 1246-1249.

[90] U. Varshney, S. Sneha, Patient monitoring using ad hoc wireless networks: Reliability and power management, IEEE Communications Magazine, 44 (2006) 49-55.

[91] J. Luis Fernandez-Aleman, I. Carrion Senor, P.A. Oliver Lozoya, A. Toval, Security and privacy in electronic health records: A systematic

literature review, Journal of Biomedical Informatics, 46 (2013) 541-562.

[92] D. Vouyioukas, I. Maglogiannis, Communication Issues in Pervasive Healthcare Systems and Applications, in: Pervasive and Smart Technologies for Healthcare: Ubiquitous Methodologies and Tools, IGI Global, 2010, pp. 197-227.

[93] M. Weiser, Ubiquitous Computing, Computer, 26 (1993).

[94] A.K. Tripathi, Reflections on challenges to the goal of invisible computing, Ubiquity, (2005) 1-1.

[95] D. Nallaperuma, K. Lokuge, Cardiology Predictor: Cardiology Interpretations for Medical Diagnosis, Sri Lanka Journal of Bio-Medical Informatics, 2 (2012) 144-155.

[96] T.S.S. Genders, E.W. Steyerberg, H. Alkadhi, S. Leschka, L. Desbiolles, K. Nieman, T.W. Galema, W.B. Meijboom, N.R. Mollet, P.J. de Feyter, F. Cademartiri, E. Maffei, M. Dewey, E. Zimmermann, M. Laule, F. Pugliese, R. Barbagallo, V. Sinitsyn, J. Bogaert, K. Goetschalckx, U.J. Schoepf, G.W. Rowe, J.D. Schuijf, J.J. Bax, F.R. de Graaf, J. Knuuti, S. Kajander, C.A.G. van Mieghem, M.F.L. Meijs, M.J. Cramer, D. Gopalan, G. Feuchtner, G. Friedrich, G.P. Krestin, M.G.M. Hunink, C.A.D. Consortium, A clinical prediction rule for the diagnosis of coronary artery disease: validation, updating, and extension, European Heart Journal, 32 (2011) 1316-1330.

[97] D.R. Holmes, Jr., J.B. Rich, W.A. Zoghbi, M.J. Mack, The Heart Team of Cardiovascular Care, Journal of the American College of Cardiology, 61 (2013) 903-907.

[98] R. Onutu, A. Onofrei, Non-compliance in a large population of elderly patients with cardiovascular disease, European Journal of Medical Research, 15 (2010) 200-200.

[99] J. Gortmaker, M. Janssen, Business process orchestration in e-government: A gap analysis, Innovations Through Information Technology, Vols 1 and 2, (2004) 1243-1245.

[100] L. Molensky, W. Ketter, J. Collins, J. Bloemhof, H. van de Koppel, Business intelligence gap analysis: A user, supplier and academic perspective, Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business, (2010) 119-128.

[101] K. Lyytinen, Y. Yoo, Ubiquitous computing, Communicationsof the ACM, 45 (2002) 63.

[102] N.A.B. John, Expert Talk for Time Machine Session: Designing Calm Technology;as Refreshing as Taking a Walk in the Woods, in: 2012 IEEE International Conference on Multimedia and Expo, IEEE Computer Society, Los Alamitos, CA, USA, 2012, pp. 423-423.

[103] M. Friedewald, O. Raabe, Ubiquitous computing: An overview of technology impacts, Telematics and Informatics, 28 (2011) 55-65.

[104] E.H.L. Aarts, J.L. Encarna, True Visions: The Emergence of Ambient Intelligence (Frontiers Collection), Springer-Verlag New York, Inc., 2006.

[105] C.-C. Lin, R.-G. Lee, C.-C. Hsiao, A pervasive health monitoring service system based on ubiquitous network technology, International Journal of Medical Informatics, 77 (2008) 461-469.

[106] S.M. Lee, S.-H. Park, S.N. Yoon, T. Hwang, Ubiquitous computing and its effects on small businesses, Service Business, 2 (2008) 347-358.

[107] M. Rigby, Applying emergent ubiquitous technologies in health: The need to respond to new challenges of opportunity, expectation, and responsibility, International Journal of Medical Informatics, 76 (2007) S349-S352.

[108] C. Orwat, A. Rashid, C. Holtmann, M. Woelk, M. Scheermesser, H. Kosow, A. Graefe, Adopting Pervasive Computing for Routine Use in Healthcare, IEEE Pervasive Computing, 9 (2010) 64-71.

[109] I. Brown, A.A. Adams, The ethical challenges of ubiquitous healthcare, International Review of Information Ethics, 8 (2007) 53-60.

[110] B.N. Schilit, M.M. Theimer, DISSEMINATING ACTIVE MAP INFORMATION TO MOBILE HOSTS, Ieee Network, 8 (1994) 22-32.

[111] A.K. Dey, G.D. Abowd, D. Salber, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human-Computer Interaction, 16 (2001) 97-+.

[112] L. Barkhuus, A. Dey, Is context-aware computing taking control away from the user? Three levels of interactivity examined, Ubicomp 2003: Ubiquitous Computing, 2864 (2003) 149-156.

[113] C. Gu, Y. Zheng, A Context-Aware Method for Service Composition in Pervasive Computing Environments, Advances in Computer Science, Environment, Ecoinformatics, and Education, Pt 5, 218 (2011) 118-122.

[114] A.K. Dey, Providing architectural support for building context-aware applications, in, Georgia Institute of Technology, 2000, pp. 240.

[115] N. Kasaki, S. Kurabayashi, Y. Kiyoki, A geo-location context-aware mobile learning system with adaptive correlation computing methods, Ant 2012 and Mobiwis 2012, 10 (2012) 593-600.

[116] F. Andreini, F. Crisciani, C. Cicconetti, R. Mambrini, Ieee, Context-Aware Location in the Internet of Things, 2010 Ieee Globecom Workshops, (2010) 300-304.

[117] M.J. Barranco, J.M. Noguera, J. Castro, L. Martinez, A Context-Aware Mobile Recommender System Based on Location and Trajectory, Management Intelligent Systems, 171 (2012) 153-162.

[118] C.-D. Wang, M. Gao, X.-F. Wang, An 802.11-based location determination approach for context-aware system, Intelligent Computing in Signal Processing and Pattern Recognition, 345 (2006) 1-8.

[119] Y. Kaneko, K. Harumoto, S. Fukumura, S. Shimojo, S. Nishio, A location-based peer-to-peer network for context-aware services in a ubiquitous environment, IEEE Computer Society, Trento, IT, 2005.

[120] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, The anatomy of a context-aware application, Wireless Networks, 8 (2002) 187-197.

[121] Y. Dahl, 'You have a message here': Enhancing interpersonal communication in a hospital ward with location-based virtual notes, Methods of Information in Medicine, 45 (2006) 602-609.

[122] F. Delmastro, Pervasive communications in healthcare, Computer Communications, 35 (2012) 1284-1295.

[123] J. Luprano, J. Sola, S. Dasen, J.M. Koller, O. Chetelat, Combination of Body Sensor Networks and On-Body Signal Processing Algorithms: The practical case of MyHeart project, BSN 2006: International Workshop on Wearable and Implantable Body Sensor Networks, Proceedings, (2006) 76-79.

[124] K. Wac, R. Bults, B. van Beijnum, I. Widya, V. Jones, D. Konstantas, M. Vollenbroek-Hutten, H. Hermens, Mobile Patient Monitoring: the MobiHealth System, in: Annual International Conference of the IEEE-Engineering in Medicine and Biology Society, Minneapolis, MN, 2009, pp. 1238-1241.

[125] D.W. Curtis, E.J. Pino, J.M. Bailey, E.I. Shih, J. Waterman, S.A. Vinterbo, T.O. Stair, J.V. Guttag, R.A. Greenes, L. Ohno-Machado, SMART - An integrated wireless system for monitoring unattended patients, Journal of the American Medical Informatics Association, 15 (2008) 44-53.

[126] S. Idwan, S. Alramouni, M. Al-Adhaileh, A. Al-Khasawneh, Enhancing mobile advertising via Bluetooth technology, International Journal of Mobile Communications, 6 (2008) 587-597.

[127] C. Gomez, J. Oller, J. Paradells, Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology, Sensors, 12 (2012) 11734-11753.

[128] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, M. Gerla, Measuring Effective Capacity of IEEE 802.15.4 Beaconless Mode, in: IEEE Wireless Communications and Networking Conference, Las Vegas, NV, 2006, pp. 493-498.

[129] W. Lu, Y. Zhu, W. Liu, Y. Ou, Study on Bluetooth technology and its access to BACnet, ISTM 2007: 7th International Symposium on Test and Measurement, Vols 1-7, Conference Proceedings, (2007) 5931-5934.

[130] A. Metzner, P. Pepper, A self-healing protocol stack architecture for Bluetooth (R) wireless technology, Human Interaction with Machines, (2006) 23-34.

[131] W.u.R. Butt, S. Abbas, Study of Bluetooth Technology and its Impacts, in: AIC 2008 - Proceedings of the 8th conference on Applied Informatics and Communications, World Scientific and Engineering Academy and Society (WSEAS), Wisconsin, USA, 2008, pp. 124-133.

[132] J.M. Cabero, V. Molina, I. Urteaga, F. Liberal, J.L. Martin, Acquisition of human traces with Bluetooth technology: Challenges and proposals, Ad Hoc Networks, 12 (2014) 2-16.

[133] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, Personal and Ubiquitous Computing, 10 (2006) 255-268.

[134] M. Munoz-Organero, P.J. Munoz-Merino, C. Delgado Kloos, Using bluetooth to implement a pervasive indoor positioning system with minimal requirements at the application level, Mobile Information Systems, 8 (2012) 73-82.

[135] K. Morsi, Q. Gao, H.G. Xiong, Analysis and modelling of interference in Bluetooth device discovery, IET Communications, 5 (2011) 890-900.

[136] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, G.B. Laleci, A survey and analysis of Electronic Healthcare Record standards, ACM Computing Surveys, 37 (2005) 277-315.

[137] T. Beale, Archetypes: Constraint-based domain models for future-proof information systems, Eleventh OOPSLA Workshop on Behavioral Semantics: Serving the Customer, (2002) 16-32.

[138] R. Lenz, S. Miksch, M. Peleg, M. Reichert, D. Riaño, A. Teije, A. González-Ferrer, B. Verhees, J.-M. Verlinden, C. Marcos, Data Integration for Clinical Decision Support Based on openEHR Archetypes and HL7 Virtual Medical Record, in: Process Support and Knowledge Representation in Health Care, Springer Berlin Heidelberg, 2013, pp. 71-84.

[139] S. Hissam, B. Russo, M. Mendonça Neto, F. Kon, L. Cavalini, T. Cook, Health Informatics: The Relevance of Open Source and Multilevel Modeling, in: Open Source Systems: Grounding Research, Springer Berlin Heidelberg, 2011, pp. 338-347.

[140] L.T. Cavalini, T.W. Cook, Knowledge engineering of healthcare applications based on minimalist multilevel models, in: e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on, 2012, pp. 431-434.

[141] J. Gibbons, W. MacCaull, L. Cavalini, T. Cook, Use of XML Schema Definition for the Development of Semantically Interoperable Healthcare Applications, in: Foundations of Health Information Engineering and Systems, Springer Berlin Heidelberg, 2014, pp. 125-145.

[142] L. Schmitt, T. Falck, F. Wartena, D. Simons, Novel ISO/IEEE 11073 Standards for Personal Telehealth Systems Interoperability, in: High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP. Joint Workshop on, 2007, pp. 146-148.

[143] L. Michelsen, S.S. Pedersen, H.B. Tilma, S.K. Andersen, Comparing Different Approaches to Two-Level Modelling of Electronic Health Records, Connecting Medical Informatics and Bio-Informatics, 116 (2005) 113-118.

[144] T. Beale, Archetypes and the EHR, Advanced Health Telematics and Telemedicine: the Magdeburg Expert Summit Textbook, 96 (2003) 238-244.

[145] T. Beale, S. Heard, Archetype Definition Language, in, openEHR Foundation, 2007, pp. 47.

[146] C. Ma, H. Frankel, T. Beale, S. Heard, EHR Query Language (EQL) - A Query Language for Archetype-Based Health Records, in: K.A. Kuhn, J.R. Warren, T.Y. Leong (Eds.) Medinfo 2007: Proceedings of the 12th World Congress on Health, 2007, pp. 397-401.

[147] P. Fremantle, S. Weerawarana, R. Khalaf, Enterprise services, Communications of the Acm, 45 (2002) 77-82.

[148] T. Erl, SOA Principles of Service Design, Prentice Hall, 2009.

[149] M. Crasso, A. Zunino, M. Campo, A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures, Journal of Database Management, 22 (2011) 102-132.

[150] B. Upadhyaya, Y. Zou, H. Xiao, J. Ng, A. Lau, Migration of SOAP-based Services to RESTful Services, 13th IEEE International Symposium on Web Systems Evolution (WSE), (2011) 104-113.

[151] M. Papazoglou, What's in a service?, Act4soc 2008: Proceedings of the 2nd International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing, (2008) 3-3.

[152] R.T. Fielding, Architectural styles and the design of network-based software architectures, in, University of California, Irvine, 2000, pp. 162.

[153] C. Pautasso, O. Zimmermann, F. Leymann, Restful web services vs. 'big' web services: making the right architectural decision, in: Proceedings of the 17th international conference on World Wide Web, ACM, Beijing, China, 2008, pp. 805-814.

[154] S. Vinoski, REST Eye for the SOA Guy, Internet Computing, IEEE, 11 (2007) 82-84.

[155] F. Belqasmi, J. Singh, S.Y.B. Melhem, R.H. Glitho, SOAP-Based vs. RESTful Web Services A Case Study for Multimedia Conferencing, Ieee Internet Computing, 16 (2012) 54-63.

[156] C. Pautasso, RESTful Web service composition with BPEL for REST, Data & Knowledge Engineering, 68 (2009) 851-866.

[157] E. Wilde, C. Pautasso, P. Adamczyk, P. Smith, R. Johnson, M. Hafiz, REST and Web Services: In Theory and in Practice, in: REST: From Research to Practice, Springer New York, 2011, pp. 35-57.

[158] A.S. Rao, M.P. Georgeff, BDI-agents: from theory to practice, in: Proceedings of the First Intl. Conference on Multiagent Systems, 1995.

[159] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 2003.

[160] N.R. Jennings, Agent-oriented Software Engineering, Multiple Approaches to Intelligent Systems, Proceedings, 1611 (1999) 4-10.

[161] M.Q. Wang, M. Nowostawski, M. Purvis, Declarative agent programming support for a FIPA-compliant agent platform, Programming Multi-Agent Systems, 3862 (2006) 252-266.

[162] Y.-J. Yoon, D.-U. Kim, S.-J. Na, D.-R. Shin, Etri, Implementation of the converting module from FIPA SL to XML, 10th International Conference on Advanced Communication Technology: Innovations toward Future Networks and Services, Vols I-III (2008) 1150-1153.

[163] K.-X. Liu, Y.-Q. Feng, E-commerce oriented automated negotiation based on FIPA interaction protocol specification, Proceedings of 2007 International Conference on Machine Learning and Cybernetics, Vols 1-7, (2007) 3199-3204.

[164] A.S. Rao, M.P. Georgeff, Decision procedures for BDI logics, Journal of Logic and Computation, 8 (1998) 293-343.

[165] M. Bratman, Intention, Plans, and Practical Reason, Cambridge University Press, 1999.

[166] N. Howden, R. Ronnquist, A. Hodgson, A. Lucas, JACK Intelligent Agents - Summary of an Agent Infrastructure, in: Proceedings of the 5th ACM International Conference on Autonomous Agents, 2001.

[167] M.J. Huber, JAM: a BDI-theoretic mobile agent architecture, in: Proceedings of the third annual conference on Autonomous Agents, ACM, Seattle, Washington, USA, 1999.

[168] R. Bordini, J. Hübner, M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak Using Jason, J. Wiley, 2007.

[169] A. Pokahr, L. Braubach, W. Lamersdorf, A Gaol Deliberation Strategy for BDI Agent Systems, Multiagent System Technologies, Proceedings, 3550 (2005) 82-93.

[170] M. Ganzha, L.C. Jain, A. Pokahr, L. Braubach, K. Jander, The Jadex Project: Programming Model, in: Multiagent Systems and Applications, Springer Berlin Heidelberg, 2013, pp. 21-53.

[171] F. Bouchoul, M. Mostefai, Agent-services and mobile agents for an integrated HCIS, International Journal of Computer Integrated Manufacturing, 22 (2009) 458-471.

[172] G. Lanzola, L. Gatti, S. Falasconi, M. Stefanelli, A framework for building cooperative software agents in medical applications, Artificial Intelligence in Medicine, 16 (1999) 223-249.

[173] D. Allsopp, P. Beautement, C. Booth, M. Cusack, M. Kirton, M. Butler, D. De Roure, M. Leuschel, S. Lo Presti, Trusted software agents for pervasive healthcare environments, IEEE Pervasive Computing, 2 (2003) 96.

[174] C. Mazzi, P. Ganguly, M. Kidd, Healthcare applications based on software agents, in: V.L. Patel, R. Rogers, R. Haux (Eds.) Medinfo 2001: Proceedings of the 10th World Congress on Medical Informatics, Pts 1 and 2, 2001, pp. 136-140.

[175] K. Decker, J.J. Li, Ieee, Coordinated hospital patient scheduling, International Conference on Multi-Agent Systems, Proceedings, (1998) 104-111.

[176] J. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. Molina, V. Julian, R. Silveira, R. Unland, S. Giroux, R. Alonso, D. Tapia, G. Villarrubia, J. Paz, Agent Technology and Wireless Sensor Networks for Monitoring Patients in Residences and Their Homes, in: Highlights on Practical Applications of Agents and Multi-Agent Systems, Springer Berlin Heidelberg, 2013, pp. 417-428.

[177] L. Chung, J.C.S. do Prado Leite, On non-functional requirements in software engineering, in: Conceptual modeling: Foundations and applications, Springer, 2009, pp. 363-379.

[178] R.A. Krueger, The Future of Focus Groups, Qualitative Health Research, 5 (1995) 524-530.

[179] R.A. Krueger, Is it a focus group? Tips on how to tell, Journal of Wound Ostomy and Continence Nursing, 33 (2006) 363-366.

[180] R.A. Krueger, Analyzing focus group interviews, Journal of Wound Ostomy and Continence Nursing, 33 (2006) 478-481.

[181] J. Reynolds, S. Apple, A systematic approach to pacemaker assessment, AACN clinical issues, 12 (2001) 114-126.

[182] P.R. Roberts, Follow up and optimisation of cardiac pacing, Heart, 91 (2005) 1229-1234.

[183] C.-p. Lau, C.-w. Siu, Pacing technology: advances in pacing threshold management, Journal of Zhejiang University-Science B, 11 (2010) 634-638.

[184] N.M. van Hemel, Remote monitoring of implanted cardiac devices: a plea for a nationwide exploration, Netherlands Heart Journal, 17 (2009) 434-437.

[185] J.L. Elion, T. Becker, A. Keller, R. Simon, T. Sippel, H. Solomon, W. Weintraub, Ieee, Integrating the Healthcare Enterprise (IHE) interoperability for cardiology: Year 1 demonstration, Computers in Cardiology 2005, Vol 32, 32 (2005) 691-694.

[186] A. Leff, J.T. Rayfield, Web-application Development using the Model-View-Controller Design Pattern, Fifth IEEE International Enterprise Distributed Object Computing Conference, Proceedings, (2001) 118-127.

[187] C. Nock, Data Access Patterns: Database Interactions in Object-Oriented Applications, Prentice Hall Professional Technical Reference, 2003.

[188] D.T. Ross, Structured Analysis (SA): A Language for Communicating Ideas, Software Engineering, IEEE Transactions on, SE-3 (1977) 16-34.

[189] L. Spagnoli, I. Almeida, K. Becker, A.P. Blois, C. Werner, Adaptation and composition within component architecture specification, Reuse of Off-the-Shelf Components, Proceedings, 4039 (2006) 142-155.

[190] R.M.M. Braga, C.M.L. Werner, M. Mattoso, Odyssey-Search: A multi-agent system for component information search and retrieval, Journal of Systems and Software, 79 (2006) 204-215.

[191] M. Marcos, J.A. Maldonado, B. Martínez-Salvador, D. Boscá, M. Robles, Interoperability of clinical decision-support systems and electronic health records using archetypes: A case study in clinical trial eligibility, Journal of Biomedical Informatics, 46 (2013) 676-689.

[192] D. Alur, D. Malks, J. Crupi, Core J2EE Patterns: Best Practices and Design Strategies, Prentice Hall PTR, 2001.

[193] G. De Moor, D. Kalra, J. Devlies, J. Bellon, M. Foulon, J. VanDerLei, Certification of Electronic Health Record systems and the Importance of the Validation of Clinical Archetypes, Collaborative Patient Centred Ehealth, 141 (2008) 82-91.

[194] A. Tapuria, D. Kalra, S. Kobayashi, Contribution of Clinical Archetypes, and the Challenges, towards Achieving Semantic Interoperability for EHRs, Healthcare informatics research, 19 (2013) 286-292.

[195] K.B. Nelson, J.H. Ellenberg, Apgar Scores as Predictors of Chronic Neurologic Disability, Pediatrics, 68 (1981) 36-44.

[196] S. Garde, R. Chen, H. Leslie, T. Beale, I. McNicoll, S. Heard, Archetype-based knowledge management for semantic interoperability of electronic health records, Studies in health technology and informatics, 150 (2009) 1007-1011.

[197] G. Booch, Object-Oriented Development, IEEE Transactions on Software Engineering, 12 (1986) 211-221.

[198] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, IEEE Software, 16 (1999) 96.

[199] S. Garde, S. Heard, E.J.S. Hovenga, Archetypes in Electronic Health Records: Making the Case and Showing the Path for Domain Knowledge Governance, in: National Health Informatics Conference (13th : 2005 : Melbourne, Vic.), Health Informatics Society of Australia, Brunswick East, Vic., 2005.

[200] M.J. Field, K.N. Lohr, Guidelines for Clinical Practice:From Development to Use, The National Academies Press, 1992.

[201] G. Kupfer, D. Prince, Consensus-based standards development processes - Serving the needs of the environment and public-health community, Journal of Environmental Health, 65 (2002) 20-21.

[202] J.J. Gagnier, G. Kienle, D.G. Altman, D. Moher, H. Sox, D. Riley, C. Group, The CARE guidelines: consensus-based clinical case

reporting guideline development, Journal of medical case reports, 7 (2013) 223-223.

[203] R.d.A. Falbo, G. Guizzardi, K.C. Duarte, An ontological approach to domain engineering, in: Proceedings of the 14th international conference on Software engineering and knowledge engineering, ACM, Ischia, Italy, 2002.

[204] R. Nkambou, J. Bourdeau, R. Mizoguchi, A. Zouaq, A Survey of Domain Ontology Engineering: Methods and Tools, in: Advances in Intelligent Tutoring Systems, Springer Berlin Heidelberg, 2010, pp. 103-119.

[205] A.L. Rector, W.A. Nowlan, S. Kay, Foundations for an Electronic Medical Record, Methods of Information in Medicine, 30 (1991) 179-186.

[206] T. Beale, S. Heard, openEHR Architecture Overview Release 1.0.2, in, Ocean Informatics, 2008, pp. 79.

[207] R.E. Schantz, D.C. Schmidt, Middleware for Distributed Systems, in: Wiley Encyclopedia of Computer Science and Engineering, John Wiley & Sons, Inc., 2007.

[208] L. Richardson, S. Ruby, RESTful Web Services, First Edition edition ed., O'Reilly Media, 2007.

[209] A. Moreno, A. Valls, J. Bocio, Management of hospital teams for organ transplants using multi-agent systems, Artificial Intelligence in Medicine, Proceedings, 2101 (2001) 374-383.

[210] K.-C. Yeh, Y.C. Chen, C.C. Chen, R.-S. Chen, An RFID-Based Software Agent Framework on Pervasive Health Service, Proceedings of the 2009 2nd International Conference on Biomedical Engineering and Informatics, Vols 1-4, (2009) 2031-2034.

[211] M.N.K. Boulos, Q. Cai, J.A. Padget, G. Rushton, Using software agents to preserve individual health data confidentiality in micro-scale geographical analyses, Journal of Biomedical Informatics, 39 (2006) 160-170.

[212] A. Moreno, C. Garbay, Software agents in health care, Artificial Intelligence in Medicine, 27 (2003) 229-232.

[213] K.P. Logan, Prognostic software agents for machinery health monitoring, in: Aerospace Conference Proceedings, IEEE, 2003, pp. 3213-3225.

[214] D. Isern, D. Sanchez, A. Moreno, Agents applied in health care: A review, International Journal of Medical Informatics, 79 (2010) 145-166.

[215] N.T. Nguyen, G.S. Jo, Agent and Multi-Agent Systems: Technologies and Applications: Second KES International Symposium, KES-AMSTA 2008, Incheon, Korea, March 26-28, 2008, Proceedings, Springer, 2008.

[216] J. Cucurull, R. Marti, G. Navarro-Arribas, S. Robles, B. Overeinder, J. Borrel, Agent mobility architecture based on IEEE-FIPA standards, Computer Communications, 32 (2009) 712-729.

[217] B. Chandrasekaran, J.R. Josephson, V.R. Benjamins, What Are Ontologies, and Why Do We Need Them?, IEEE Intelligent Systems, 14 (1999) 20-26.

[218] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE: A software framework for developing multi-agent applications. Lessons learned, Information and Software Technology, 50 (2008) 10-21.

[219] G.M. Rajakaruna, K.D. Saminda, H.K.S.R.C. Kumara, P.A.P.R. Athukorala, W.J.L.N. Wijayaweera, A.S. Karunananda, Agent technology to model evolvable ontologies, in: 2011 6th International Conference on Industrial and Information Systems, ICIIS 2011 - Conference Proceedings, 2011, pp. 310-315.

[220] Y. Labrou, T. Finin, Y. Peng, Agent Communication Languages: The Current Landscape, IEEE Intelligent Systems & Their Applications, 14 (1999) 45-52.

[221] E. German, L. Sheremetov, An agent framework for processing FIPA-ACL messages based on interaction models, Agent-Oriented Software Engineering Viii, 4951 (2008) 88-102.

[222] C.D. Nguyen, A. Perini, P. Tonella, Experimental Evaluation of Ontology-Based Test Generation for Multi-agent Systems, Agent-Oriented Software Engineering Ix, 5386 (2009) 187-198.

[223] Z. Cui, Y. Li, J. Shepherdson, A framework for ACL message translation for information agents, Knowledge-Based Intelligent Information and Engineering Systems, Pt 1, Proceedings, 2773 (2003) 1380-1386.

[224] S. Li, M. Kokar, Agent Communication Language, in: Flexible Adaptation in Cognitive Radios, Springer New York, 2013, pp. 37-44.

[225] F. Bellifemine, G. Rimassa, Developing multi-agent systems with a FIPA-compliant agent framework, Softw. Pract. Exper., 31 (2001) 103-128.

[226] J. Carbo, J.M. Molina, A JADE-based ART-inspired ontology and protocols for handling Trust and Reputation, 9th International Conference on Intelligent Systems Design and Applications, (2009) 300-305.

[227] C. Castelfranchi, Y. Lespérance, F. Bellifemine, A. Poggi, G. Rimassa, Developing Multi-agent Systems with JADE, in: Intelligent Agents VII Agent Theories Architectures and Languages, Springer Berlin Heidelberg, 2001, pp. 89-103.

[228] R. Fikes, Ontologies: What are they, and where's the research?, Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (Kr '96), (1996) 652-655.

[229] Y. Labrou, Standardizing agent communication, Multi-Agent Systems and Applications, 2086 (2001) 74-97.

[230] Q. Wang, C. Ren, S. Bing, J. Dong, H. Ding, W. Wang, SCPM: Facilitating Process-centric Deployment in Enterprise System Implementation, in: IEEE International Conference on e-Business Engineering, Xian, PEOPLES R CHINA, 2008, pp. 461-466.

[231] M. Hall-May, T. Kelly, Using agent-based modelling approaches to support the development of safety policy for systems of systems, Computer Safety, Reliabiltiy, and Security, Proceedings, 4166 (2006) 330-343.

[232] A. Tolk, L.C. Jain, Intelligence-Based Systems Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[233] S. Bandini, S. Manzoni, G. Vizzari, Agent Based Modeling and Simulation: An Informatics Perspective, Jasss-the Journal of Artificial Societies and Social Simulation, 12 (2009) A51-A66.

[234] L. Iocchi, D. Nardi, M. Salerno, Reactivity and Deliberation: A Survey on Multi-Robot Systems, in: Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Springer Berlin Heidelberg, 2001, pp. 9-32.

[235] E.A.M.P. Feigenbaum, The fifth generation: artificial intelligence and Japan's computer challenge to the world, Addison-Wesley, Reading, Mass., 1983.

[236] D. Sharma, F. Shadabi, An intelligent multi agent design in healthcare management system, in: N.T. Nguyen, G.S. Jo, R.J. Howlett, L.C. Jain (Eds.) Agent and Multi-Agent Systems: Technologies and Applications, Proceedings, 2008, pp. 674-682.

[237] B.B. Kawulich, Participant Observation as a Data Collection Method, 2005.

[238] F.D. Davis, Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Q., 13 (1989) 319-340.

[239] P.Y.K. Chau, P.J.H. Hu, Investigating Healthcare Professionals' Decisions to Accept Telemedicine Technology: an Empirical Test of Competing Theories, Information & Management, 39 (2002) 297-311.

[240] D.A. Adams, R.R. Nelson, P.A. Todd, Perceived Usefulness, Ease of Use, and Usage of Information Technology - A Replication, MIS Quarterly, 16 (1992) 227-247.

[241] J.-H. Wu, W.-S. Shen, L.-M. Lin, R.A. Greenes, D.W. Bates, Testing the technology acceptance model for evaluating healthcare professionals' intention to use an adverse event reporting system, International Journal for Quality in Health Care, 20 (2008) 123-129.

[242] A.-E. Bennani, R. Oumlil, Review of ten years relevance of Technology Acceptance Model in healthcare context, 2010.

[243] N.A.K.S. Mohamadali, J.M. Garibaldi, A Novel Evaluation Model of User Acceptance of Software Technology in Healthcare Sector, in: A. Fred, J. Filipe, H. Gamboa (Eds.) Healthinf 2010: Proceedings of the Third International Conference on Health Informatics, Valencia, SPAIN, 2010, pp. 392-397.

[244] F.-Y. Pai, K.-I. Huang, Applying the Technology Acceptance Model to the introduction of healthcare information systems, Technological Forecasting and Social Change, 78 (2011) 650-660.

[245] M.P. Gagnon, E. Orruno, J. Asua, A. Ben Abdeljelil, J. Emparanza, Using a Modified Technology Acceptance Model to Evaluate Healthcare Professionals' Adoption of a New Telemonitoring System, Telemedicine and E-Health, 18 (2012) 54-59.

[246] Q. Ma, L. Liu, The Technology Acceptance Model: A Meta-Analysis of Empirical Findings, Journal of Organizational and End User Computing (JOEUC), 16 (2004) 59-72.

[247] Y.-H. Chuo, C.-H. Tsai, Y.-L. Lan, C.-S. Tsai, The effect of organizational support, self efficacy, and computer anxiety on the usage intention of e-learning system in hospital, African Journal of Business Management, 5 (2011) 5518-5523.

[248] M. Igbaria, J. Iivari, The effects of self-efficacy on computer usage, Omega, 23 (1995) 587-605.

[249] V. Venkatesh, F.D. Davis, A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, Management Science, 46 (2000) 186-204.

[250] R. Likert, A Technique for the Measurement of Attitudes, [s.n.], New York, 1932.

[251] J.C. Nunnally, Psychometric Theory, Current Contents/Social & Behavioral Sciences, (1979).

[252] J. Henseler, W.W. Chin, A Comparison of Approaches for the Analysis of Interaction Effects Between Latent Variables Using Partial Least Squares Path Modeling, Structural Equation Modeling-a Multidisciplinary Journal, 17 (2010) 82-109.

[253] R.F. Goldin, Review: Statistical Models-Theory and Practice, The American Mathematical Monthly, 117 (2010) 844-847.

[254] R.J. Holden, B.-T. Karsh, The Technology Acceptance Model: Its past and its future in health care, Journal of Biomedical Informatics, 43 (2010) 159-172.

[255] F.D. Davis, V. Venkatesh, Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management, IEEE Transactions on Engineering Management, 51 (2004) 31-46.

[256] J.E. Bardram, H.B. Christensen, Pervasive computing support for hospitals: An overview of the activity-based computing project, IEEE Pervasive Computing, 6 (2007) 44-51.

[257] J.E. Bardram, Activity-Based Computing for Medical Work in Hospitals, ACM Transactions on Computer-Human Interaction, 16 (2009) 36.

[258] J. Kjeldskov, M.B. Skov, Exploring context-awareness for ubiquitous computing in the healthcare domain, Personal and Ubiquitous Computing, 11 (2007) 549-562.

[259] M.A. Munoz, M. Rodriguez, J. Favela, A.I. Martinez-Garcia, V.M. Gonzalez, Context-aware Mobile Communication in Hospitals, Computer, 36 (2003) 38-46.

[260] M.R. Santos, M.P. Bax, D. Kalra, Building a logical EHR architecture based on ISO 13606 standard and semantic web technologies, Studies in Health Technology and Informatics, (2010) 6.

[261] R.D.M. Dias, T.W. Cook, S.M. Freire, Modeling healthcare authorization and claim submissions using the openEHR dual-model approach, Bmc Medical Informatics and Decision Making, 11 (2011).

[262] R. Chen, G.O. Klein, E. Sundvall, D. Karlsson, H. Ahlfeldt, Archetype-based conversion of EHR content models: pilot experience with a regional EHR system, BMC Medical Informatics and Decision Making, 9 (2009) 13.

[263] J. Lahteenmaki, J. Leppanen, H. Kaijanranta, Interoperability of Personal Health Records, Annual International Conference of the IEEE Engineering in Medicine and Biology Society., (2009) 1726-1729.

[264] W.D. Yu, A. Joshi, J. Patel, V. Mehta, A SOA Service Governance Approach to u-Healthcare System with Mobility Capability, in: IEEE International Conference on Communications (ICC), Kyoto, JAPAN, 2011, pp. 6.

[265] R. Kazi, R. Deters, RESTful dissemination of healthcare data in mobile digital ecosystem, in: 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST), Menlo Park, CA, 2013, pp. 78-83.

[266] I. Baffo, G. Stecca, T. Kaihara, A Multi Agent System Approach for Hospital's Drugs Management using Combinatorial Auctions, in: 8th IEEE International Conference on Industrial Informatics (INDIN), Osaka, 2010, pp. 945-949.

[267] R. Annicchiarico, U. Cortés, C. Urdiales, F. Campana, A. Moreno, D. Riaño, L. Varga, K4Care: Knowledge-Based Homecare e-Services for an Ageing Europe, in: Agent Technology and e-Health, Birkhäuser Basel, 2008, pp. 95-115.

[268] D. Isern, A. Moreno, D. Sanchez, A. Hajnal, G. Pedone, L.Z. Varga, Agent-based execution of personalised home care treatments, Applied Intelligence, 34 (2011) 155-180.

[269] M. Batet, D. Isern, L. Marin, S. Martinez, A. Moreno, D. Sanchez, A. Valls, K. Gibert, Knowledge-driven delivery of home care services, Journal of Intelligent Information Systems, 38 (2012) 95-130.

[270] V. Urovi, A.C. Olivieri, S. Bromuri, N. Fornara, M.I. Schumacher, A peer to peer agent coordination framework for IHE based cross-community health record exchange, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, ACM, Coimbra, Portugal, 2013, pp. 1355-1362.

[271] R.M. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in BPMN, Information and Software Technology, 50 (2008) 1281-1294.

[272] G. Decker, A. Grosskopf, A. Barros, A graphical notation for modeling complex events in business processes, in: 11th International Enterprise Distributed Object Computing (EDOC), IEEE, Annapolis, MD, 2007, pp. 27.