An Architecture-Centric and Ontology-Based Approach to Cross-Domain Interoperability of Health Information Systems for Diabetes Care

GUSTAVO ANDRÉS URIBE GÓMEZ

Tesis de Doctorado en Ingeniería Telemática

Directores:
PhD. Diego Mauricio López, Universidad del Cauca, Colombia
PhD. Bernd Blobel, Universidad de Regensburg, Germany

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación en eSalud
Popayán, Septiembre de 2015

An Architecture-Centric and Ontology-Based Approach to Cross-Domain Interoperability of Health Information Systems for Diabetes Care

GUSTAVO ANDRÉS URIBE GÓMEZ

Doctoral Thesis in Telematics Engineering

Supervisors:
Diego Mauricio López PhD, University of Cauca, Colombia
Bernd Blobel PhD, University of Regensburg, Germany

University of Cauca
Electronics and Telecommunication Engineering Faculty
Telematics Department
eHealth Research Area
Popayán, September, 2015

GUSTAVO ANDRÉS URIBE GÓMEZ

AN ARCHITECTURE-CENTRIC AND ONTOLOGY-BASED APPROACH TO CROSS-DOMAIN INTEROPERABILITY OF HEALTH INFORMATION SYSTEMS FOR DIABETES CARE

Thesis presented to the Electronics and Telecommunication Engineering Faculty of the University of Cauca for obtaining the title of:

Doctor in:
Telematics Engineering

Supervisors:
Diego Mauricio López PhD, University of Cauca, Colombia
Bernd Blobel PhD, University of Regensburg, Germany

Popayán
2015

to the Creator, who gives me all

to my dear wife, who gives me her unconditional love

to my children, who give me hope.

Acknowledgments

I want to give my acknowledgements to my supervisor, Professor Diego López for his work not only as supervisor but also as a mentor. Thanks to my co-supervisor Professor Bernd Blobel for the education and training provided and for essential contributions to the research work. Thanks to the kind collaboration of the medical doctor Alonso Ruiz. He delivered his knowledge and experience to support the system. Thanks to Stefan Schultz and his team at Medical University of Graz, Austria for the support, as well as to the University of Magdeburg in Germany, for hosting one of my research stays. Further, study stays have been realized at the University of Regensburg, Germany, so I'm grateful. Thanks to the University of Cauca, especially the Telematics Engineer Group for their financial and administrative support. Finally, thanks to Colciencias for their financial support under grant entitled “Crédito Educativo Condonable Programa Nacional de Formación de Investigadores 2011”. This work could not be possible without any of the aforementioned persons and institutions, therefore my sincere and heartfelt thanks to them.

Resumen Estructurado

Problema

El presente trabajo de doctorado se centra en mejorar la colaboración entre los distintos actores involucrados en el cuidado de la salud de las personas, específicamente pacientes que sufren diabetes mellitus tipo 2. Esta es una enfermedad crónica que causa más de 4.9 millones de muertes anuales y altos costos para su atención. Se estima que el 7% de la población Colombiana mayor a 30 años está afectada por esa enfermedad. Una característica importante de este problema de salud es que requiere de un cuidado multidisciplinar, debido a que la enfermedad es afectada por diferentes aspectos de los estilos de vida de las personas y dado a que deriva en múltiples complicaciones manejadas por diferentes especialistas médicos. Otra característica importante es que se requiere una colaboración activa por parte del paciente; este debe realizar el auto-cuidado y cambiar apropiadamente su estilo de vida.

Desde el punto de vista computacional, resolver la falta de interoperabilidad entre los sistemas informáticos es crucial. Especialmente, la interoperabilidad entre los registros de salud electrónicos (historia clínicas electrónicas) de los proveedores de salud, con los sistemas de registros electrónicos personales de salud (PHR) que manejan los pacientes para su auto-cuidado. Los sistemas PHR permiten a los pacientes manejar la información personal que es relevante para su salud (ej. ingesta diaria de calorías, actividad física, niveles diarios de glucosa en sangre). Existen muchos trabajos que intentan solucionar los problemas de interoperabilidad a nivele técnico, pero usualmente la falta de interoperabilidad se pretende solucionar ignorando las diferencias entre los actores y sin considerar la semántica de la información que se está transmitiendo. Este trabajo pretende considerar estos aspectos y así lograr una interoperabilidad a un nivel más amplio.

Metodología

La propuesta de solución está basada en el modelo genérico de componentes, el cual es un marco conceptual y metodológico que permite describir cualquier tipo de sistema mediante tres dimensiones. La primera dimensión separa los diferentes dominios de conocimiento permitiendo diferentes perspectivas sobre el sistema, la segunda dimensión considera la descomposición del sistema en sus partes, la tercera dimensión representa las diferentes vistas consideradas en el desarrollo del software. La primera dimensión realiza la separación entre dominios, representado por las diversas ontologías existentes (ej. SNOMED-CT). La segunda dimensión considera cuatro niveles de granularidad del sistema, ocultando en principio los detalles y posteriormente describiendo solo hasta el nivel deseado de complejidad. La tercera dimensión reúsa las vistas definidas por el estándar RM-ODP y añade una descripción del sistema independiente de la computación. Usando las dimensiones descritas y los principios del GCM se realiza un modelo del sistema (arquitectura). Los diagramas en cuboide del GCM son complementado mediante diagramas UML y BPMN. Posteriormente, se formaliza la descripción por medio de los lenguajes OWL y SPIN con el fin de hacerla interpretable por una máquina. Basados en ese modelo se especifica y desarrolla el sistema informático que da soporte a la colaboración de los diferentes actores en el cuidado de la diabetes.

Resultados

Inicialmente se realizó una descripción genérica del sistema y posteriormente se especializó para ciertos casos de uso importantes dentro del cuidado de la diabetes. La descripción o arquitectura del sistema se realizó usando los siguientes diagramas y lenguajes:

1. Diagramas GCM: Estos diagramas en bloques representan los dominios del sistema, así como los diferentes niveles de granularidad.

2. Diagramas UML: Estos diagramas representan de manera explícita las relaciones estáticas o estructurales entre los componentes del sistema.

3. Diagramas BPMN: Estos diagramas representan explícitamente los aspectos dinámicos del sistema.

4. Ontología en OWL: Es usado para formalizar los aspectos representados en los diagramas GCM y UML.

5. Reglas en el lenguaje SPIN: Estas formalizan las reglas que gobiernan el comportamiento del sistema y permiten la definición de políticas.

Un método combinando los principios de MDA, la Web Semántica y la descripción de procesos de negocio fue propuesto, en el cual se implementan los principios del GCM en una solución software. Este método soluciona algunos problemas presentes en los procesos tradicionales de desarrollo y ayuda en la solución de sistemas de alta calidad. Usando estos métodos, se desarrolló un prototipo de sistema informático que soporta el caso de uso de control glucémico. El sistema demostró ser flexible, adaptable, inteligente y permitía la interoperabilidad.

Discusión

El modelado y desarrollo de sistemas para la salud usando las metodologías tradicionales impide la fácil creación de sistemas que soporten la interoperabilidad de los diversos actores. El problema es que se omite la descripción formal del dominio y no se sigue una aproximación arquitectónica que garantiza un modelado de alta calidad. Adicionalmente, la descripción formal permite realizar inferencias lógicas que son de gran utilidad en la creación de sistemas para el soporte a la toma de decisiones.

Structured Abstract

Problem

This doctoral work focuses on the improvement of collaboration between the different actors involved in the care of patients suffering type 2 diabetes mellitus. This chronic disease causes more than 4,9 million deaths each year and high costs to health systems. Around 7% of the Colombian population older than 30 years is affected by this disease. An important characteristic of this health problem is that it requires a multi-disciplinary care team. This is because of the different aspect of life style of the persons impacting that disease and the multiple complications managed by different medical specialties. Another important aspect is that the need of active collaboration of the patient, performing self-care and changing his life style.

From the computational perspective, information systems interoperability is a crucial challenge. Specially, interoperability between Personal Health Record systems (PHRs) and Electronic Health Record systems (EHRs) is required. PHRs are managed by the patient during self-care and allow managing information about his health (e.g. calorie ingests, physical activity, glucose levels). There is a lot of research trying to solve the interoperability problem from technical perspectives, but ignoring the actor differences and the information semantics. This work intends to consider these aspects and to achieve interoperability at a more comprehensive level.

Methods

The proposed solution is based on the Generic Component Model, a conceptual and methodological framework that allows to describing any system using three dimensions. The first dimension separates the different knowledge domains enabling different perspectives on the system, the second dimension considers the decomposition of the system into its parts, and the last dimension represents the different views in the software development process. The first dimension realizes the

separation of domains, represented by existing ontologies (e.g. SNOMED-CT). The second dimension considers four levels of granularity of the system, hiding the details in the beginning and describing afterwards the desired level of the complexity. The third dimension reuses the RM-ODP standard views and adds an ICT independent description. Using these dimensions and the GCM principles, a model (architecture) of the system was build. The cuboid diagrams of the GCM are complemented using UML and BPMN diagrams. Later, with the goal of obtaining a machine interpretable model, the description is formalized through the OWL and SPIN languages. Based on this description, a prototype software solution supporting the actor collaboration in the type 2 diabetes care has been specified and implemented.

Results

First, a generic description (architecture) of the system was provided. Specialized architectures for important use cases of the diabetes care were derived from the generic architecture. The architectures were described using the following diagrams and languages:

1. GCM Diagrams: These block diagrams represent the domains in the system and the different granularity levels.
2. UML Diagrams: They represent explicitly the static or structural relationships between the components.
3. BPMN Diagrams: They represent explicitly the behavioral aspects of the system.
4. OWL Ontology: It is used to formalize the aspects represented in GCM and UML diagrams.
5. SPIN language rules: They formalize the rules governing the behavior of the system and allow the definition of policies.

A method combining principles of the MDA, the Semantic Web and the Business Process description was proposed, to implement the principles of the GCM in a software solution. This method solves some problems present in traditional development processes and helps to build high quality systems. Using the described methodology, a prototype of a software system for the glycemic control use case was developed. The system demonstrates flexibility, adaptability, intelligence, and supports interoperability.

Discussion

Modelling and developing health systems using traditional methodologies prevents from the easy creation of systems, which support interoperability between heterogeneous actors. The problem is that formal description of domain knowledge is omitted and the models do not follow an architectural approach. The architectural approach presented in this thesis helps to build high quality models. Furthermore, the formal description allows to performing logic inferences that are useful in the creation of decision support systems.

Content

Figure List

Index of Tables

Acronym List

AI	Artificial Intelligence
ATL	Atlas Transform Language
BFO	Basic Formal Ontology
BPMN	Business Process Modeling Notation
btl2	BioTop Lite Ontology
CAD	Computer-aided design
CIM	Computation Independent Model
DICOM	Digital Imaging and Communications in Medicine
DL	Description Logics
dm2co	Type 2 Diabetes Mellitus Care Ontology
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DRL	Drools Rule Languages
ECLIF	Extended Common Logic Interchange Format
EDI	Electronic Data Interchange
EHR	Electronic Health Record
GCM	Generic Component Model
GFO	General Formal Ontology
GLIF	Guideline Interchange Format
GST	General System Theory
HbA1C	Glycated Hemoglobin
HL7	Health Level Seven
ICD10	International Classification of Diseases

ICT	Information and Communications Technology
IHE	Integrating the Healthcare Enterprise
IHTSDO	International Health Terminology Standards Development Organisation
ILO	International Labor Organization
ISCO	International Standard Classification of Occupations
ISO	International Organization for Standardization
LIS	Laboratory Information Systems
LOINC	Logical Observation Identifiers Names and Codes
MDA	Model Driven Architecture
MedDRA	Medical Dictionary for Regulatory Activities
OBO	Open Biomedical Ontologies
OCL	Object Constraint Language
OMG	Object Management Group
OpenEHR	Open Electronic Health Record
OWL	Web Ontology Language
PACS	Picture Archiving and Communication System
PHR	Personal Health Record
PIM	Platform Independent Model
PSL	Process Specification Language
PSM	Platform Specific Model
QVTL	Query/View/Transformation Language
RDF	Resource Description Framework
RDFS	RDF Schema
REST	Representational State Transfer
RIF	Rule Interchange Format
RM-ODP	Reference Model - Open Distributed Processing
SBO	Semantic Bridge Ontology
SGSSS	Sistema General de Seguridad Social en Salud (General System of the Social Security in Health)
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms

SOA	Service Oriented Architecture
SPIN	SPARQL Inference Notation
SUMO	Suggested Upper Merged Ontology
SWRL	Semantic Web Rules Language
T2DM	Type 2 Diabetes Mellitus
UML	Unified Modeling Language
UP	Unified Process
WHO	World Health Organization

Chapter 1

Introduction

In this chapter, the problem of type 2 diabetes mellitus (T2DM) and the principles of improving T2DM care through interoperability of actors involved are described. For a better understanding, some basic concepts are described. Furthermore, the objectives, methods and related works of the proposed solution are presented.

1.1 Problem Definition

More than 347 million people around the world suffer from diabetes mellitus. In 2004, estimated 3.4 million people died from consequences of high fasting blood sugar. More than 80% of deaths caused by diabetes occurred in low- and middle-income countries [1].

The National Library of Medicine defines diabetes (mellitus) as follows: “Diabetes is usually a lifelong (chronic) disease in which there is a high level of sugar in the blood” [2]. The World Health Organization (WHO) describes diabetes as a chronic disease that either occurs when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces [1]. Hyperglycemia, or increased blood sugar, is a common effect of uncontrolled diabetes and can over time lead to serious damage of several organs and body systems, especially nerves and blood vessels, but also kidneys, eyes, and feet. Two main types of diabetes mellitus exist. Type 1 refers to insulin-dependent patients (usually starting in childhood), and type 2 refers to patients that do not depend on insulin. Type 2 is far more prevalent, representing 90% of people with diabetes around the world, and is largely the result of obesity due to wrong nutritional habits and physical inactivity.

A diabetes care system is characterized by the collaboration and interaction between many human actors and organizations, information systems and medical devices. An example for a complex diabetes care system is shown in Figure 1.1.

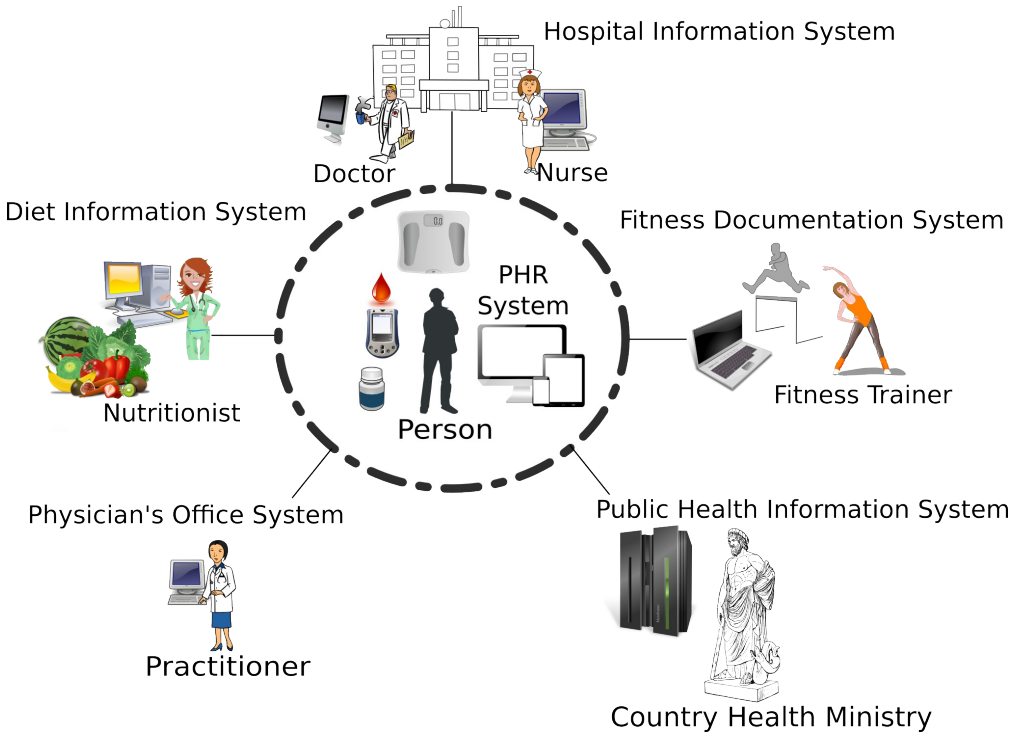

Figure 1.1: Diabetes care system

Each healthcare provider organization involved in T2DM is more or less specialized, using domain specific knowledge and terminologies, practicing specific methodologies and following specific policies, furthermore deploying specific devices and software systems. The most challenging part, however, is the collaboration between humans because of their different capabilities in terms of languages, knowledge domains, education, experiences, cultural backgrounds and views. For establishing interoperability, these differences have to be overcome either by standards, regulations and policies and their enforcement, or by harmonizing environment and context through intelligent technologies.

Interoperability requires the sharing of knowledge needed to perform intended cooperation as introduced in [3], [4]. Knowledge not shared a-priori must be communicated at runtime [3]. Depending on the knowledge missed, actors in a system need different levels of interoperability to achieve cooperation [3]. Syntactical

interoperability enables the interchange of data using common messages, vocabularies or clinical documents. Semantic interoperability enables the common interpretation of data towards information (understanding) by harmonizing the data models, terminologies or ontologies amongst the actors. Therefore, semantic interoperability is only possible between knowledge-based systems and implies sharing of knowledge. Service interoperability enables the performance of actions based on the information provided. If a-priori sharing of corresponding knowledge and skills is guaranteed, the lower level of interoperability is sufficient to enable comprehensive interoperability [3]. The aforementioned interoperability levels can be performed directly by the actors in simple systems. However, in complex systems requiring high level of knowledge, flexibility and adaptability, like T2DM, computer systems are necessary to enable interoperability.

Currently, standards such as those proposed by HL7, OpenEHR, IHE, ISO, OMG, IHTSDO and DICOM provide good solutions for syntactical interoperability, and also support semantic interoperability and service interoperability. However, advanced semantic and service interoperability is still a matter of research and development [5]–[8].

1.2 Research Question

With the purpose to contribute to the health interoperability problem in the context of Type 2 Diabetes Mellitus management, the following research question is proposed:

How to achieve cross-domain interoperability in health informatics systems for supporting Type 2 Diabetes Mellitus care?

1.3 Hypothesis

By using an architectural-centric approach to analyze, design and implement health information systems based on the Generic Component Model (GCM) and representing the components through ontologies it is possible to achieve cross-domain interoperability of health information systems supporting the diabetes care.

For a better understanding of the GCM Framework, the proposed hypothesis and its graphical representation, please refer to section 1.7.1.

1.4 Basic Concepts

1.4.1 Interoperability

Software systems interoperability is a long lasting challenge because software systems are still created in isolation by different vendors, from different perspectives and without following a common process. Overcoming this problem is particularly complex, especially in a heterogeneous and multidisciplinary environment like healthcare, because each medical specialty manages its own vocabulary and knowledge.

Interoperability is defined as a relation between/among objects, a mutual capability necessary to ensure successful and efficient interoperation, supporting cooperation [9]. In practice, interoperability describes successful collaboration between actors to achieve a common business goal [10]. For achieving interoperability through Electronic Health Record (EHR) systems some requirements need to be fulfilled. Blobel [11] presents a list of desired features of an EHR system architecture to provide interoperability. Those features are: openness, scalability, flexibility, portability, distribution, standard-conformance, interoperability at appropriate level, service-orientation, user-acceptance, applicability to any media, trustworthiness and lawfulness, and the existence of a common development process.

As mentioned in Section 1.1 already, it is possible to identify different levels and types of interoperability among actors, as given in Table 1.1 and 1.2.

Software service interoperability is led by Service Oriented Architecture (SOA) standards. Most of the current interoperability solutions only consider the intra-domain type of interoperability, while the more challenging inter-domain type of interoperability is still unsolved. Other and even trickier types of interoperability are human-related, but need to be managed in order to achieve the interoperability required for T2DM systems.

Information Perspective		**Organizational Perspective**
Interoperability Level	**Instances**	**Interoperability Level**
Technical interoperability	Technical plug&play, signal- and protocol compatibility	Light-weight interactions
Structural interoperability	Simple EDI, envelopes	Information sharing
Syntactic interoperability	Messages, clinical documents, agreed vocabulary	
Semantic interoperability	Advanced messaging, common information models, terminologies and ontologies.	Coordination
Organizations/Service interoperability	Common business process	Collaboration Cooperation

Table 1.1: Interoperability levels from both informational and organizational perspectives [3]

Interoperability Type	**Actors**	**Condition**
Intra-domain	Domain specialties and services	Share one policy domain and harmonize knowledge
Inter-domain	Knowledge domains	Harmonize different policy and knowledge domains
Individual	Individual persons	Share skills, languages, experiences, etc.
Institutional	Organizations (e.g. hospital)	Share business objectives and business use cases

Table 1.2: Interoperability types

1.4.2 Ontologies

The term “ontology” dates back to ancient Greek philosophy and has since acquired several meanings [12]–[19]. This ambiguity renders its use problematic, especially in the communication between different scientific disciplines, e.g. philosophy and

artificial intelligence (AI). Although there seems to be a consensus that ontologies are representational artifacts, it is controversial whether they represent (i) knowledge, (ii) terms, (iii) concepts, or (iv) real entities [17]. The first view is popular in the IA context, whereas the second and the third views refer, primarily, to thesaurus-like, not formally grounded artifacts providing terms and relations close to human language. The last view has been endorsed by philosophers and popularized in biomedical sciences. It presumes the existence of an objective, user-independent reality, about which assertions can be discovered by scientific methods [20] and to which we have at least partial access. Despite controversies, a realist approach seems to have some significant advantages: “given consensus about the things that exist in a domain of interest, agreement can easily be reached about definitions of classes of entities and, consequently, on what is universally true for all members of that class” [17].

The language used for ontological assertions defines its level of decidability and expressiveness. Currently, logic-based languages, first of all Description Logic (DL) languages are frequently used due to their availability for reasoning through deterministic algorithms [21]. The World Web Wide Consortium (W3C) has standardized several Description Logics (DL) language used for the Semantic Web. From this language family, Web Ontology Language (OWL) [22] has been widely used.

There are several hierarchies for ontologies considering their level of abstraction or generality. Some examples can be found in [23]–[25]. In the cited hierarchies, top-level ontologies (also called upper-level ontologies) introduce general types (kinds, universals) and definitions that help unambiguously categorize the entities of the world into a small set of basic categories and their relations [26]. These ontologies aim at being domain independent and a skeleton for the definition of the domain specific ontologies. Examples are Basic Formal Ontology (BFO) [27], Suggested Upper Merged Ontology (SUMO) [28], Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [29], and General Formal Ontology (GFO) [30]. Each of these top level ontologies follows certain philosophical principles, most of them based on the Aristotelian principle of *genus proximum* and *differentia specifica*. Their similarities and differences have been extensively analyzed [15], [26], [31]–[33]. Several classes and relations are common in the mentioned top-level ontologies, like *Process*, *Quality*, but their definitions differ under a closer scrutiny, so that their harmonization is only possible to a certain level. Each ontology is geared to preferred use cases, e.g., DOLCE for social sciences and BFO for natural sciences [34].

Whereas top-level ontologies are, principally, domain independent, top-domain ontologies (also called upper-domain ontologies) hold the essential core classes and relations of a domain, such as BioTop [35] and OntoCAPE [36]. The content of domain ontologies is intended to comprehensively describe the universally accepted facts, definition, and ordering principles of a domain of interest, e.g. the Gene Ontology, ChEBI, or other OBO Foundry ontologies. BioTopLite provides high compatibility with the top-level ontologies BFO and DOLCE, however considering, additionally, some relevant and general aspects of the biological domain. Two important design criteria for BioTopLite were user-friendliness and the reasoning performance. OntoCAPE is a large-scale ontology for the domain of Computer Aided Process Engineering (CAPE) and is restricted to describe Information and Communications Technology (ICT) systems. Therefore, it is an ICT specific ontology. It contains consensual classes used in the process engineering domain in a generic way such that it can be reused. An important feature of OntoCAPE is the ontological description of the General System Theory (GST) classes.

1.4.3 System Theory

The term “system” is used in many different scientific disciplines such as mathematics, physics, biology, psychology, sociology, engineering, cybernetics, and informatics. Each discipline defines the term according to its focus of interest. However, the studied systems present some commonalities explored by the General System Theory (GST). A system is defined in the GST as:

“A set of elements standing in interrelation among themselves with environment” [37]

According to the concepts of the GST, a system can be an abstract (mathematical based systems) or a concrete system (considering material objects) [38]. Usually, abstract systems are used for building models of concrete systems. So, the former ones are frequently the basis for modeling the latter. All systems serve some purpose, defined by the investigator or designer. The definition of the system environment is guided by the definition of three different purposes: “the purpose of the system, of its parts, and of the system of which it is a part, the supra-system” [39].

In this dissertation, the following topic-relevant definitions are used [3], [40], [41]:

- A *system* groups structurally and/or functionally interrelated components, which are separated from the environment defining components by system boundaries.

- Systems interact with their environment.
- Systems can be composed (aggregated) to super-systems or decomposed (specialized) to sub-systems. This relation can be recursively expressed by the system-component pair.
- The *architecture* of a system describes its components, their functions and relations.
- *Interoperability* describes motivation, willingness, interest, ability and skills to cooperate for meeting common business objectives.

A system can be studied by considering its inputs and outputs, which can be material, energy or information [42].

1.4.4 EHR and PHR

EHR is commonly defined as **"**a repository of information regarding the health of a subject of care, in computer processable form" [43]**.** Accordingly, the core component in any electronic health information system is the EHR. Health covers several knowledge disciplines like medicine, biology, chemistry, security, physic, informatics, etc. Therefore the EHR covers information related to an individual´s health status from several knowledge disciplines or domains [11]. An EHR system is the set of components that form the mechanism by which electronic health records are created, used, stored, and retrieved. It includes people, data, rules and procedures, processing and storage devices, and communication and support facilities [44]. It is a legal record moderated by accountable staff of an accredited healthcare establishment.

A Personal Health Record (PHR) represents documents related to a person's health according to the perspective of the subject of care. A PHR system manages all the functionality related with patient's PHR. The three main differences between EHR and PHR systems are that a PHR system is controlled and managed by a person outside an accredited healthcare establishment, that the user of these systems could be any individual (not only a patient), and that it is not a legal repository of the patient's health. A PHR system allows persons to self-manage his/her health, including self-control of diseases and life style improvement. Additionally, a PHR can provide communication mechanisms with health providers and other health actors. The main concept behind PHR systems is the empowerment of persons to manage his/her own health.

1.5 Objectives

1.5.1 Main Objective

Propose an approach to achieve cross-domain interoperability of health information systems in the Type 2 Diabetes Mellitus care.

1.5.2 Specific Objectives

1. Define the general architecture of a diabetes care system, its components and relationships.
2. Define use case specific architectures for the relevant use cases in the diabetes care including the related actors.
3. Develop a pilot software solution to support the relevant diabetes care use cases enabling interoperability.
4. Evaluate the interoperability functionalities of the software solution developed.

1.6 Related Works

In this section, the most relevant related works found in the literature are presented. To narrow the bibliographic analysis, four topics that are central in the problem and solution space are proposed. These topics are: Type 2 Diabetes Mellitus EHR and PHR, interoperability in diabetes care, ontology-based and architectural-centric interoperability services.

1.6.1 Diabetes Mellitus EHR and PHR

EHR systems are often implemented in healthcare establishments such as hospitals, clinics and health centers, generally improving efficiency and quality of health services. Cebul et al. [45], O'Connor et al. [46], Ran et al. [47] and Wang [48] discuss the evaluation of EHR systems in the diabetes context, reporting improvements on organizational and clinical aspects. However, details on the deployed EHR system are not mentioned. Commercial EHR systems such as GE Centricity Physician Office and Kaiser Permanente were evaluated in [49]–[51], demonstrating improvements in diabetes care and clinical outcomes. Also OpenMRS, an open-source EHR system, has been used and evaluated in the diabetes context. In [52] for example, this software system is selected as the most appropriated alternative to be used in Sub Saharan Africa.

The evaluation of this system in terms of its use and economic viability was reported, obtaining good results. OpenEMR is another open-source EHR system used in the diabetes context, which was satisfactory implemented in India as shown in [53]. A globally important EHR system is VistA, developed and used by the U.S. Veterans Health Administration. Governmentally funded, is this solution internationally reusable. The use of this EHR was evaluated from 1995 to 2005, obtaining satisfactory results in the diabetes care, improving clinical measures and information quality [54].

Santana [51] highlights the importance of decision support systems connected with EHR systems to improve diabetes care.

In the diabetes context, the use of PHR systems is increasing due to the need of special processes like changing the life style, social care and home care that can't be managed by EHR systems. These systems have been evaluated in the literature, e.g., in the research reported in [55]–[60]. The main conclusions of these evaluations are that the use of PHR systems is effectively reducing glycated hemoglobin levels, improving patient safety especially in pharmacy services, improving concordance between documented and patient-reported medication regimes, and reducing potentially harmful medication discrepancies. However, the improvements were dependent on the specific functionalities provided by the application, workflow, interface, and evaluation, so generalization is not intended.

The usability of PHR systems is one key factor for success. This aspect is evaluated in [61]–[63] recommending the use of human-centered design to improve outcomes. Main PHR systems evaluated in the context of diabetes include: Kaiser Permanente’s My Health Manager, EMIS Access, Renal Patient View, My Diabetes My Way [64], Patient Gateway, DiabetesCoach, Microsoft HealthVault, My HealtheVet, Indivo, SANA Platform and HealthView. Indivo and SANA Platform are the unique open source systems in this list and the only ones that interoperate with diabetes mobile applications [65]–[67].

A list of mobile applications for diabetes care available before December 2012, has been published in [60] and is presented in Table 1.3.

Summarizing, the review on Diabetes Mellitus EHR and PHR showed the importance, due to its effectiveness, that the use of EHR and PHR systems has in the context of the Diabetes Mellitus care. Usually these systems are evaluated separately, but it is expected to get even better results when working together.

Nutrition

- Fooducate
- Carbs and Cals
- Carb Master Free
- Carb counting with Lenny (kids)
- Calorie Tracker
- Calorie Counter
- Daily Burn
- Lose It!
- SparkPeople Food and Fitness Tracker
- GoMeals
- Weight Watchers Mobile

Diabetes news:

- Diabetes Headline News

Exercise

- Fitness
- My Fitness Pal
- Workout Trainer
- Run Tracker
- My Fitness Companion (Added by the author)

Glucose:

- Bant
- Blood Glucose Tracker
- On Track
- Diabetes App
- Diabetes Companion
- dLife
- Diabetes Buddy Life
- Diabetes Log
- GluCoMo
- Glucose Buddy
- WaveSense Diabetes Manager
- Glucol
- Glooko
- Handylogs sugar
- Islet – diabetes assistant
- Diabetes Reference
- dbees.com
- Glucose Meter
- Diabetes Log Book
- GlucaTrend Diabetes
- SiDiary
- vRee for Diabetes

Table 1.3: List of mobile application for diabetes care

1.6.2 Diabetes Mellitus EHR and PHR Interoperability

In this section, the interoperability between the PHR and EHR systems is analyzed. Table 1.4 shows reported PHR and EHR systems interoperability projects.

PHR / EHR	Kaiser Permanente	OpenMRS	VistA
Kaiser Permanente's My Health Manager	(Kaiser Permanente, 2013)		
Microsoft HealthVault	(Microsoft et al., 2011)		
Indivo		(OpenMRS, 2011)	
SANA Platform		(Costa et al., 2012)	
My HealtheVent			(Kupersmith et al., 2007)

Table 1.4: Relevant projects addressing PHR and EHR systems interoperability

Systems like OpenMRS, Microsoft HealthVault and Indivo support Clinical Document Architecture (CDA), especially the Continuity of Care Document (CCD), which is the CDA representation of the ASTM E2369 standard Continuity of Care Record (CCR) [68], and provides an API facilitating their interoperability with other systems. Therefore, the development of interfaces for those EHR and PHR systems is feasible. Additionally, OpenMRS and Indivo provide a REST API facilitating easy collaboration between these systems and other web services. None of the mobile applications for diabetes care listed in the previous section describe EHR systems interoperability functionalities.

Despite the increasing dissemination of EHR and PHR systems, the possible interoperability between them is limited, and a mechanism to facilitate interoperability is needed.

1.6.3 Ontology-based Interoperability Services

Interoperability is a common need in different domains such as e-health, e-learning, manufacturing and networking, just to name some of them. Following, existing ontology-based interoperability approaches are analyzed.

In the networking domain, Castano et al. [69] propose a model for collaboration in open networked systems. The model is partially implemented using the matching tool H-MATCH. This tool uses matchmaking techniques considering linguistic and contextual features.

In the manufacturing domain, Chungoora et al. [6] propose the combination of separated views in a Model Driven Architecture (MDA) and the use of common logic-based ontologies. The concept has been implemented under the Interoperable Manufacturing Knowledge Systems (IMKS) project. The concept proposed is applied to the development process of systems and uses model transformations to generate ontologies expressed in Extended Common Logic Interchange Format (ECLIF) language [70]. Tessier [5] deploys a hybrid ontology approach, where a shared base ontology is used to convey the concepts that are common among different Computer-aided design (CAD) systems. The use of OWL and Semantic Web Rules Language (SWRL) rules enables automatic transformation of concepts to a target CAD system. Chungoora's work shows an implementation of this approach, however the evaluation is not provided. The CAD ontology (domain ontology) was manually created using Protégé, a free, open-source ontology editor and framework developed at Stanford University [71].

In the e-learning domain, Archer et al. [72] propose a Semantic Ontology Mapping service for Interoperability of Learning Resource Systems. To enable semantic ontology mapping, this research proposes conflict detection and resolution techniques for both semantic and structural conflicts. Ontology-based learning object metadata is generated and used by a semantic query engine to facilitate user queries of learning objects across heterogeneous learning resource systems. This work has adopted the Common Learning Object Ontology (CO), expressed in Web Ontology Language (OWL) as common ontology, which incorporates common metadata schemes in e-learning domain such as, IEEE LOM (Learning Object Metadata) and the Dublin Core. To enable conflict resolution, this work proposes a Semantic Bridge Ontology Mapping tool to generate the Semantic Bridge Ontology (SBO). The tool provides a mapping interface to map terminologies of different local ontologies to a common set

of ontologies and terminologies defined in CO. SBO enables the automatic resolution of mapping using SWRL rules, but the discovery mapping process is not automatic. The SBO ontology formally describes possible conflicts between two ontologies. The paper didn't show an evaluation of the conflict detection and solution algorithms.

In the e-health domain, Sonsilphong et al. [7] adapt the SBO developed for the e-learning domain, proposing a Semantic Interoperability Framework for Data Integration (SIDI), which enables integration of information from heterogeneous health databases. In this work, the HL7 (ICT) ontology is used as global ontology for the mapping process. The SIDI framework is designed as a layer of collaborating stakeholders. The Resources Layer is the layer of the provider system, the Mediator Layer acts as a broker system, and the Application Layer is the layer of the data requester. An evaluation of the data recovery is shown with very good results measuring precision and recall. The global ontology is too small because it is based on general concepts of the HL7 (ICT) ontology. Therefore, the scope of knowledge that can be expressed is limited.

Snyder et al. [73] propose a system for managing and exchanging electronic medical information. The components are: a rule management component for executing conceptual rules, an ontology management component, an information model management component, and a system configuration management component. The ontology management component manages mappings between members of different ontologies. The ontology management component is configured for managing a domain of terms representing at least one of the following terminologies: Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT), Medical Dictionary for Regulatory Activities (MedDRA), and an organization specific terminology. HL7 standards are also used as reference in the development of the database and for the development process. Uribe [74] proposes an ontology-based interoperability service for EHR using SNOMED CT as domain ontology and COMA CE [75] as matching tool. The work offers automatic matching at the terminological level supporting the interoperability process.

All these works offers interoperability at some level and use ontologies as mechanism to represent knowledge. None of the aforementioned proposals is available as open-source project. Furthermore, they do not report quality evaluations.

This section on Ontology Based Interoperability Services demonstrates the power of applying ontologies for the interoperability of systems in a set of different domains.

However, our proposal goes beyond as it connects domains which haven't been ontologically interrelated so far (e.g. medical, resource and policy domains). This is an essential feature in healthcare systems' interoperability.

1.6.4 Architecture-based Interoperability Services

The most important basic principles of the architecture-based approach are presented in [11]. This work presents the use of the Generic Component Model (GCM) (introduced in the next section) in the analysis, design and implementation of health information systems considering the systems of interest as composition of components and relationships. These components and relationships can correspond to different ontological domains. Finally, the described system is a simplified model of the reality according to the business process, expressed in a formal way. The needed of interoperability of different ontological domains and the complexity in healthcare environment is clearly shown in [40]. Blobel [25] explains the importance of considering the business process and the entity interoperability. Entity interoperability covers the collaboration of all actors involved in the system and not just data interchange between computers.

Most of these principles were implemented in the Health Information Systems – Development Framework (HIS-DF). The development framework aims at providing a comprehensive architecture development process and supporting semantic interoperability when designing healthcare systems [76].

Currently, none of the architecture-based works provides interoperability considering computer independent aspects.

1.7 Methods:

1.7.1 A General Framework for Systems Architectures

The GCM is a framework for the analysis, design and implementation of systems (in the most general sense) following an architectural approach, derived from the GST. It is visualized as a cuboid, Figure 1.2, due to its three-dimensional make-up: (i) the domain perspective, (ii) the development process perspective and (iii) the architectural perspective [11]. The latter describes the system through the decomposition/composition of its components and their functions and relationships.

The selection of components and the constraints on their functions and relationships according to the current business objective of the system describe the system’s

behavior. The architectural perspective considers four different generic levels of granularity. Structural properties of the systems can be described using relationships “is part of” or “is connected with”. Granularity is expressed by the relationships “is a” (from more general to more specific descriptions) and “is part of” / “has part” (by describing components and subcomponents at different levels of detail). The domain dimension (domain perspective) brings order into the description by separating inter-related domains of the system in order to manage them independently. A domain is characterized by common properties of its architectural components. Each domain in GCM usually reflects the interest of a different group of persons and is often represented by a domain specific ontology. The domain specific ontologies should be harmonized by an upper-level ontology in order to facilitate interoperability. The last dimension describes the development process, represented by the different views of the system according to ISO 10746 Information technology - Open Distributed Processing – Reference Model (RM-ODP) [77].

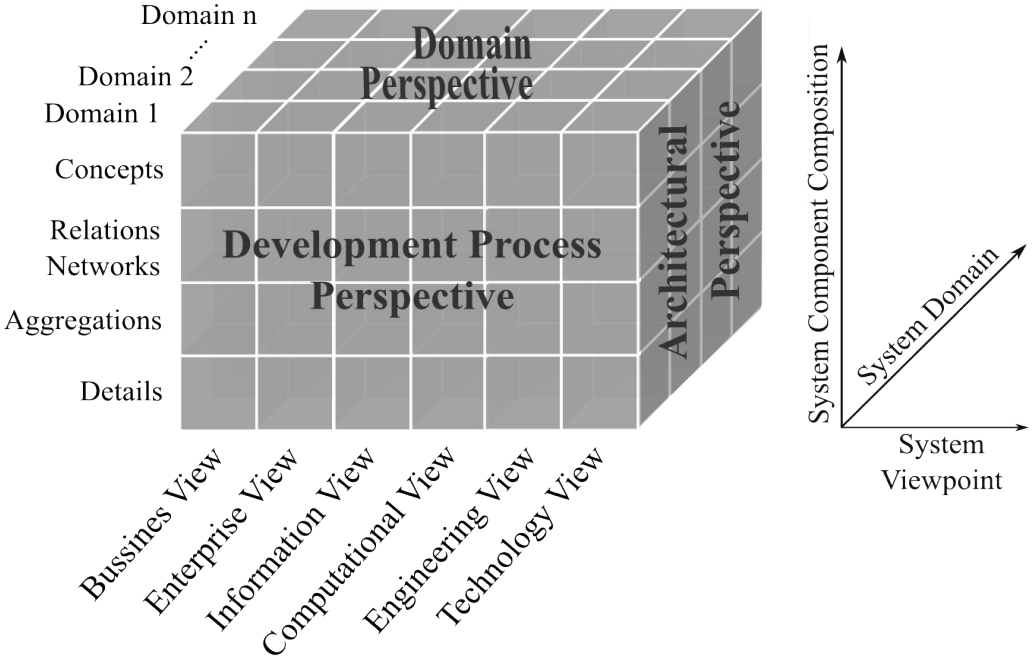

Figure 1.2: The Generic Component Model according with [25].

The GCM framework additionally considers the “Business View”, i.e. the description of a real system (ICT-independently) [25], considering the business process of the system and its use cases represented by the aforementioned domain specific

ontologies and their harmonization. It thereby goes beyond the RM-ODP which always focuses on ICT systems, represented using ICT ontologies.

In order to build an understandable architecture with the GCM it is needed to take into account the following design principles (Good Modeling Practice): orthogonality (not linking independent aspects), generality (not introducing multiple similar entities), parsimony (not introducing irrelevant aspects), and propriety (not restricting inherent aspects) [78]. An important principle derived from the orthogonality is the not linking of entities at different levels of granularity.

GCM combines system theory and ontology sciences for representing the architectural components of a system [79]. In that context, ontological assertions expressed in domain ontologies are amended by functional constraints and relationships specific for the system in consideration [3], [41]. The result is named application ontology and is finally implemented using ICT ontologies [25], [80]. ICT ontologies support the software development process, implementing for example specific software applications.

1.7.2 System Representation

The system in question is designed using the system-theoretical, architectural approach according to the GCM framework by defining the system with its boundary and its environment, the system's perspectives (domains), and the system's architecture refined for each domain. As mentioned before, the GCM framework additionally integrates the development process for the related ICT system.

To represent use case specific GCM components regarding their names and underlying concepts, but also their basic relations, domain specific ontologies of the domains considered in that GCM instance are deployed. To interconnect components across domain boundaries, ontology harmonization must be performed.

For representing the rules for the use case specific selection of components and the constraint of their functions and relations (policies to rule the system's behavior), XML-based policy languages and/or logic languages at different level of formalization are used.

For representing the system's processes defined by the components' functions and relations, process description languages are exploited.

To represent use case specific ICT viewpoints of the GCM components, their

functions and relations, the IBM ICT ontology, the SOA ontology [81] and health informatics specific representations such as HL7 RIM [82] and its vocabulary are deployed.

For the graphical notation of the use case specific GCM instances, UML (Unified Modeling Language) including its constraint language OCL (object constraint language) are used.

1.7.3 Methods for Business Use Cases Modeling

The GCM framework proposes a methodology for describing, designing and implementing systems considering its components and their relationships. The most important output of this methodology is the architecture of the system. The architectural model of the system is usually shown as a cuboid that explicitly separate the different perspectives of the system (domains), the granularity levels (aggregation levels) relevant for the description/design and the viewpoints of the system according with the RM-ODP. For concrete instances, the block diagram elements are represented using UML [83]. Regarding the illustration of the system's behavior, block diagrams and UML diagrams can be complemented by the Business Process Modeling Notation (BPMN) [84]. UML and BPMN – introduced in some more details later on – are broadly used for the development of software systems and allow the automation of some steps of this process. UML through its structural diagrams formally defines some important system component relationships such as aggregations, compositions, generalizations and realizations. UML also provides activity and sequence diagrams for describing the behavior of the system. However, the behavioral diagrams are limited to the description of software systems. Therefore, the BPMN language describes more easily the behavior of complex systems like the T2DM care. Consequently, the architecture in this thesis is graphically represented using the GCM cuboid representation, complemented by UML class diagrams for the structural aspects and by BPMN diagrams for the behavioral or procedural aspects.

Additionally to the graphical representation, the rules applied in the T2DM care system are described using a formal language. This description allows developing intelligent and adaptive systems. This methodology is explained in the following section.

1.7.4 Business Process Modeling and Execution

The business process realized by a system is defined by the system's components,

their functions and interrelations in the context of a specific business case. The business process can be constrained by policies applied to the system, defining the system's behavior as exemplified later [85]. For correctly reflecting a system's architecture and its ontological representation, the business process model shall be derived from the system's architectural model.

The formal description of the business processes or workflows of organizations is a shared problem of many disciplines. Such a description enables the use of tools for designing, optimizing, implementing and monitoring business processes. This formal description can consider ICT independent aspects, but it is usually intended to consider at least the partial support of the business process by computer systems. For solving this problem, the Object Management Group (OMG) has developed a standard called Business Process Model and Notation (BPMN). Version 2 [86] of this standard also presents an execution semantics enabling a standard implementation of the business process.

BPMN version 2 is supported by many tools. However, most of them require a license and have a proprietary file format or business process execution platforms, limiting the use of the tools and its outcomes. The use of freeware/open source tools supporting the modeling and execution of business processes is desired. Table 1.5 provides a comparison of the available tools. This table considers the description of rules as important factor guiding the execution of business processes. However, an extended discussion of those rules, rule languages and tools supporting them is out of scope of this dissertation.

The tool BonitaBPM [87] presents similar functions as the other tools. However, the BPMN file format used for this tool is not completely standardized. Therefore, other tools would have limitations to process a BonitaBPM outcome. None of the listed tools supports the execution of all BPMN elements, despite that the list of supported elements is similar. The tool Activiti 5.15 [88] is unable to model the elements not supported by the engine, like the message flow elements [89]. The tools provided by Camunda [90], [91] enable the use of external business rules tools. They provide an example of integration with Drools using rules for Drools Rule Languages (DRL).

Tool	Provider	Integrated Technologies	Rules Description Language	License
Activiti 5.15	Alfresco	Spring, Drools Expert engine, JTA	DRL for business rules, Java	Apache License 2.0
BonitaBPM 6.3 community version	BonitaSoft	JavaAPI, REST	Decision tables for business rules, Java	General Public License, version 2
Camunda modeler 1.2 and BPM platform 7.1	Camunda	JavaEE / Spring Framework, REST	Java	Apache License 2.0
JBPM 6.0 [92]	RedHat Jboss	Drools, JBoss Server, Spring, OSGi, REST, JMS, Maven, JPA	DRL and decision tables for business rules, Java	Apache License 2.0

Table 1.5: Comparison of open source tools for BPMN version 2 modeling and execution

1.7.5 Rules and Languages

The term “rule” has different meanings, i.e, it refers to varied concepts [93]. Rules used for analyzing, describing, and implementing systems can be expressed in the form “if … then ...”. These rules can be classified in two groups. The first one is named “production rules”, and the second one “declarative rules” (also known as inference rules). Production rules determine a behavior plan. If a certain condition holds, then some action is performed (e.g. “If the body temperature measurement is greater than 37.5 Celsius degrees, then take a pill.”). The declarative rules state a fact about the world (e.g. “If the body temperature measurement is greater than 37.5 Celsius, then is a fever finding”) [94]. These two types of rules can be described deploying different languages such as SWRL [95], SPARQL Inference Notation (SPIN) [96], or Rule Interchange Format (RIF) [97] and then be processed by rule engines such as Drools [98], Jess [99], or IBM Operational Decision Manager [100]. The mentioned rule engines were designed focusing on production rules, and this is done independently of ontology languages such as the OWL [101] or the Resource Description Framework (RDF) [102]. SPIN, SWRL, and more recently RIF, are languages that allow the definition of rules using ontologies. The RIF language is a W3C standard based on the commonalities of all the current solutions, in order to allow sharing rules between systems. Unfortunately, RIF standard implementations

are still immature [103].

In the medical domain, there are several domain specific languages for describing rules in the context of the medical guidelines definition. Some examples are PROforma [104], Arden Syntax [105], Asbru [106], Guideline Interchange Format (GLIF) [107], and SAGE [108]. These solutions are compared and discussed in [109], [110]. All these solutions have many similarities. Thereby, the SAGE system builds on prior work such as GLIF, PROforma, and Arden Syntax. An important disadvantage of these languages is their exclusive focus on the medical domain, so making the harmonization with the administrative, ethical, security and privacy domains difficult. Therefore, in order to harmonize different domains the use of general purpose, standardized and broadly accepted rule language such as SPIN is convenient. Furthermore, contrary to the other languages, there are Integrated Development Environments (IDE) for the implementation of ontology-based systems with SPIN language.

1.8 Contributions

During the development of this thesis were obtained the next contributions:

1. The paper “Towards automated biomedical ontology harmonization” describe a pathway to achieve interoperability through the use of software systems. This paper was published in *Studies in health technology and informatics* 200 in the year 2014 and presented in the international event pHealth2014 [111].
2. The paper “A Generic Architecture for an Adaptive, Interoperable and Intelligent Type 2 Diabetes Mellitus Care System” describe the generic architecture for the diabetes care system. This paper was published in *Studies in health technology and informatics* 211 in the year 2015 and presented in the international event pHealth2015 [112].
3. The paper “Specializing Architectures for the Type 2 Diabetes Mellitus Care Use Cases with a Focus on Process Management” describe the specialized architecture for the pharmacological glycemic control use case. This paper was published in *Studies in health technology and informatics* 211 in the year 2015 and presented in the international event pHealth2015 [113].
4. The paper “Design and Implementation of an Adaptive, Interoperable and Intelligent Type 2 Diabetes Mellitus Care System” summarizes this thesis including the implementation of the software system and the evaluation performed. This paper is not published yet and is added as the Appendix B.

1.9 Structure of the Dissertation

In Chapter 2, the generic architecture of the T2DM care system is presented. This architecture is valid for any use case of Diabetes Mellitus care. The architecture is described in its structure and behavior. The structure of the system is described through GCM models as block diagrams and UML class diagrams. The behavior of the system is described through BPMN models. Chapter 3, contains the specialization of the generic architecture for the glycemic control in pharmacotherapy use case. In this use case, the description of the behavior is enriched using SPIN rules describing the policies governing the behavior of the system in this use case. In Chapter 4, the implementation process of a software pilot for the T2DM care is presented. The implementation process has as input the description provided in Chapter 3. The implementation is an ontology-based, flexible, adaptable and intelligent system allowing the interoperability of the heterogeneous actors involved in the diabetes care. Finally, Chapter 5 contains the conclusion of the entire dissertation and the future work proposed.

Chapter 2

Generic Architecture for Type 2 Diabetes Mellitus Care System

In this chapter, the T2DM care system architecture and its business process description is presented.

2.1 Generic Model of the T2DM Care System

Figure 2.1 provides a GCM presentation of the T2DM care system at high level of abstraction. At this abstract level, it is important to define the scope of the system, its inputs from, and outputs to, the environment. The system is described considering three domains: medical, policy and resource. The medical domain describes the components and related processes of the medical discipline health professionals represent (e.g. physicians and nutritionists). This domain is represented by evidence-based axioms and is independent of the organization or jurisdiction. The resource domain considers the actors (i.e. humans and organizations, but also devices, etc.) and other resources like locations and facilities (e.g. drugs and equipment). The policy domain includes as sub-domains clinical, ethical, security, privacy, regulatory, and administrative policies. It represents the rules applied for actors to perform specific medical activities. Policies might be defined internally to the system (e.g. within organizations such as hospitals) or externally to it (e.g. regionally, nationally, internationally). Usually, policies are defined at the levels of jurisdictions and healthcare organizations. The clinical policies are mostly known as clinical guidelines. In [114], the policy sub-domains are grouped to clinical, contextual, and organizational/administrative policies.

The interaction of different domains happening in any multi-domain system, enabling the system's purpose, is called "cross-domain interoperability". This interoperability is performed at different level of granularity or specialization regarding both the GCM architectural dimension as well as the domain dimension. This means that the interaction between different medical sub-domains (e.g. specialties) is also cross-domain interoperability. As the domains are usually developed independently, it is important to define mechanisms for achieving this type of interoperability. The use of the architectural hierarchy of the ontology system from application ontologies (details) through domain ontologies (aggregations) to top-level ontologies (relations network) [23] is a key factor to achieve this interoperability and to maintain it over time.

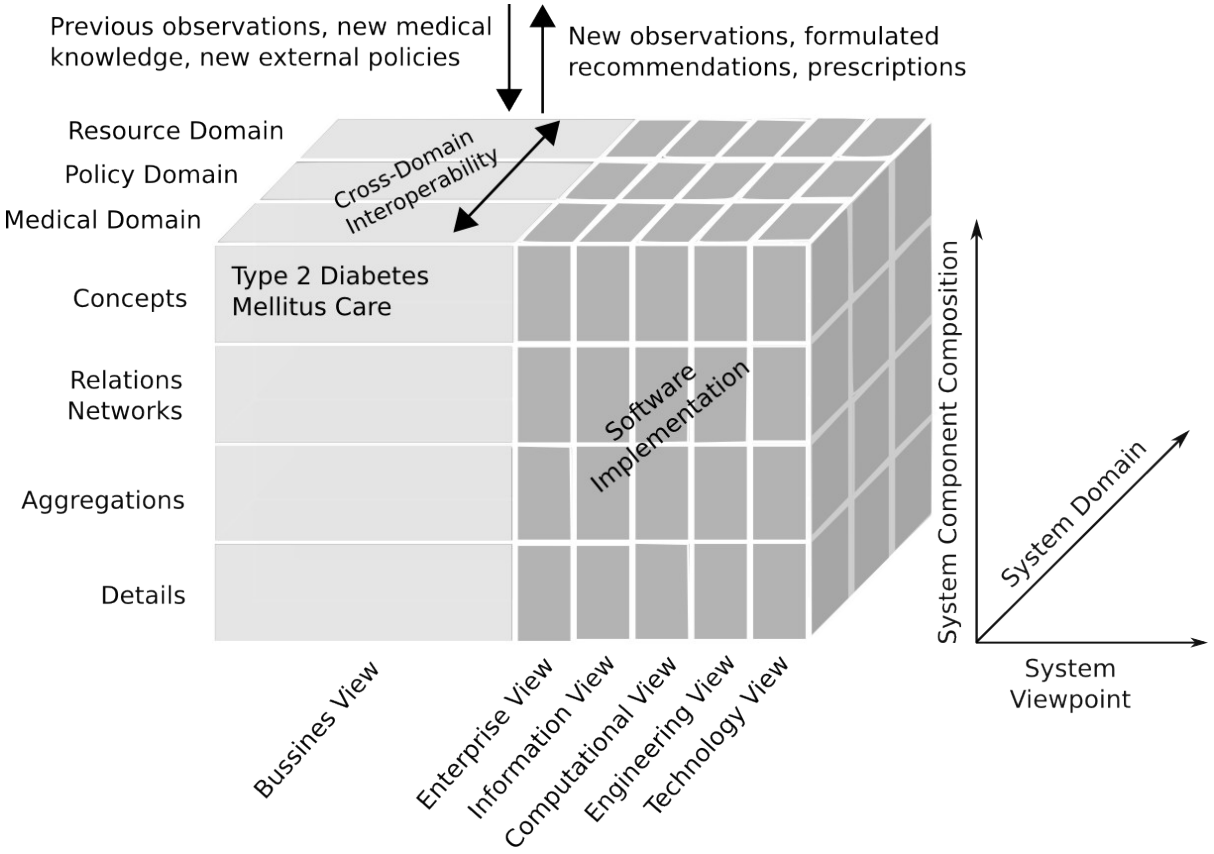

Figure 2.1: Abstract description of the system

When the domains have been defined, it is important to define the inputs and outputs of the sub-systems from, and to, the environment. Inputs are previous observation results of the health of the person, the medical knowledge, and external policies (e.g.

international or national guidelines). The outputs are observation results and plans like recommendations and prescriptions resulting from some medical process. The T2DM care system needs to be adapted according to changes in external policies and in the medical knowledge.

In Figure 2.1 and in the GCM figures following, the explicit representation of functions and relationships as inherent part of the GCM model is omitted. In the class diagrams presented in Section 2.3, the most relevant relations will be provided.

2.2 GCM Representation of T2DM Care System Domains

The medical domain defines the main components and resulting processes performed in the T2DM system, and so the main business use cases of the system (e.g. diagnosis or treat patients). For correctly performing medicine, the system architectural principles of healthcare organizations must be properly interrelated to those of the medical domain. In consequence, medical ontologies and ontologies representing concepts and relationships of organization sciences are interrelated as well and have to be managed in interoperability business cases. The level of medical complexity of some specific use cases corresponds with the level of organizational complexity needed. For understanding the behavior of the T2DM care system, it is necessary to decompose it into its parts and their interactions, finally obtaining its architecture. The medical/care domain of the T2DM system can be refined into specific sub-domains with specific ontologies, partially defined by their view on medical practice or by regulations and representing different levels of complexity. Therefore, the medical/care domain can be decomposed in the following sub-domains:

- regulated intra-organizational interdisciplinary collaborative care (e.g. provided in hospitals),
- regulated subject-specific care (e.g. provided in health professional offices), enabling just inter-organizational interdisciplinary care,
- non-regulated subject-specific care (e.g. provided by specific health service providers),
- non-regulated interdisciplinary care (e.g. provided in home care and self-care).

Figure 2.2 presents this architectural decomposition. Because of the aforementioned dualism of medical and organizational complexity, the sub-domains are simply named according to the typical organizational instance. In the hospital domain with its higher complexity level, T2DM can be managed by different clinics, institutes, or departments (e.g. internal medicine / endocrinology, cardiology, ophthalmology, imaging, radiology, lab medicine, emergency, dietary), summarized as units. Each unit performs some health services related to diagnosis, treatment, or prevention of the disease. Finally, all the units collaborate for caring the T2DM patient. Health professional offices provide subject-specific T2DM related health services. Interdisciplinary collaboration is provided at inter-organizational level between different offices or between them and hospitals. The services provided by those organizations are composed of many tasks and some of these tasks can also be provided by an independent health service provider (e.g. a nutritionist or a fitness trainer) or by a home care organization. The home care organization can also include some health services usually out of the scope of the regulated healthcare system. Finally, the patient performs the tasks needed for completing the health service satisfactorily (self-care tasks).

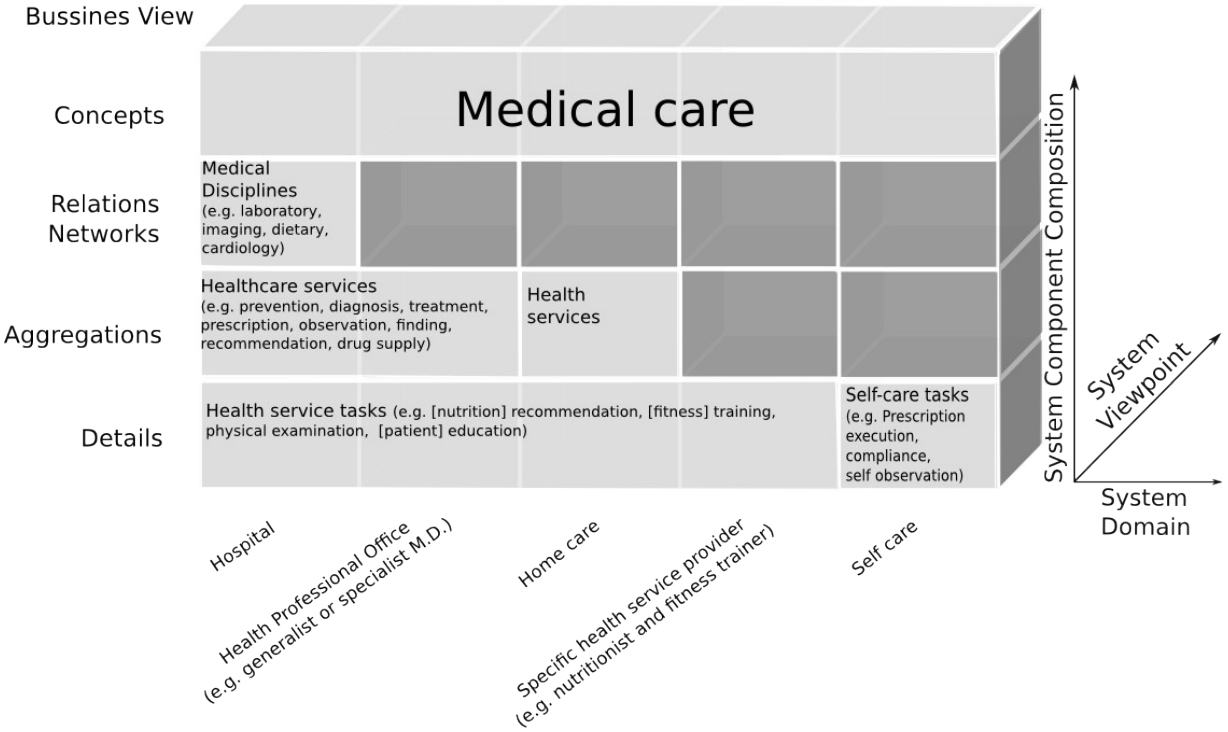

Figure 2.2: Medical domain GCM business view representation

The policy domain contains concepts and relations ruling the deployment of the T2DM care system. It covers the sub-domains of medical, ethical, security and privacy policies. Medical policies, also called clinical guidelines, define the workflow of the medical activities, the internal medical terminology used and constraints on processes and actors. They may vary according to the jurisdiction. Medical policies, which are human-defined, have to be distinguished from natural medical processes, which are represented in the medical domain. Ethical, security and privacy policies define selections of components and constrain functions, attributes, and relations within the medical domain as well as between the medical and the resource domain. In summary, those policies constrain medical processes according to pre-existing principles and values that are universally or locally accepted, or dynamically established by a user group. For example, the execution of a treatment procedure may be constrained by the informed consent of the patient, or monitoring data is not possible due to a patient's privacy policy. Figure 2.3 describes the architecture of the policy domain considering the sub-domains application and management.

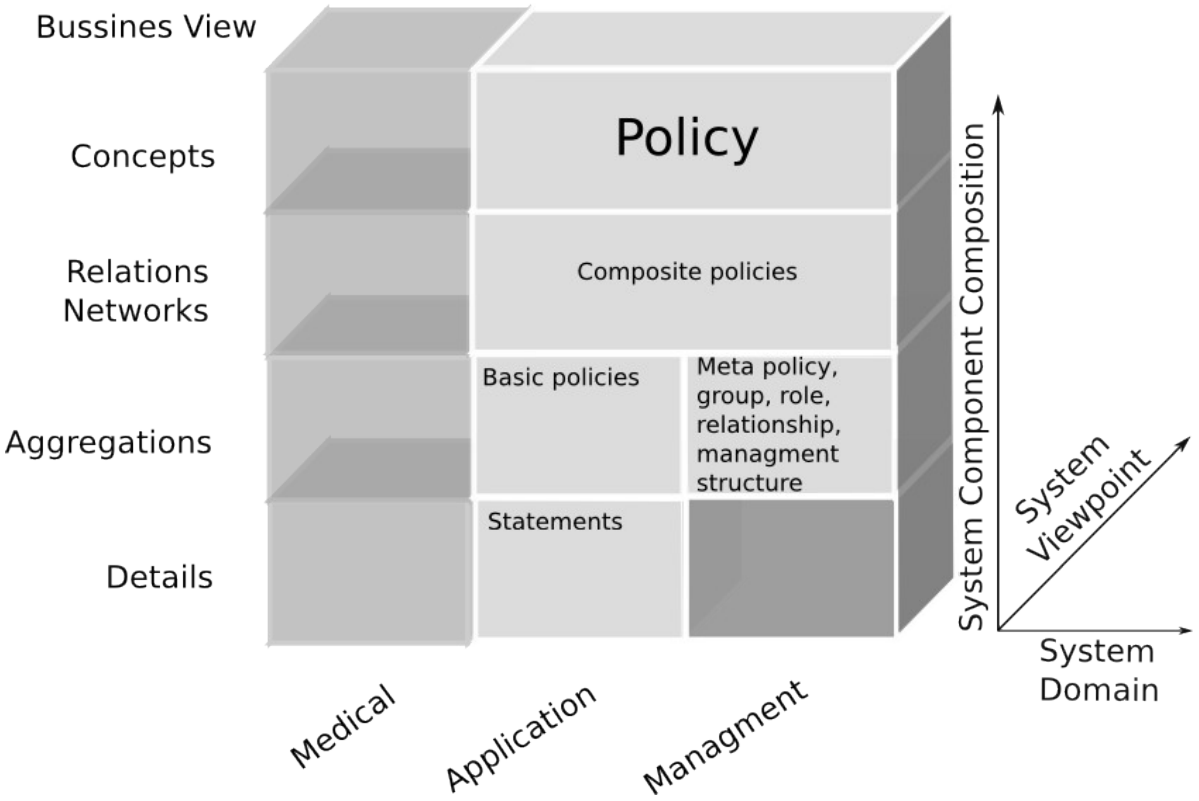

Figure 2.3: Policy domain GCM business view representation

The application sub-domain contains the components needed for applying the policies in a specific scenario. The management sub-domain contains the components needed for defining and harmonizing the policies prior to the performance of a specific task or service. The meta-policy enables the definition of policies. Composite policies are the main components in charge of the harmonization of policies, also called policy bridging. Additionally, for grouping and aggregating policies the components groups, role, relationships and management structure are used. After the harmonization of policies, a basic policy is obtained. Basic policies are the main components for selecting and constraining components, functions, and actors in a system. The basic policy is composed of single statements describing the rules applied. The presented policy architecture and ontology follows ISO 22600 Health informatics – Privilege management and access control [115], also described, e.g. in [116]. For another discussion of the policy domain see [114].

The resource domain can be decomposed into the following sub-domains: actor, facility and location. Temporal aspects are considered, e.g., in the medical (process) domain. Figure 2.4 shows the architecture of the resource domain.

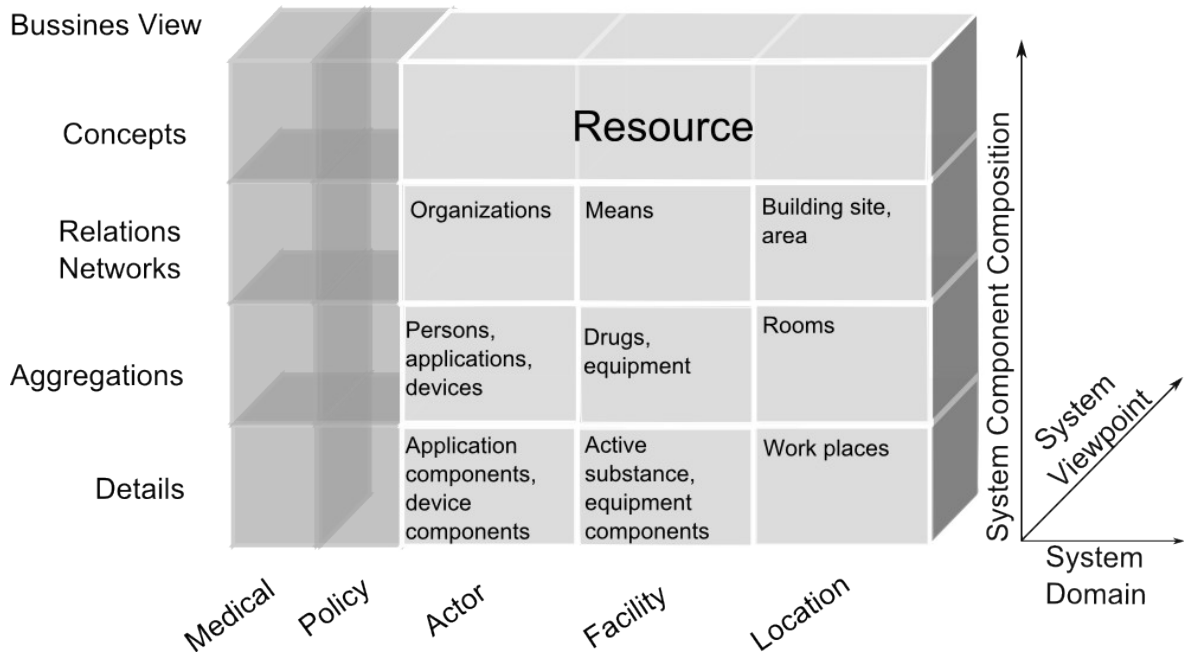

Figure 2.4: Resource domain GCM business view representation

The actor sub-domain includes the resources able to perform tasks in the system (i.e. organizations, persons, applications and devices) and their components. Organizations perform multi-disciplinary processes. Persons, applications and devices perform specific tasks in order to provide health services. There are some complex devices and applications able to provide a complete service, while their components perform more specific tasks.

According to [117], facilities are the means or equipment needed for an activity. Therefore, the facility sub-domain includes the objects used by actors to perform tasks in the system, such as means, equipment or drugs.

The location sub-domain includes the one-, two- or three-dimensional space occupied by the facilities and actors. These can be buildings sites, areas, rooms and workplaces.

2.3 Class Diagrams of the Detailed Architectural Models

In this section, the classes of the components derived from the GCM architecture models including basic relations will be presented. Figure 2.5 shows the classes of the medical domain using a UML class diagram. In this domain, the classes represent the medical care process or the organization that performs these activities.

The medical discipline, healthcare service, health service, healthcare service task and the self-care task classes represent the different level of complexity of the medical care process. The health service class represents the health related process not covered by the healthcare institutions (e.g. social care). The specializations of the medical care process classes are introduced as an example. Therefore, this is not an exhaustive list. The organization classes considered correspond to the sub-domains in the GCM model.

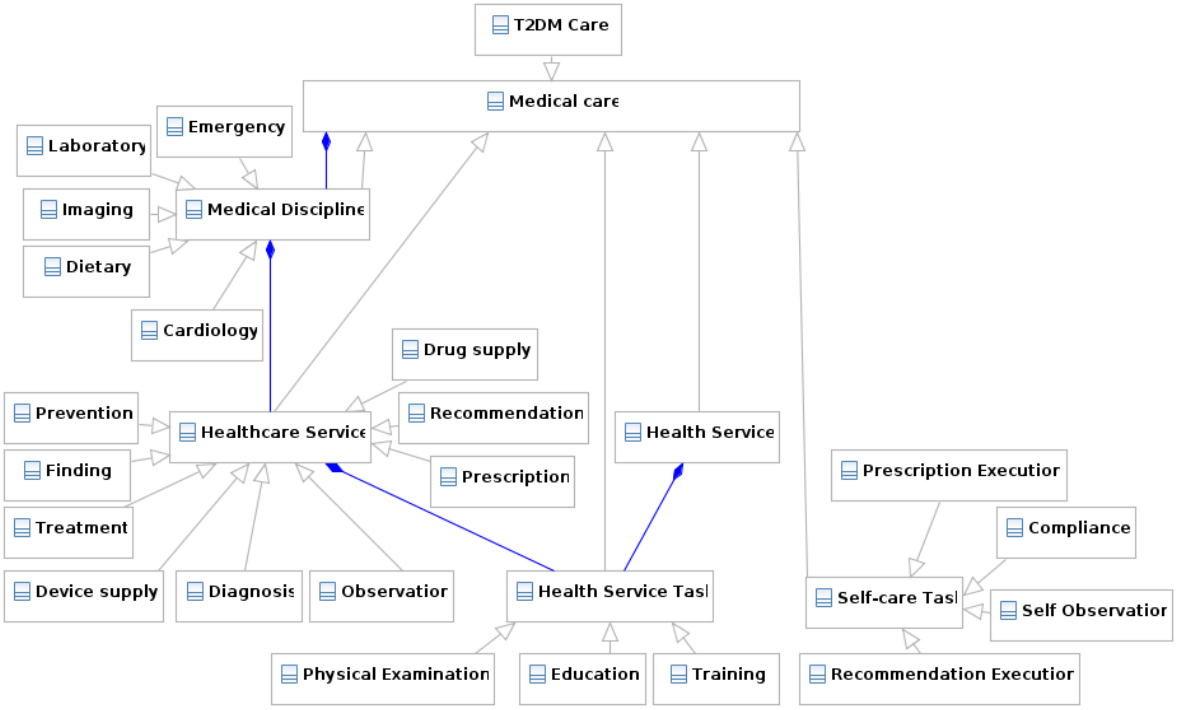

Figure 2.5: Medical domain classes

Figure 2.6 shows the policy domain classes. These classes are defined in the ISO 22600 standard [115]. All policies are specializations of the policy class. The basic policy can be specialized in authorization, obligation, delegation, and refrain policies. The authorization policy and the delegation policy provide a positive or a negative decision. A further explanation of the classes can be found in [118].

The resource domain classes are shown in Figure 2.7. The specializations of the resource class are divided in three groups according the sub-domains: actor, facility and location. The actor class is realized in organizations, persons, devices, and applications. An organization is composed of persons, devices and applications. Device and application can be decomposed if they fulfill tasks in the system. To provide services by performing actions, actors use facilities, which are associated with locations. The facility class can be realized in means, equipment, drugs or its components. Means can be specialized in equipment or drugs. The latter two are composed of equipment components or active substances, respectively. All the classes correspond to the components introduced in Section 2.2.

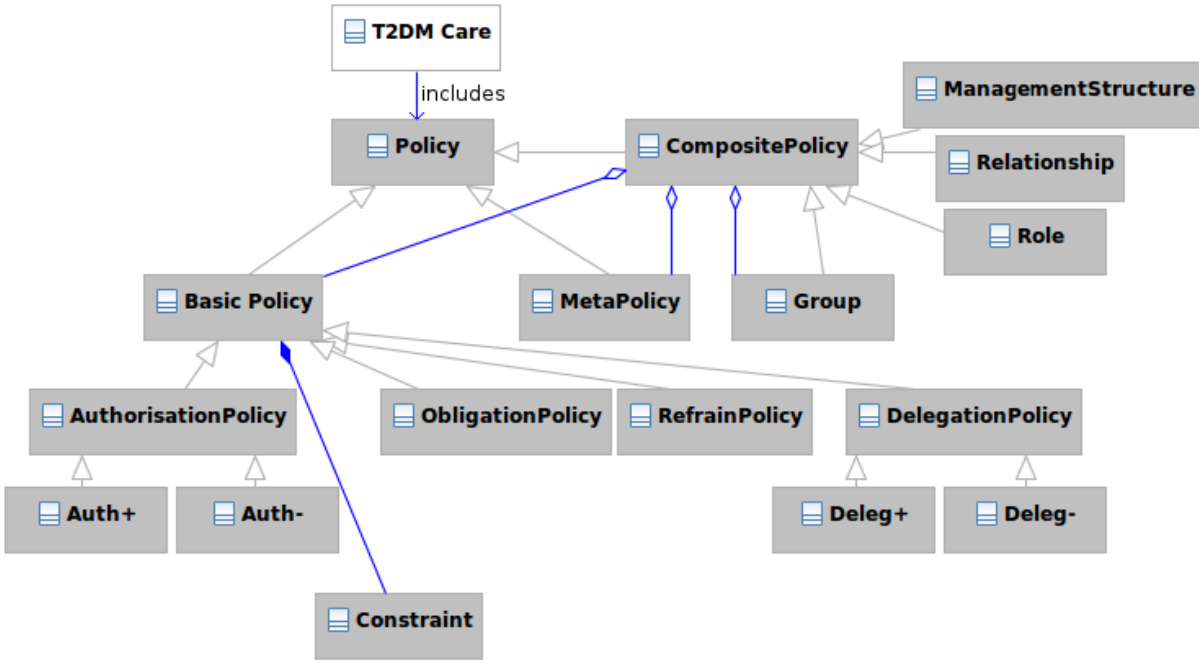

Figure 2.6: Policy domain classes

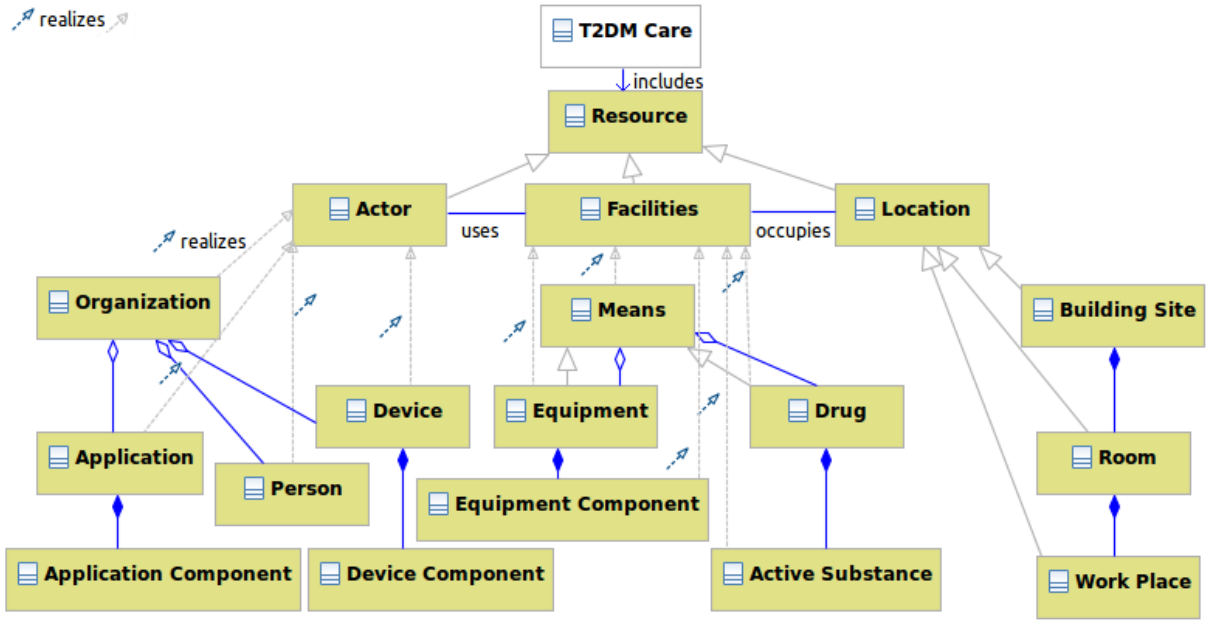

Figure 2.7: Resource domain classes

Figure 2.8 presents the relationships between the considered domains. This model demonstrates that the relations between medical care and the resource classes are regulated by the policy class. It means that the resource participating in the medical care process is ruled by the defined policies. Medical discipline and organization class instances are regulated by composite policies due their multi-disciplinary nature. Person, device and applications are regulated by the basic policies in order to perform healthcare service or healthcare instances. Finally, specific statements (constraints) can be used to rule the specific tasks.

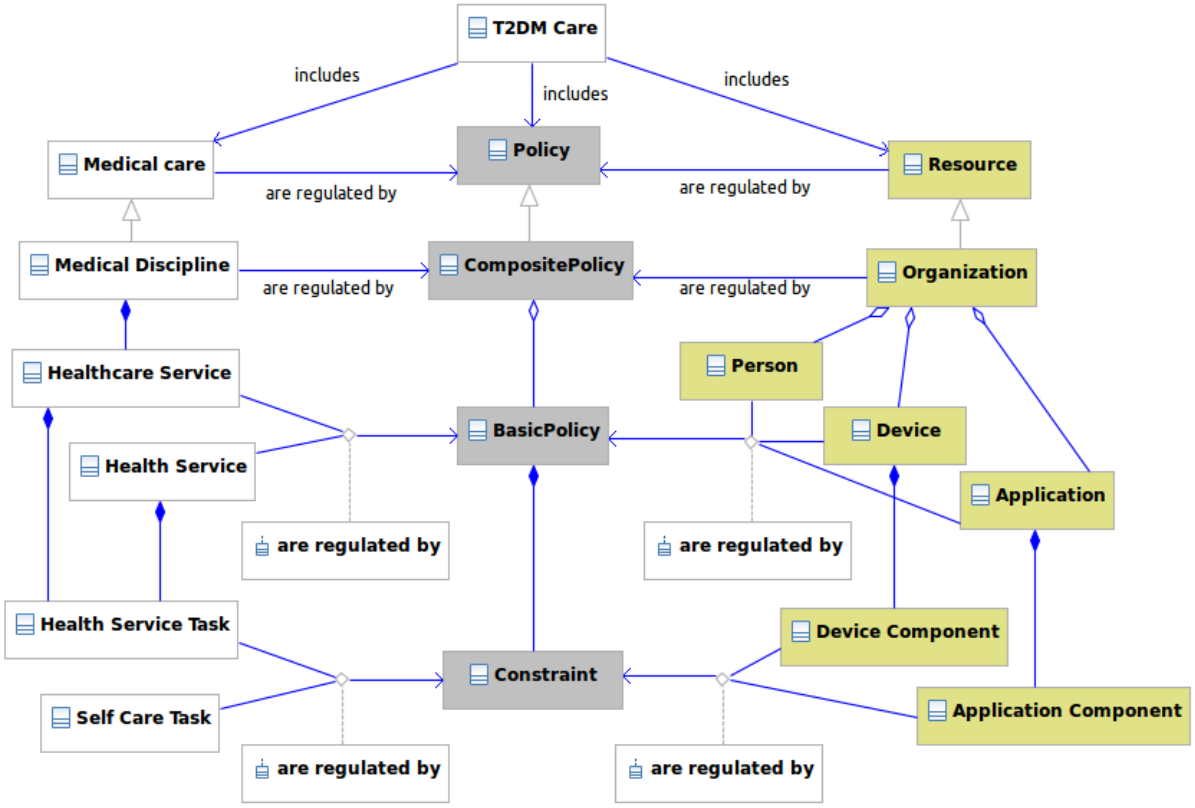

Figure 2.8: Inter-domain class diagram

2.4 Ontological Representation of the T2DM Care System

Ontologies are used for naming and describing the types of components they

represent as well as basic relations in the system architecture. The composition / decomposition hierarchy follows architectural principles of the system in question, thus constituting a mereological order, opposed to the taxonomic backbone of the ontology.

In the medical domain, several terminologies and ontologies describe the basic concepts of the medical domain and the terms used. Some examples are Logical Observation Identifiers Names and Codes (LOINC) [119], International Statistical Classification of Diseases and Related Health Problems (ICD10) [120], OBO Foundry ontologies [121] and SNOMED CT [122]. The maturity level of the evolution towards an ontology is quite different for the given examples. Current medical ontologies do not meet all the criteria desired for interoperability [123]. Nevertheless, SNOMED CT is the most comprehensive ontological effort in this field and therefore used as main domain ontology. Nevertheless, other terminologies or ontologies can be used for sub-domains (e.g. LOINC in the laboratory discipline). Evidence-based axioms related to the T2DM disease (e.g. if you have metabolic syndrome, then you are at risk of suffering from T2DM) are not present in the current ontologies. This kind of knowledge is beyond what is commonly considered ontological, but which nevertheless needs to be declared in a formal language, e.g. a rule language.

The professional (occupational) roles of human actors are defined in the International Standard Classification of Occupations (ISCO) of the International Labor Organization (ILO) [124] and specialized for health informatics in ISO 21298 Health informatics – Functional and structural roles [125]. The occupations considered for the description of the T2DM care system are medical doctor, nutritionist, dietitian, nurse, psychologist and pharmacist. Medical doctors can be generalist medical practitioner or specialist medical practitioner. Specialist medical practitioners in the context of T2DM are nephrologists, cardiologists, neurologists, surgeons and ophthalmologists. Furthermore, some additional roles are considered like family roles.

2.5 Formal Description of the T2DM Care Business Process

The business process of T2DM is presented using BPMN diagrams for each level of granularity. However, the processes have strong dependencies between the different granularity levels as shown in Figure 2.9. For example, health tasks need the health service knowledge/context for performing correctly. Consequently, this principle can

be extended to all the other granularity levels.

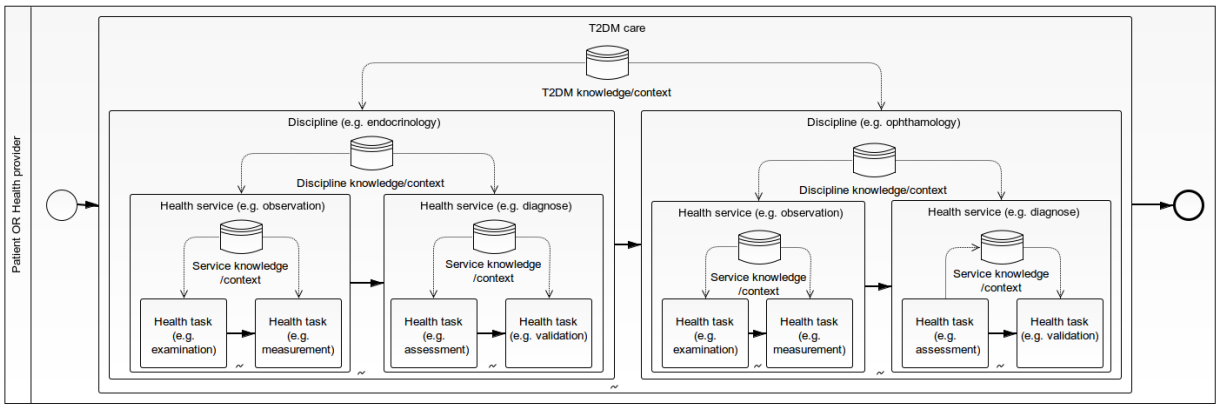

Figure 2.9: Architectural consideration in the business process

Ad-hoc sub processes enable the modification of the workflow according to the policies and rules present in the domain knowledge. Therefore, this representation allows the construction of adaptive systems.

Figure 2.10 presents the business process of the relations networks corresponding with the care at discipline level. This process is usually performed by a hospital, however, the representation is valid for any collaborative interdisciplinary organization. The sub-process workflow fixes the 'natural' functionality of the system including the most relevant specialties in the T2DM care. The starting point in the care is frequently the General Practitioner. This health professional defines the disciplines needed for the care of the particular patient. The next step can be the emergency discipline, the diagnosis support disciplines (i.e. laboratory and imaging), or the other T2DM medical specialties. The diamond-shaped elements with the cross are exclusive gateways (only one path can be taken) and the diamond-shaped elements with the circle are inclusive gateways (many paths can be taken). A special case is given by preventive disciplines, which can be connected with other disciplines in different ways according to the organization and contextual policies.

The business process model presented in Figure 2.11 represents the GCM's aggregations granularity level of the care process. At this level, the process is usually performed by a health professional office. However, the representation is valid for any interdisciplinary organization offering health services. In the T2DM context, the collaboration between regulated healthcare providers and non-regulated

interdisciplinary care (e.g. home care) is frequently practiced. The health service can start from three care cases: preventive care, acute care or chronic care. The preventive care process is implemented in a heterogeneous way. Therefore, the details are hidden in order to keep generality. Observation and finding services are the first processes in the other cases.

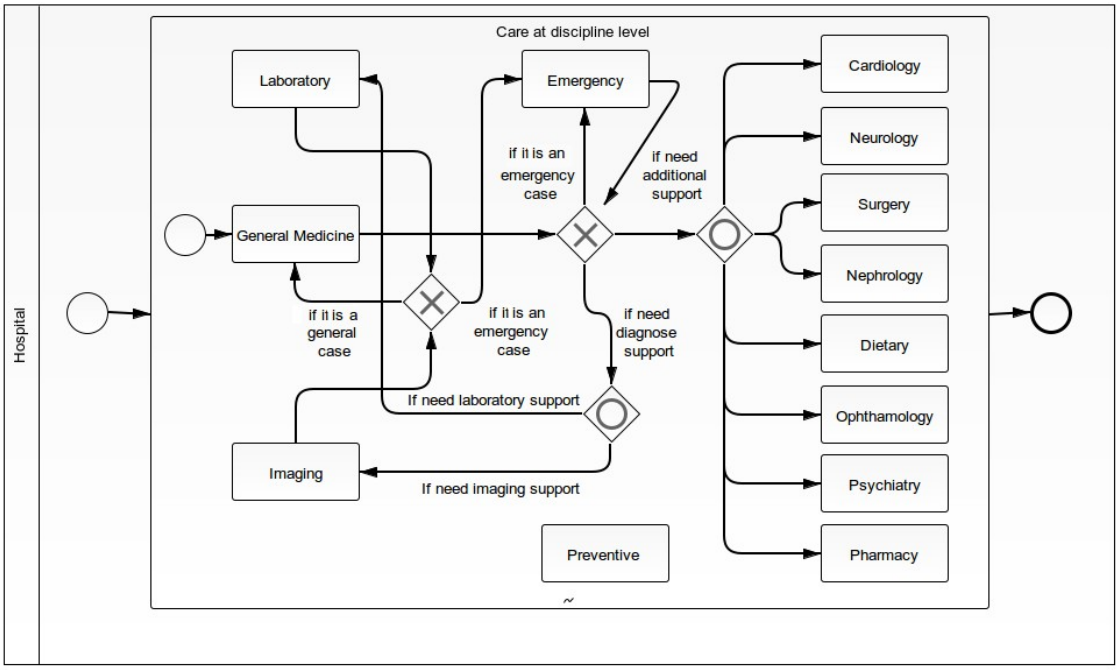

Figure 2.10: Relation network business process diagram

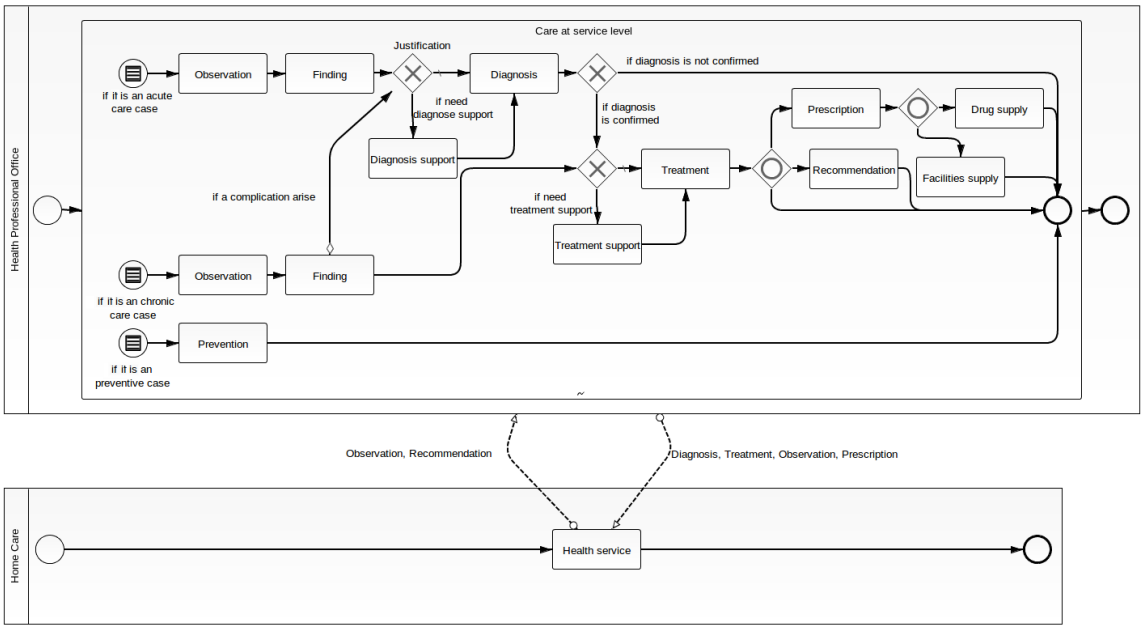

Figure 2.11: Business process diagram for GCM aggregations

Observation and finding process cannot be separated. In the case of acute care, after obtaining a finding, a decision is required whether the justification is sufficient or whether a diagnosis support service should be requested in order to finally obtain a justified diagnosis. If the diagnosis is confirmed, the treatment precedes the finding. This treatment can also require some additional support, usually provided by a different specialty.

Diagnosis and treatment support are the main collaboration points between disciplines. Then, in order to perform correctly a diagnosis or treatment support process, it is required to go up to the discipline level and to take a decision about the next step, according to the discipline's knowledge. This fact highlights again the need of the architectural consideration for correctly representing and executing the business process. After the treatment is finished, a recommendation or prescription can be provided. If there is some prescription, then it has to be delivered. This is done by the drug supply and/or the facilities supply processes.

The confirmed diagnosis of a chronic disease results in a chronic care case. In this case, the care constitutes a series of treatment events with its posterior process. If in a chronic case, the finding corresponds with a disease complication, then an acute care case arises.

Figure 2.12 shows the business process at a detailed granularity level. This corresponds to the care at task level and represents the tasks needed for accomplishing the services represented in the aggregation level. The relevant part of the diagram is the representation of the collaboration with the patient. Basically, the patient realizes self-observations and performs the prescription/recommendation execution. A special prescription execution is the self-monitoring, as this is the unique case where the patient is allowed to report self-observations without the direct presence of one health professional. The compliance task is a feedback to the health professional about the satisfactory execution of the prescription.

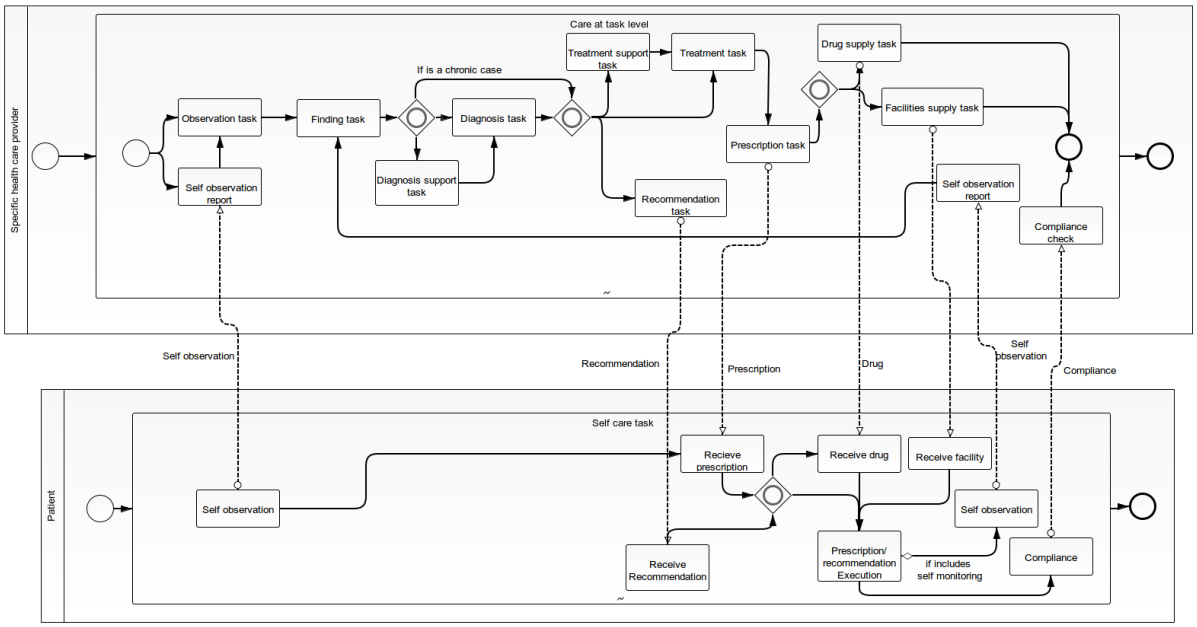

Figure 2.12: Business process diagram for GCM details

2.6 Discussion

This section highlights the achieved results and discusses related works.

2.6.1 The Importance of an Architectural Approach

The GCM framework extends the potential of traditional representations of domain entities such as UML diagrams or ontologies by providing a mechanism for explicitly representing the architectural aspects of systems (e.g. granularity levels) and considering multi-domain harmonization for enabling interoperability. The representation of the architectural aspects enables the description of the compositional nature of the modeled domain and at the same time hides the complexity by abstraction. According to the principles of the GCM framework [11], [126], GCM components can be interrelated only at the same level of granularity. So, the interrelation of components demands to go up to the level of common parents, as the context of a component in a system is provided by the compositionally related upper levels of granularity. That is similar to the connections in a tree. Two leaves from the same branch are connected directly, but the connection of two distant leaves needs a shared branch. It is also important to consider that the leaves are connected

to the branches and these to the trunk. As consequence, the complexity of the representation of a system decreases because that one component has only relationships with its neighbors, super-component and sub-components. Ignoring these architectural aspects will result in inconsistent inferences [127] or in unpredictable systems.

The presented diagrams consider the architectural aspects of the T2DM care system in a generic way. Based on these diagrams, it is possible to derive use case specific architectures and – if desired – to implement software solutions supporting them. Due to the generality as well as the consideration of architectural aspects and ontological descriptions, the solutions based on those architectures will be adaptive, intelligent and interoperable.

2.6.2 Related Works

For solving the lack of interoperability problem, there are many alternatives and works. In the following, works will be discussed that use an architectural approach, and thereafter, some alternative works are considered. The alternative works considered in this section deal with the integration of ontologies and BPMN.

- **Applications of the GCM Architectural Approach**

So far, the architectural approach provided by the GCM has been used for different purposes, but often with the intention to achieve interoperability. A set of applications based on the GCM architectural approach have dealt with the formalization of international standards. One example proposed a solution for automatic transformation among the different versions of the HL7 communication standard. This transformation is based on an architectural re-engineering of those standards, their formal representation and harmonization using a communication standards top level ontology [128]–[130]. This work facilitates the interoperability between these incompatible standards. Other examples are the HL7 Security and Privacy Domain Analysis Model, the HL7 Security and Privacy Ontology [131, p. 7], ISO standards 22600 [115] and 21298 [125], but also approaches to clinical models [132] or IT system analysis and design [76]. The architectural approach has also been used for the creation of a software development framework supporting HL7 specifications [76]. However, these works faced the lack of interoperability from a technical perspective, ignoring business process aspects and therefore the context.

[133], [134] use the architectural approach of the GCM for modeling an information

system in an obstetrics-gynecology department. This application describes the information flow within this department, but the GCM business view is not complete due the lack of an architectural description of the business process.

In [127], the architectural approach was used for proposing a mechanism for asserting the relationships in an ontology. The proposal demonstrates the importance of the architectural aspects in the ontology development.

Architectural approach in the context of ICT system analysis, design, and implementation are increasingly deployed. However, all of them ignore the architecture of the ICT-independent real world system [135].

- **The Integration of Ontologies into BPMN**

The integration of ontologies and business process modeling is often called semantic business process. In this field, several important articles have been published, e.g. Process Specification Language (PSL) [136], semantic case management [137], or semantic computer-interpretable guidelines [138]. In this dissertation, the work related to the BPMN standard is considered due to its wide acceptance by processes engineers and the ability to represent the process with graphical diagrams. The integration of BPMN and ontologies takes place in two different ways. The first one is the domain independent formalization of the BPMN semantics through ontologies, and the second one is the use of domain specific ontologies for classifying the objects represented in a particular model. Proposals like [139], [140] only focus on the formalization, presenting an ontology for notation. These works were carefully built and intended to cover all the terminologies expressed in the standard. However, it did not follow any ontological framework (e.g. upper level ontologies), and therefore, it works more like a mind map.

Paper [141] discussed the compatibility of upper level ontologies with BPMN 2.0, considering BFO, SOWA and BWW as main options. The final conclusion of the work was that no upper level ontology meets all the requirements of BPMN. Other work only considers the classification of the objects in the model. For example, in [142] BPMN 2.0 is used for modelling adaptive processes through the use of an ad-hoc sub-process. The rules for selecting paths in the adaptive process at run time are fixed in a Drool system. The authors used SWRL for medicine-specific rules (e.g., patient diagnosis, treatment). Ontologies have been used for describing the important domain concepts. The SWRL rules operate on these concepts. For specific applications, a clinical context ontology was implemented. The system architecture

presented in that paper looks promising as some of the proposed modules lend themselves to reuse. Other examples are [143], [144], which proposed a graphical annotation in the BPMN diagrams in order to improve re-usability and business process analysis, although they ignored its execution.

Some authors have focused on the BPMN formalization and the classification of the represented objects. An integration of BPMN 2.0 with the WSMO studio tool (framework for semantic web services) is presented in [145]. The WSMO studio tool and the background ontologies and languages are part of the SUPER project [146], [147]. This tool is not maintained by the project anymore, and its community is weak.

So, the support of the integrating BPMN 2.0 and the OWL language is weak as well. Paper [148] demonstrates the implementation of clinical guidelines using the CP ontology and BPMN 1.1. The process finally runs on the IBM Lombardy Engine. The CP ontology provides healthcare specific meaning for the activities. Finally, the work [149] proposes a rule-based procedural semantics for a relevant fragment of BPMN. The semantics defines state transitions and specifies state changes in terms of preconditions and effects. The paper also shows how the procedural process knowledge can be seamlessly integrated in the domain knowledge specified by using the rule-based ontology language OWL-RL [150]. The authors offer a tool, based on a framework to support the semantics, providing a wide range of reasoning services by using standard logic programming inference engines. Unfortunately, the tool is not fully compatible with the BPMN 2.0 specification, and the OWL-RL profile presents some expressivity restrictions not desired at the design time [151].

All aforementioned solutions differ from our approach, as they start from the ICT process, thereby the real world system architecture is not reflected. Therefore, existing real-world domain ontologies have not been mapped according to the architectural systems' requirements. Even more, there is a tendency to develop domain ontologies in an ad-hoc manner, largely ignoring existing domain and top-level ontologies, best practice guidelines [26], [152], as well as nearly twenty years of Applied Ontology research [153].

2.7 Conclusions

The presented approach enables comprehensive interoperability, also integrating the non-ICT aspects that have been ignored in most if not all alternative solutions. The architecture-centric approach considers the compositional nature of the real world

system and its functionalities in the sense of a system-theoretical White Box approach, and therefore, guarantees coherence, providing correct inferences. The consideration of the ontologies facilitates the harmonization between the different domains involved in the system. The level of generality used in the description facilitates the adaptive nature of the system. Finally, from the model presented for T2DM care, intelligent, adaptive and interoperable systems can be derived. However, this generic architecture is not implementable due its level of generality. So, use case specific specialized architectures need to be defined for starting the development process. This issue will be taken in the next chapter considering the architecture of the three relevant use cases in the T2DM care.

Chapter 3

Specialized Architecture for Type 2 Diabetes Mellitus Care System

The present chapter describes the architecture for the T2DM care use case glycemic control in pharmacotherapy. The glycemic control is a relevant issue in diabetes care. It is important for mitigating the development of complications as retinopathy, nephropathy and neuropathy. Self-monitoring provides a feedback from the effects of lifestyle changes and pharmacological treatment, and it increases patient empowerment and adherence to treatment [154]–[156]. Usually, the glycemic control starts with a lifestyle intervention, but finally, a pharmacotherapy will be performed to keep the blood glucose levels as normal as possible [157]. The telemedicine intervention improves clinical effectiveness, reduces direct costs, increases productivity, and is by that way very cost-effective [158]. However, such solution is not widely implemented in the diabetes care yet, especially in the Colombian context. Finally, patient's education and training regarding physical activity and proper nutrition are usually the main part of any lifestyle intervention, and therefore inevitable. This has been demonstrated in improving the glycemic control for both prevention and treatment [159], [160]. Patient's education in the self-monitoring process helps to improve the feedback to the health professional and therefore accomplishing more effective interventions.

The context of the use cases modelled in this chapter is limited by the Colombian policies, specifically by those defined in national guidelines for T2DM [156].The system in this work is described according to the policies issued by the Colombian Ministry of Health and Social Protection such as approved medical guidelines, ethical

principles, the World Medical Association Declaration of Helsinki [161], security and privacy regulations as well as professional and administrative refrains, obligations, etc. [156], [162]–[164].

3.1 T2DM Care System Architecture for the Glycemic Control in Pharmacotherapy

The glycemic control process serves the purpose to keep the blood glucose level under a low risk threshold. Epidemiological studies define the threshold for the glycated hemoglobin (HbA1C) level at 7.0%. Higher values increase the risk for microvascular and macrovascular complications [165]–[167]. Optimal glycemic control is fundamental to the management of diabetes [154]. Lifestyle intervention is the most recommended mechanism to start the glycemic control [156], but for finally meeting the goals, pharmacotherapy is necessary [157]. Independently of the type of intervention, the feedback of the patient through the self-monitoring process allows for individualized glycemic targets and a personalized configuration of the intervention [154]. The health professional identifies from the patient data some relevant risk factors (alerts) for taking decisions in the treatment. This is especially important in the pharmacotherapy because of the need for reducing the medication side effects.

3.1.1 GCM Representation

The architecture for the glycemic control use case is specialization of the generic architecture of the T2DM [168].

Figure 3.1 shows the GCM representation of the medical care domain of this use case. At the Relations Networks level, the medical disciplines related with the glycemic control are: general medicine, internal medicine, endocrinology, emergency, nursing, and laboratory. Health services provided by those medical disciplines are exemplified in the Aggregations level. A comprehensive list of health services is given in Table 3.1.

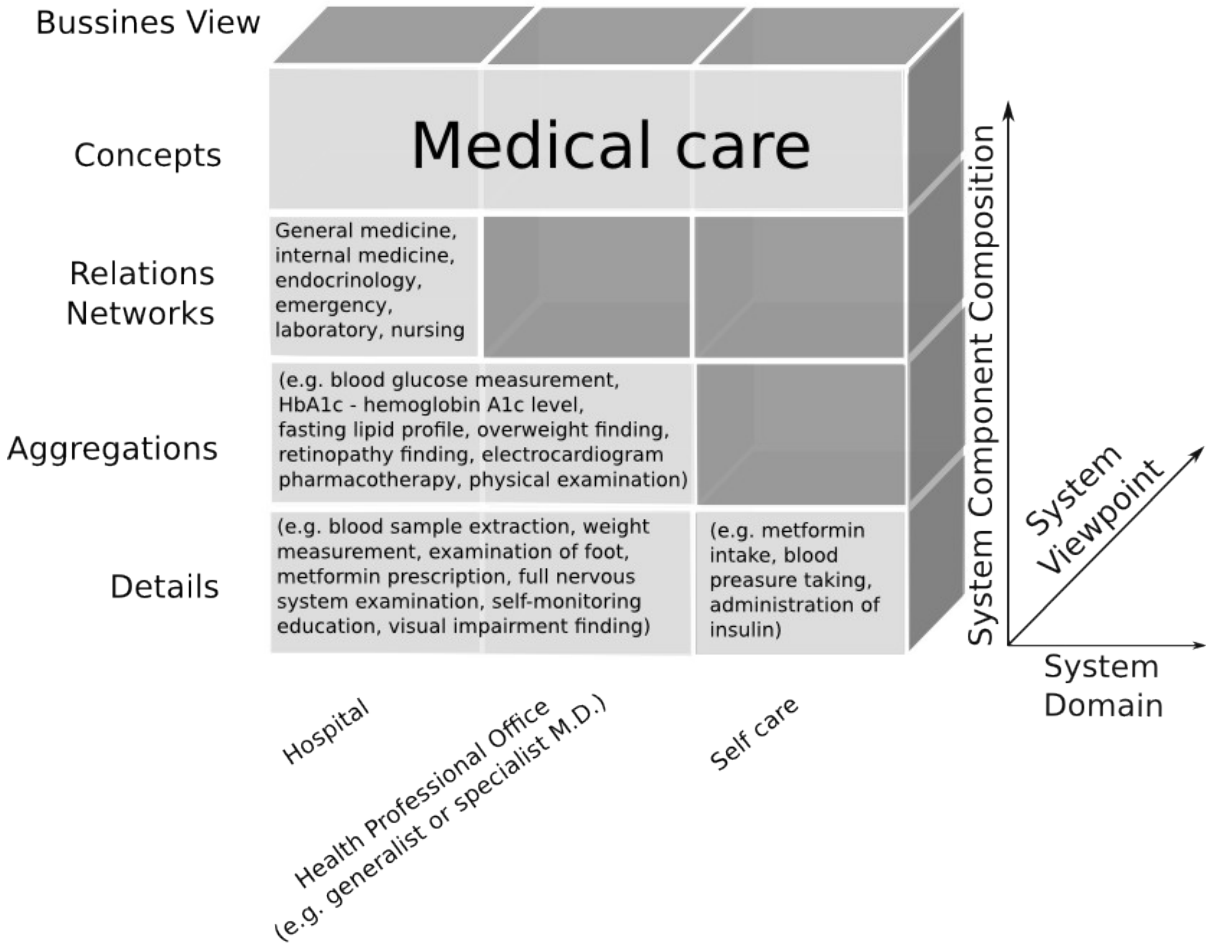

Figure 3.1: GCM representation of the medical care domain

Observations
- Clinical history evaluation
- Anamnesis
- Physical examination
- Blood glucose measurement
  - Fasting blood glucose measurement
  - Post-prandial blood glucose measurement
- Evaluation of self-monitoring of blood glucose
- HbA1c - Hemoglobin A1c level
- Fasting lipid profile
- Urinalysis
- Microalbuminuria measurement

Findings
- Overweight
- Systemic arterial hypertension
- Retinopathy
- Neuropathy
- Dyslipidemia
- Hypoglycemia
- Hyperglycemia
- Ventricular hypertrophy
- Peripheral arterial disease
- Coronary artery disease
- Nephropathy
- Diabetic foot

- Creatinine serum measurement
- Electrocardiogram

Treatments
- Pharmacotherapy

Recommendations
- Self-monitoring recommendation
- Patient education[2]
- Diagnosis support

Facilities supply
- Glucometer supply
- Lancet supply
- Blood testing strips supply
- Insulin injector supply
- Needle for insulin injector supply
- Orthopedic device supply
  - Stick supply
  - Walker supply

Prescriptions
- Drug prescription[1]

Drug supply
- Drug supply

Table 3.1: Health services in the glycemic control

These health services are composed of more specific task. Many of them have specific names in the medical domain and are represented at the GCM Details level. In Table 3.2, two examples are shown:

1 The list of drugs used in the glycemic control will be introduced below
2 Due its complexity is considered in a separate use case

Physical examination	Fasting blood glucose measurement
• General Inspection • Observation of vital signs ◦ Temperature ◦ Pulse ◦ Breath frequency ◦ Arterial blood pressure • Measuring height of patient • Weight and body mass assessment procedure • Measurement of circumference of waist • Random blood glucose measurement • Examination of head and neck ◦ Ophthalmoscopy ◦ Oral examination ◦ Ear, nose and throat examination ◦ Examination of neck • Cardiovascular physical examination • Examination of respiratory system • Exploration of abdomen • Exploration of skin • Examination of foot • Full nervous system examination	• Fasting time • Blood sample extraction • Blood sample sending • Blood analysis • Test results reporting

Table 3.3: Physical examination and fasting blood glucose measurement

Figure 3.2 represents the GCM policy domain specialized from the generic T2DM care model [168] for the glycemic control use case. Policies comprise legislation, administrative regulations, discipline-specific regulations (incl. clinical guidelines), contextual, environmental, and ethical rules including security and privacy related ones. For the glycemic control use case, the policy domain is divided in three sub-domains: clinical guidelines, security and privacy, and administrative. The clinical guidelines sub-domain includes rules for the behavior of the medical domain operating in a defined context. The medical guidelines have been defined by the Colombian Ministry of Health and Social Protection [156]. Each health organization prunes these guidelines for its implementation. These medical guidelines include alert signs for the correct glycemic control.

In the security and privacy sub-domain, the rules for assuring the integrity of the patient and his information as well as for privacy are defined. In order to standardize those rules, the Colombian government has defined the patient security guide [164] and the law 1581 of 2012 [162], also known as habeas data law. At the GCM's Relations Networks level, there are the Colombian Political Constitution [169], the General System of the Social Security in Health (SGSSS) laws [170]–[172], the medical ethics law (law 23 of 1981) [163] and the World Medical Association (WMA)

Declaration of Helsinki [161] regulating all related medical disciplines and ruling medical guidelines, and finally the security and privacy policies.

The administrative sub-domain defines the policies regulating the behaviour of organizations including the administration of the resources. The Colombian Political Constitution [169] and the SGSSS laws [170, p. 1], [171], [172] provide the political framework for any health related organization in Colombia. At the GCM's Aggregations level, organizational contracts, organizational values and procedure manuals are defined. These policies are composed of organizational rules, located at the GCM's Details level. Organizational contracts and procedure manuals define the structural roles assigned to the actors in the care process [125]. Usually, organizational value statements define some ethical principles for the procedures running in the organization.

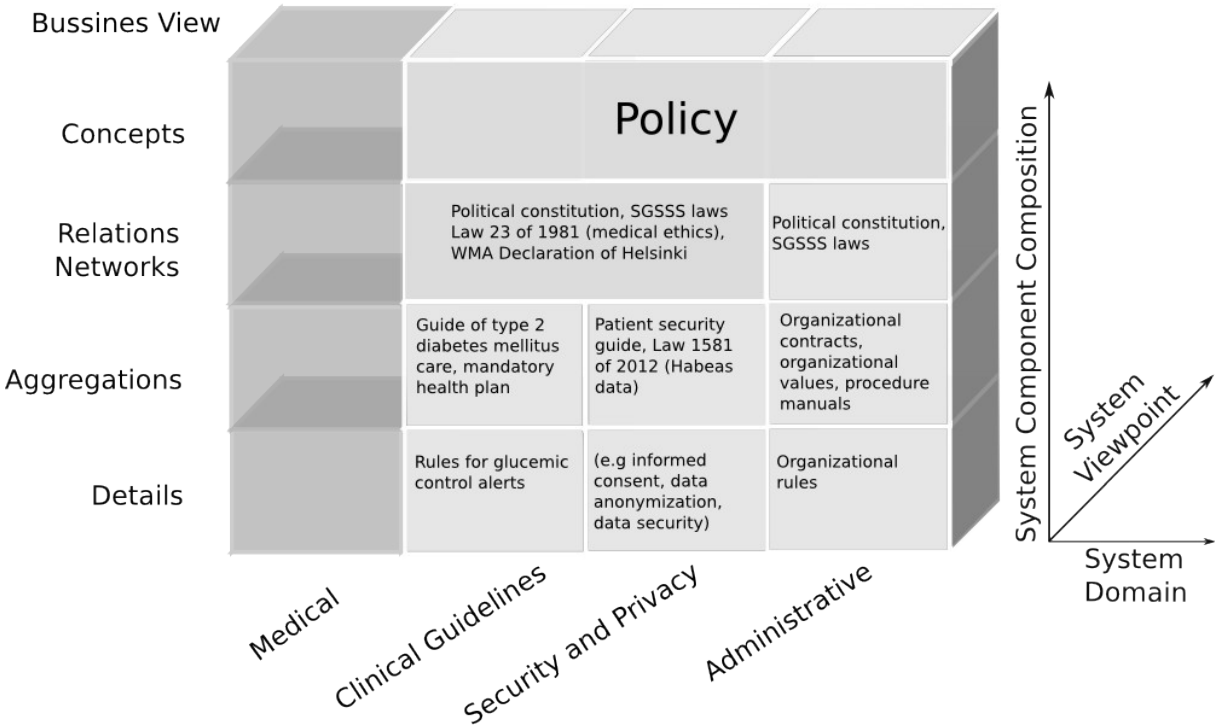

Figure 3.3: GCM representation of the policy domain

Figure 3.3 represents the GCM resource domain specialized from the generic T2DM care model [168] for the glycemic control use case of the T2DM system.

The resource domain is divided into three sub-domains: actor, facility, and location

[168]. In the actor sub-domain, the GCM's relation networks correspond to the organizations in charge of the glycemic control, these are: hospitals, medical professional offices and home care organizations. In the GCM's Aggregations level, the acting components of the organizations, i.e. persons, application and devices, have been defined. The person actor can be specialized for the glycemic control use case to: general practitioner, nurse, internist, endocrinologist, bacteriologist, primary caregiver, secondary caregiver or patient. Often, IT systems are involved in the glycemic control such as Personal Health Record Systems (PHR-S), Electronic Health Record Systems (EHR-S), Picture Archiving and Communication System (PACS) and Laboratory Information Systems (LIS). Contrary to other countries, active devices are currently not broadly used for the glycemic control in Colombia. At the GCM's Details level, parts of the applications are defined. These parts are dependent of the application architecture. In general however, it is possible to identify parts as certain functions such as the graphical interface or the application database.

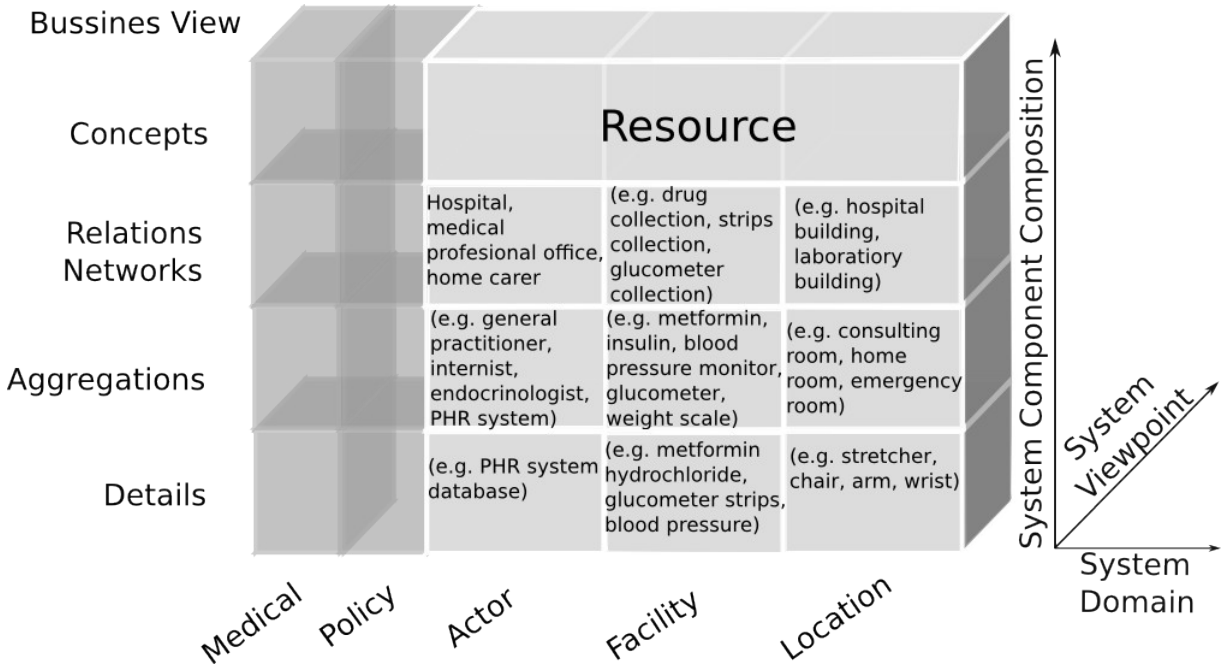

Figure 3.4: GCM representation of the resource domain

Equipment	Drugs
• Glucometers • Lancets • Blood testing strips • Insulin injector device (syringe or pen) • Glasses • Weight scales • Measuring tape • Blood pressure monitor • Pulse oximeter • Stethoscope • Electrocardiograph • Tuning fork 128 Hz • Reflex hammer • Semmens-Weinstein monofilament • Ophthalmoscope • Special footwear • Magnifying glass • Lamp • Thermometer • Orthopedic devices	• Metformin • Sulfonylurea ◦ Glimepiride • Acarbose • Dipeptidyl peptidase IV inhibitors ◦ Linagliptin ◦ Saxagliptin ◦ Vildagliptin • Insulin ◦ Short-acting insulin analogues ▪ Lispro insulin ▪ Insulin glulisine ▪ Insulin aspart ◦ Short-acting insulin ▪ Regular insulin ▪ Cristaline ▪ Actrapid ◦ Intermediate-acting insulin ▪ Isophane insulin (NPH) ◦ Long-acting insulin ▪ Insulin glargine ▪ Insulin detemir

Table 3.4: Drugs and equipment used in the glycemic control

In the facility sub-domain, the collection of drugs and equipment has been defined in the GCM's generic T2DM care relations networks [168]. The list of drugs and equipment used in the glycemic control is presented in Table 3.3.

3.1.2 Class Diagram

In this section, the UML class diagrams for the different domains involved in the glycemic control of the T2DM care are presented. Figure 3.4 corresponds to an extract of the classes in the medical domain focused on physical examination. As described in [168], the medical care processes include processes related to the different medical specialties. These processes are the aggregation of some healthcare services. An example for those services is the physical examination process, which is composed of the tasks presented in the Table 3.2 (section 4.1.1).

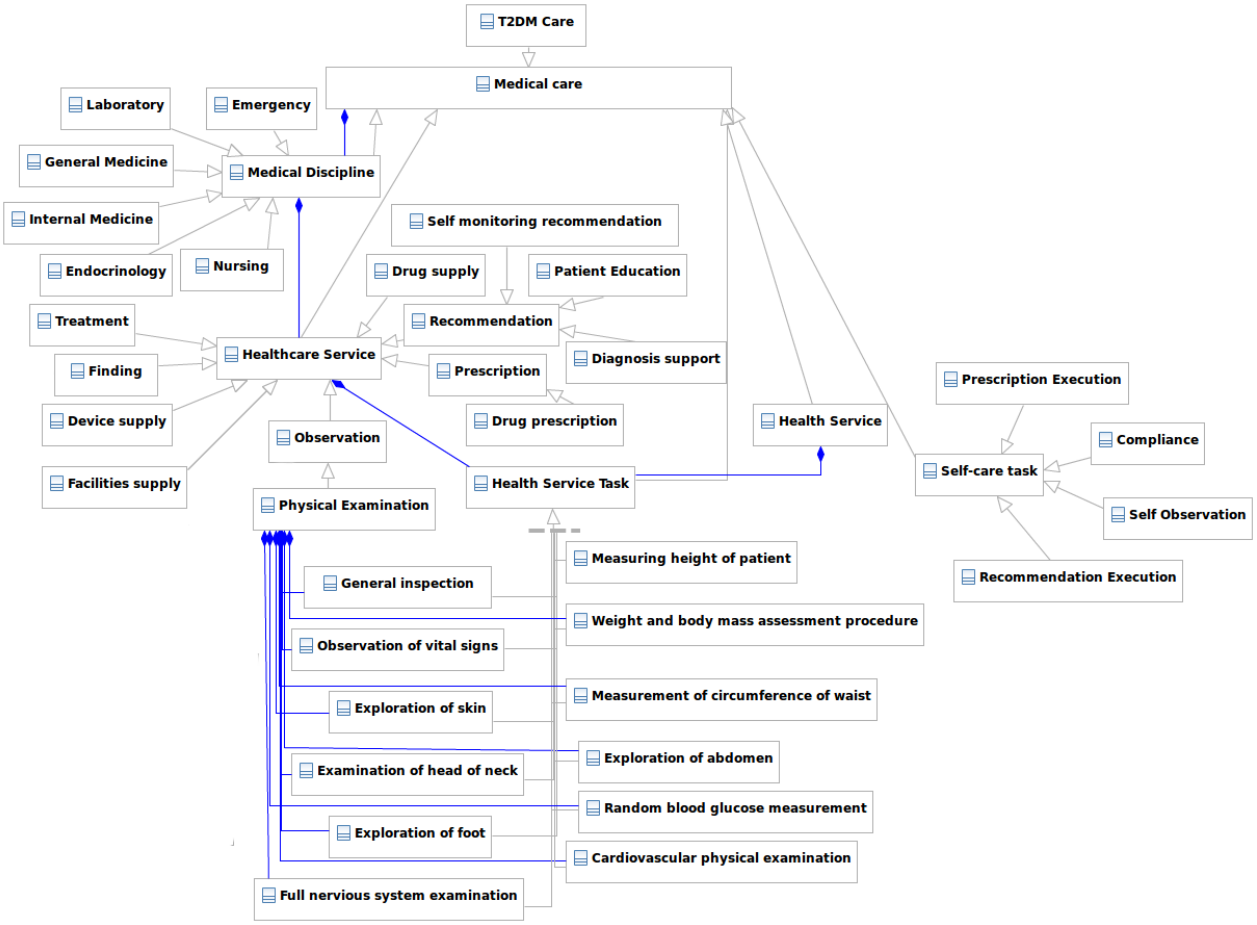

Figure 3.5: Classes of the physical examination in the glycemic control

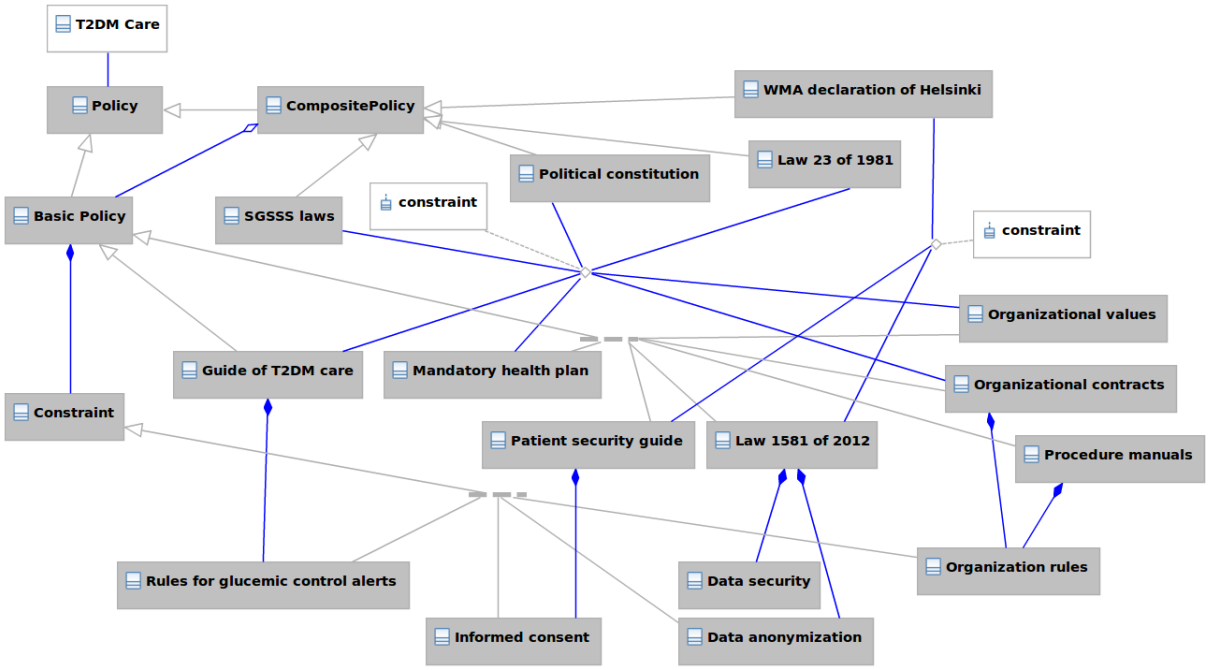

Figure 3.6: Classes for the policy domain in the glycemic control

Figure 3.5 shows the classes for the policy domain in the glycemic control use case. The political constitution, the law 23 of 1981, the WMA declaration of Helsinki, and the SGSSS laws impact the policies defined in the lower levels. All the policies need to be coherent with their upper level policies, inheriting their basic principles. WMA declaration of Helsinki and the Law 23 of 1981 defined by the Colombian government are the top level policies, as they declare the ethical principles governing the care process.

The Colombian Political Constitution also declares other principles defining the framework for all the legal policies in the country. The SGSSS laws govern the function of the health system in Colombia, and by this way also the care system. An important regulation in the Colombian health system is the mandatory health plan. This constrains the procedures, drugs and facilities that can be provided to the patient through the health promoter entities. Patient security guide and the Law 1581 of 2012 are policies defining principles for the security and privacy of the patient and his information. They include important constraints such as the informed consent, data security and data anonymization. The Colombian guide of T2DM care contains all the medical aspects of the system. For the glycemic control use case, it defines the rules for glycemic control alerts. Each health provider organization in the system defines

internal polices such as: organizational values, organizational contracts and procedure manuals. These policies include the rules applied in the organization to perform the procedures and to constrain the resources associated with the organization.

The resources used in the glycemic control use case are represented by the classes shown in Figure 3.6.

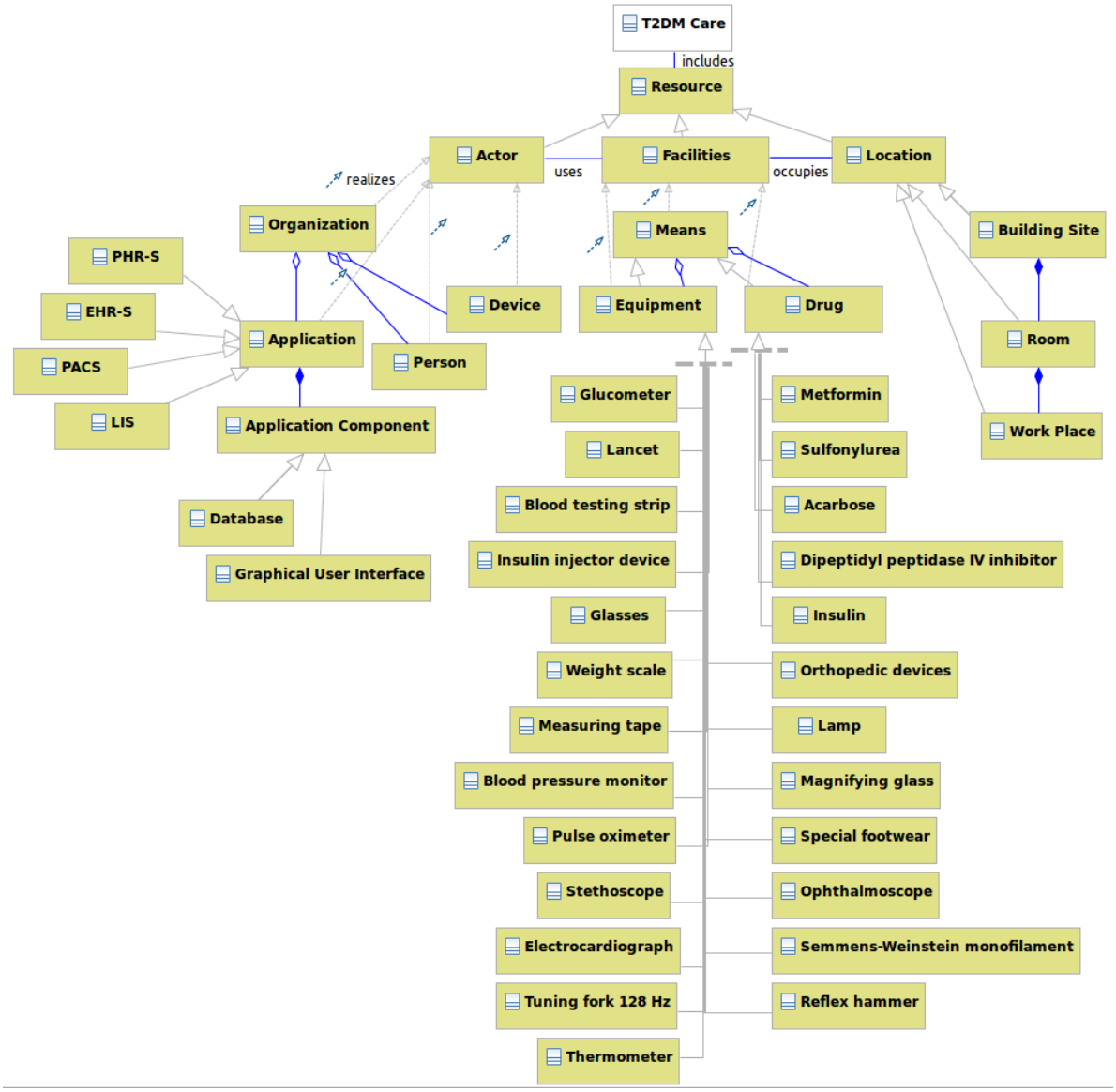

Figure 3.7: Classes for the resource domain in the glycemic control

The classes in the figure represent the elements mentioned in Section 3.1.1. It is important to highlight that the elements under the class Equipment are objects used by an actor in order to perform an activity. These objects require the direct operation of an actor. For the explanation of the general classes, the reader is referred to [168].

Figure 3.7 describes the relation between the three domains and provides an example of the interactions. In general, the policies govern the behavior of the system by constraining functionalities and relationships of the components. An example is the Guide of T2DM care policy that regulates the tasks performed in a physical examination for glycemic control. It also contains the rules for the glycemic control alerts that defines the thresholds for the normal observations, e.g. in the exploration of foot. Other examples are the inter-organizational policies, i.e. procedure manuals, organizational value statements, and organizational contracts. These policies constrain the behavior of the actors in the organization. For example, a medical doctor is contracted to perform activities only in the hospital emergency department.

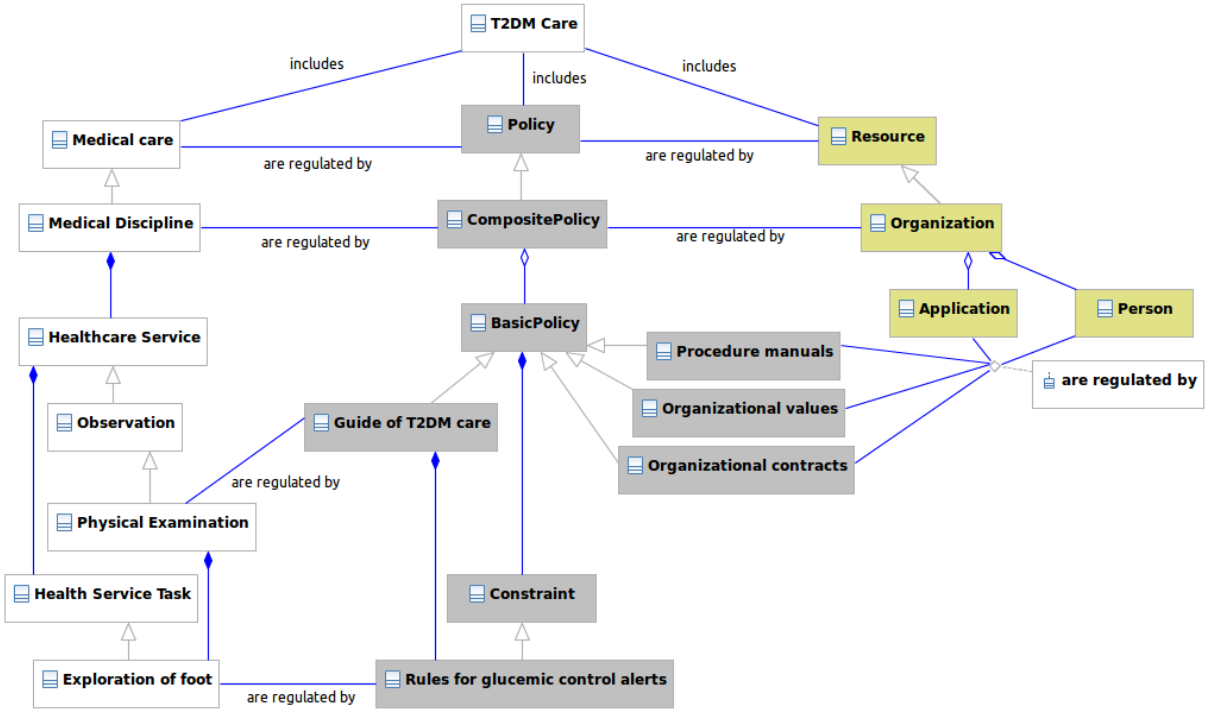

Figure 3.8: Inter-domain relationships in the glycemic control

3.1.3 Business Process Representation

Based on the generic business process model of the T2DM care [168], a specialized model for the glycemic control use case has been derived. The general medical specialties considered in the generic architecture were restricted to the related use case, i.e. general medicine, laboratory, imaging, emergency, internal medicine, endocrinology and dietary. The dietary specialty must collaborate in the glycemic control despite the patient is treated with a pharmacological means. Figure 3.8 shows the expected medical flow for glycemic control at the medical specialties level. The internal medicine specialty plays an important role in the disease treatment due its holistic and deep view on the body metabolism.

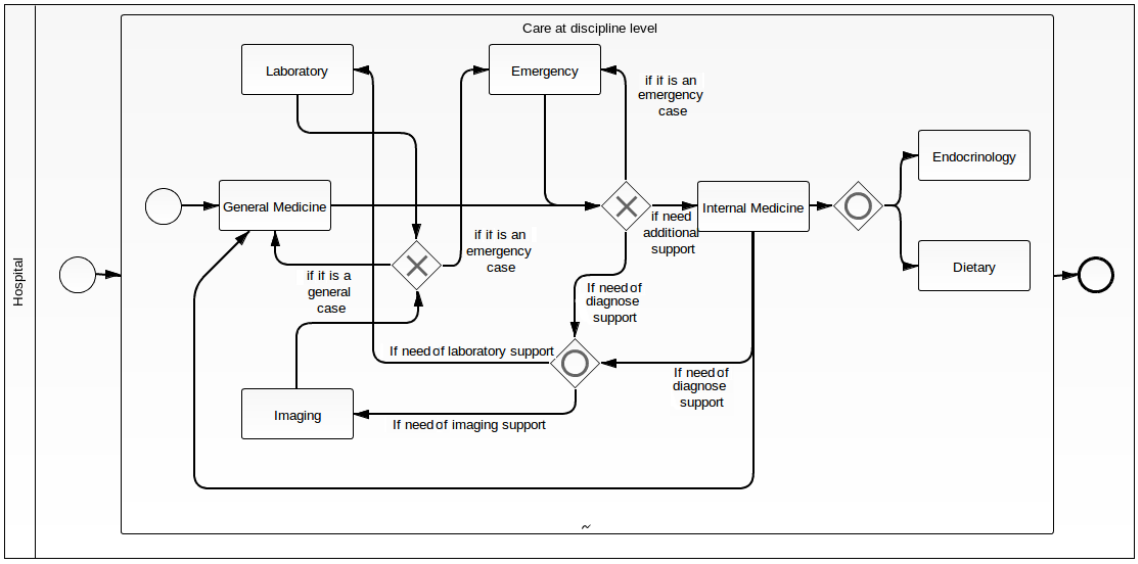

Figure 3.9: GCM's Relation networks business process model

As presented in the GCM and UML diagrams, the medical specialty processes are composed of a set of healthcare services. The healthcare services related with the glycemic control use case are presented in Table 1. Figure 3.9 describes the business process for the observation healthcare service.

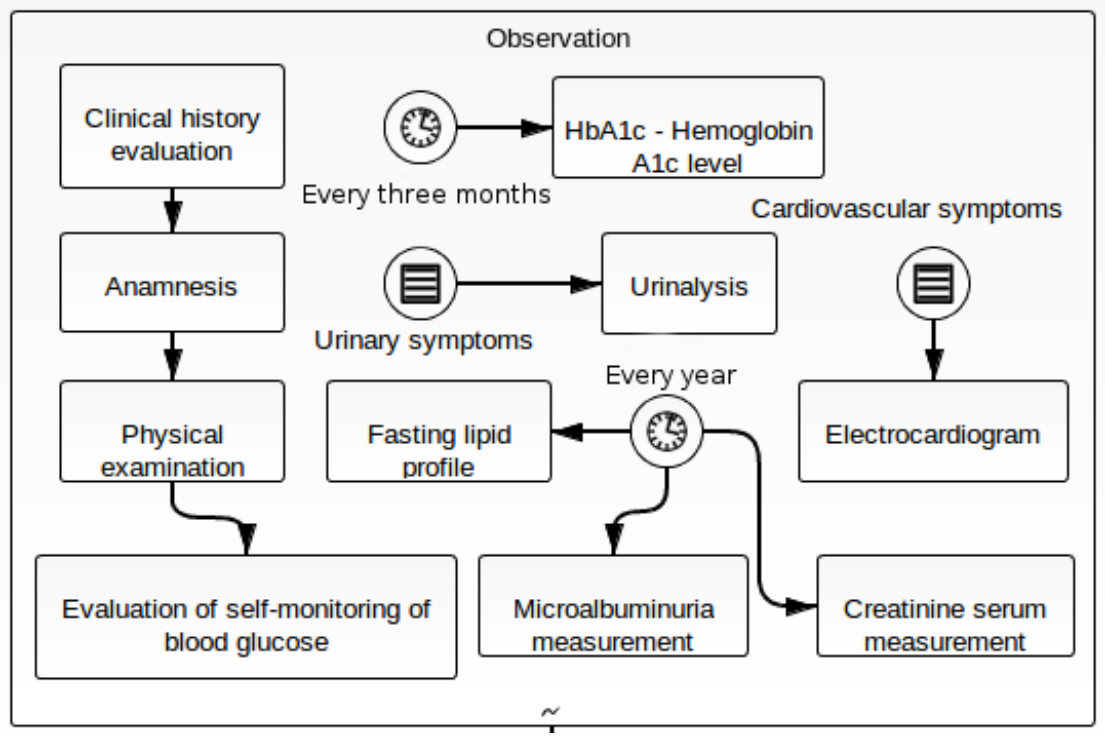

Figure 3.10: Observation business process at the GCM's aggregation level

Usually, the first process in any medical encounter is the clinical history evaluation. Following, the anamnesis or interrogatory is performed, followed by the physical examination. These three first steps are generic for any use case in the medical domain. However, in the glycemic control they are adapted according to the goals. After the physical examination, the evaluation of the self-monitoring measurements is realized. The other observations in the figure are triggered by special events. The HbA1c is controlled quarterly, while the fasting lipid profile, the microalbuminuria and the creatinine serum are checked each year. Urinalysis and electrocardiogram are performed only if cardiovascular or urinary symptoms are present.

The physical examination performed in the glycemic control is composed of the procedures listed in Table 3.2 and their execution order is presented in Figure 3.10. The main goals of those examinations are to check the general health status, to check the fulfillment of the glycemic goal, and to avoid the complications associated with T2DM.

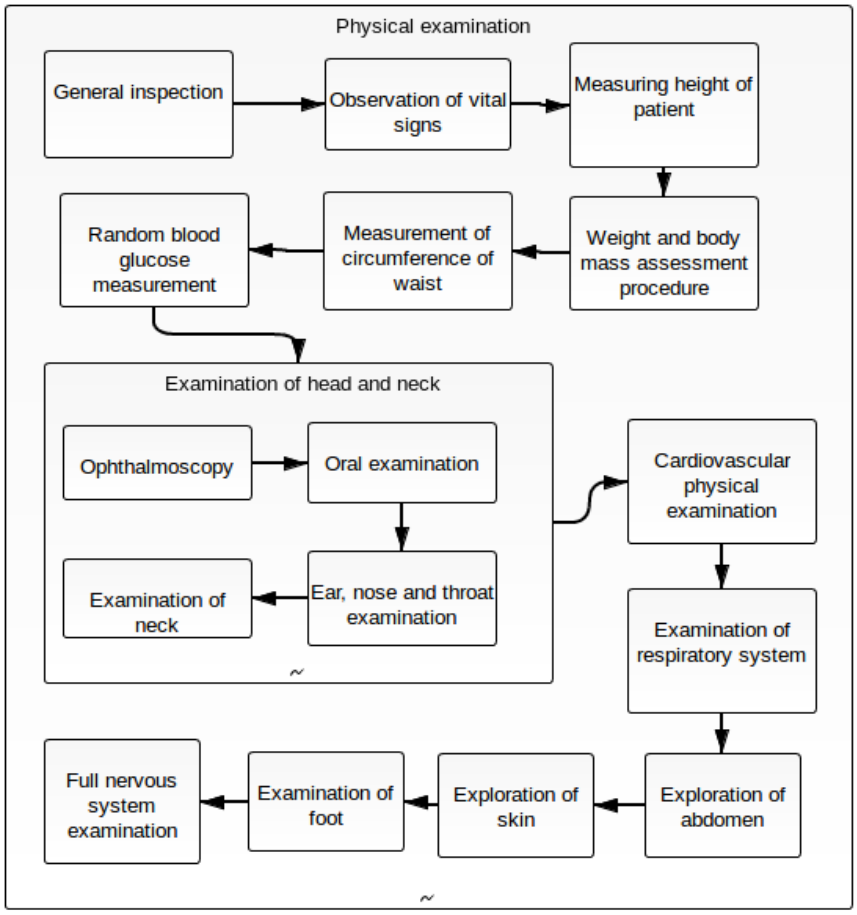

Figure 3.11: Physical examination business process at the GCM's aggregation level

3.1.4 Business Rules and Policies

In this section some policies are defined using the SPIN rule language and the type 2 diabetes mellitus care ontology (dm2co). Table 3.4 presents two example rules for processing blood glucose measurement results. The left one represents the hyperglycemia finding case, and right one illustrates the hypoglycemia case. The first one corresponds with an alert situation and the second one with an emergency situation.

```
# If  (greater than 200 mg / dL) then Hyperglycemia finding (alert case)
CONSTRUCT {
  ?id btl2:isPartOf ?patientLife .
  ?id btl2:hasCondition ?id .
  ?id a dm2co:Hyperglycemia .
  ?id a dm2co:MedicalAlert .
  ?id rdfs:label ?cause_type_en .
  ?id rdfs:label ?cause_type_es .
  ?this btl2:represents ?id .
}
WHERE {
  ?patient btl2:isBearerOf ?blood_glucose .
  ?patient btl2:hasLife ?patientLife .
  ?this btl2:represents ?blood_glucose .
  ?blood_glucose a dm2co:BloodGlucoseConcentration .
  ?this dm2co:hasValueIn_mg_dL ?value .
  FILTER ((?value >= 200.0) && (?value < 300.0)) .
  OPTIONAL {
    ?clonAlert a dm2co:MedicalAlert .
    ?this btl2:represents ?clonAlert .
  } .
  FILTER (!bound(?clonAlert)) .
  BIND (STRLANG("hyperglycemia medical alert", "en") AS ?cause_type_en) .
  BIND (STRLANG("alerta médica por hiperglucemia", "es") AS ?cause_type_es) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#", STRUUID())) AS ?id) .
}
```

```
# If  (less than 50 mg / dL) then Hypoglycemia finding (emergency case)
CONSTRUCT {
  ?id btl2:isPartOf ?patientLife .
  ?id btl2:hasCondition ?id .
  ?id a dm2co:Hypoglycemia .
  ?id a dm2co:MedicalEmergency .
  ?id rdfs:label ?cause_type_en .
  ?id rdfs:label ?cause_type_es .
  ?this btl2:represents ?id .
}
WHERE {
  ?patient btl2:isBearerOf ?blood_glucose .
  ?patient btl2:hasLife ?patientLife .
  ?this btl2:represents ?blood_glucose .
  ?blood_glucose a dm2co:BloodGlucoseConcentration .
  ?this dm2co:hasValueIn_mg_dL ?value .
  FILTER (?value <= 50.0) .
  OPTIONAL {
    ?clonEmergency a dm2co:MedicalEmergency .
    ?this btl2:represents ?clonEmergency .
  } .
  FILTER (!bound(?clonEmergency)) .
  BIND (STRLANG("hyporglycemia medical emergency", "en") AS ?cause_type_en) .
  BIND (STRLANG("emergencia médica por hipoglucemia", "es") AS ?cause_type_es) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#", STRUUID())) AS ?id) .
}
```

Table 3.5: Rules for the blood glucose measurement results

The alert situation implies that the patient needs an attention by the medical doctor as soon as possible. The emergency situation implies that the patient must be attended immediately by an emergency health provider. In our use case, this conditions creates a message as shown in Table 3.5.

Table 3.6 presents some examples of security and privacy policies. On the left side, a patient security rule is demonstrated. This rule requires a hand-washing activity in the workflow plan prior to any physical examination. The right side rule requires a patient authorization process prior to any clinical history evaluation.

Other examples of glycemic control alerts are represented in Table 3.7. This case corresponds to the blood pressure result alerts. The left rule relates to an alert by a hypertension situation represented in a diastolic blood pressure measurement result. The right rule relates to an alert by a hypotension situation represented in a systolic blood pressure measurement result.

```
# If  an medical alert occur then send an alert message
CONSTRUCT {
  ?messageId a btl2:InformationObject .
  ?messageId rdfs:label "message"@en .
  ?messageId dm2co:hasValue ?message .
  ?recipientId a btl2:InformationObject .
  ?recipientId rdfs:label "recipient"@en .
  ?recipientId dm2co:hasValue ?email_address .
  ?planId a dm2co:SendMessageByEmailPlan .
  ?planId btl2:hasPart ?messageId .
  ?planId btl2:hasPart ?recipientId .
  ?planId rdfs:label "send message by email plan" .
}
WHERE {
  ?this btl2:isPartOf ?patientLife .
  ?patient btl2:hasLife ?patientLife .
  ?patient a dm2co:HumanOrganism .
  ?result btl2:represents ?this .
  ?result dm2co:hasValue ?value .
  FILTER (!isNumeric(?value)) .
  ?result rdfs:label ?result_type .
  FILTER (lang(?result_type) = ?doctor_language) .
  ?this rdfs:label ?cause_type .
  FILTER (lang(?cause_type) = ?doctor_language) .
  ?patient btl2:isRepresentedBy ?identification_document .
  ?identification_document btl2:hasPart _:0 .
  _:0 rdfs:label ?personal_name_label .
  FILTER (?personal_name_label = STRLANG("personal
name", "en")) .
  _:0 dm2co:hasValue ?patient_name .
  ?identification_document btl2:hasPart _:1 .
  _:1 rdfs:label ?identification_number_label .
  FILTER (?identification_number_label =
STRLANG("identification number", "en")) .
  _:1 dm2co:hasValue ?patient_identification .
  _:2 btl2:hasParticipant ?patient .
  _:2 a btl2:Process .
  ?diabetes_care_plan btl2:hasRealization _:2 .
  ?diabetes_care_plan a
dm2co:Type2DiabetesMellitusCarePlan .
  ?diabetes_care_plan btl2:hasPart ?
doctor_email_information .
  ?doctor_email_information rdfs:label ?email_label .
FILTER (?email_label = STRLANG("medical doctor email",
"en")) .
  ?doctor_email_information dm2co:hasValue ?
email_address .
  ?diabetes_care_plan btl2:hasPart ?
doctor_language_information .
  ?doctor_language_information rdfs:label ?
doctor_language_label .
  FILTER (?doctor_language_label = STRLANG("medical
doctor preferred language", "en")) .
  ?doctor_language_information dm2co:hasValue ?
doctor_language .
  BIND (IF((?doctor_language = "en"),
STRLANG(fn:concat("The patient ", ?patient_name, "
identified by the number ", ?patient_identification, " has an ",
?cause_type, ", ", ?result_type, " value = ", ?value), ?
doctor_language), IF((?doctor_language = "es"),
STRLANG(fn:concat("El paciente ", ?patient_name, "
identificado con el número ", ?patient_identification, "
presenta un ", ?cause_type, ", ", ?result_type, " valor = ", ?
value), ?doctor_language), owl:Nothing)) AS ?message) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#",
STRUUID())) AS ?planId) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#",
STRUUID())) AS ?messageId) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#",
STRUUID())) AS ?recipientId) .
}
```

Table 3.6: Rules for generation of alert messages

```
# if physical examination is planned then handwashing is
planned before
CONSTRUCT {
  ?this btl2:hasPart _:b0 .
  _:b0 a bpmn:SequenceFlow .
  _:b0 btl2:hasComponentPart _:b1 .
  _:b1 a bpmn:SequenceFlow_Target .
  _:b1 btl2:represents ?physical_examination_plan .
  _:b0 btl2:hasComponentPart _:b2 .
  _:b2 a bpmn:SequenceFlow_Source .
  _:b2 btl2:represents _:b3 .
  _:b3 a dm2co:HandwashingPlan .
}
WHERE {
  ?this btl2:hasPart ?physical_examination_plan .
  ?physical_examination_plan a
dm2co:PhysicalExaminationPlan .
}
```

```
# Before a clinical history evaluation a patient authorization is
needed
CONSTRUCT {
  ?this btl2:hasPart _:b0 .
  _:b0 a bpmn:SequenceFlow .
  _:b0 btl2:hasComponentPart _:b1 .
  _:b1 a bpmn:SequenceFlow_Target .
  _:b1 btl2:represents ?clinical_history_plan .
  _:b0 btl2:hasComponentPart _:b2 .
  _:b2 a bpmn:SequenceFlow_Source .
  _:b2 btl2:represents _:b3 .
  _:b3 a dm2co:PatientAuthorizationPlan .
}
WHERE {
  ?this btl2:hasPart ?clinical_history_plan .
  ?clinical_history_plan  a
dm2co:ClinicalHistoryEvaluationPlan) .
}
```

Table 3.7: Rules for security (left) and privacy (rigth)

```
# If  (Diastolic blood pressure greater than 90 mmHg) then Hypertension finding (alert case)
CONSTRUCT {
  ?id btl2:isPartOf ?patientLife .
  ?id btl2:hasCondition ?id .
  ?id a dm2co:Hypertension .
  ?id a dm2co:MedicalAlert .
  ?id rdfs:label ?cause_type_en .
  ?id rdfs:label ?cause_type_es .
  ?this btl2:represents ?id .
}
WHERE {
  ?patient btl2:isBearerOf ?blood_pressure .
  ?patient btl2:hasLife ?patientLife .
  ?this btl2:represents ?blood_pressure .
  ?blood_pressure a dm2co:DiastolicBloodPressure .
  ?this dm2co:hasValueIn_mmHg ?value .
  FILTER (?value >= 90.0) .
  OPTIONAL {
    ?clonAlert a dm2co:MedicalAlert .
    ?this btl2:represents ?clonAlert .
  } .
  FILTER (!bound(?clonAlert)) .
  BIND (STRLANG("hypertension medical alert", "en") AS ?cause_type_en) .
  BIND (STRLANG("alerta médica por hipertension", "es") AS ?cause_type_es) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#", STRUUID())) AS ?id) .
}
```

```
# If  (Systolic blood pressure less than 60 mmHg) then Hypotension finding (alert case)
CONSTRUCT {
  ?id btl2:isPartOf ?patientLife .
  ?id btl2:hasCondition ?id .
  ?id a dm2co:Hypotension .
  ?id a dm2co:MedicalAlert .
  ?id rdfs:label ?cause_type_en .
  ?id rdfs:label ?cause_type_es .
  ?this btl2:represents ?id .
}
WHERE {
  ?patient btl2:isBearerOf ?blood_pressure .
  ?patient btl2:hasLife ?patientLife .
  ?this btl2:represents ?blood_pressure .
  ?blood_pressure a dm2co:SystolicBloodPressure .
  ?this dm2co:hasValueIn_mmHg ?value .
  FILTER (?value <= 90.0) .
  OPTIONAL {
    ?clonAlert a dm2co:MedicalAlert .
    ?this btl2:represents ?clonAlert .
  } .
  FILTER (!bound(?clonAlert)) .
  BIND (STRLANG("hypotension medical alert", "en") AS ?cause_type_en) .
  BIND (STRLANG("alerta médica por hipotensión", "es") AS ?cause_type_es) .
  BIND (IRI(fn:concat("http://purl.org/unicauca/dm2co#", STRUUID())) AS ?id) .
}
```

Table 3.8: Rules for blood pressure results

3.2 Discussion

In this section, some features of the methodology and of the obtained results are highlighted. These features are: interdisciplinary methodology, completeness, adaptability, and intelligence.

3.2.1 Interdisciplinary Methodology

The presented methodology is based on the system theory [37] and inherited the ability of abstract entities as a set of components and relations. This abstraction takes different forms, but is common to many of the health related specialties. Therefore, the decomposition of the systems in its components using the GCM cuboid representation and the UML class diagrams can be understand by heterogeneous experts. The GCM representation helps to add the architectural conceptualization to all the other descriptions. The separation in domains is crucial in order to keep each expert in its discipline and to set the framework for the inter-disciplinary collaboration. An important feature of the methodology is the extensive use of standards and top-

level ontologies, which increases the probability of maintaining a better collaboration between the different actors.

3.2.2 Completeness

The recursive use of abstraction and granularity level separation improves the completeness of system description. These practices hide the complexity of the system, thereby keeping up the coherence with the system described at the desired level. Software systems developed using these principles are expected to be of better quality [173] and able to support more precisely the system outside the Information and Communication Technologies (ICT) world.

3.2.3 Adaptability

The shown approach seeks balance between the open world and the closed world assumptions. The open world statements are represented by ontologies and correspond to the future proof assertions. The closed world statements are represented by rules and correspond to the context dependent knowledge. Keep the open world statements independently of the closed world ones, helps to create a future proof system. The correct description of domains and contexts through the rules allows the flexibility of the system. For example, the presented system is developed using a description of the Colombian context, but it can be adapted to any country by the definition of its specific context.

3.2.4 Intelligence

The techniques and methodologies used in our proposal have well defined semantics. Therefore, computer systems are able to correctly reason on them. The methodology used help to extract correctly the knowledge of the experts and allow the system to be built on, and run the rules defined by, that knowledge. The presented architecture supports non-stochastic intelligence that is desired in most of the healthcare use cases.

3.3 Conclusions

This chapter provides an extract of a T2DM care system analysis, design and development process addressed in the Thesis. This extract focuses on the process management by structurally and functionally considering the system architecture perspective policy with its relations to medicine and resources.

Methodology and models used in the architecture design facilitate the inter-disciplinary communication and allow the development of intelligent systems taking into account the experts' knowledge and the relevant policies. The methodology allows considering relevant factors in order to improve the health of the T2DM patient such as clinical guidelines, alert conditions, patient security, and emergency management. Furthermore, the methodology creates modular systems capable to adapt to policy changes. Finally, this methodology facilitates the creation of decision support systems. All those issues are relevant for providing health services in problematic access areas, where the personal is not appropriately qualified.

Chapter 4

Implementation of the Type 2 Diabetes Mellitus Care System

In this chapter, the implementation process of a software pilot for the T2DM care is presented. The implementation process starts from the description provided in Chapter 3. The description of this chapter is restricted to the pharmacological glycemic control use case.

4.1 Implementation Methods

Currently, three implementation methods have been identified that could satisfy the principles of the GCM: The model-driven architecture approach, the semantic web approach, and hybrid approaches. Following, each approach is shortly described.

4.1.1 Model Driven Architecture Approach

Model driven architecture (MDA) [174] defines three different models: The computation independent model (CIM), the platform independent model (PIM), and the platform specific model (PSM). MDA proposes the automatic or semi-automatic transformation between these models, based on appropriate tooling. Atlas Transformation Language (ATL) and Query/View/Transformation Language (QVTL) have been defined to describe these transformations.

MDA models have a correspondence with the GCM viewpoints. CIM partially corresponds to the business viewpoint, and even more to the enterprise viewpoint, as those viewpoints are computation independent. However, the GCM business viewpoint describes a real world system independent of ICT ontologies, while MDA

establishes an ICT development process. PIM corresponds to the informational and computational viewpoints, which are independent of any platform. PSM correspond to the technology and engineering viewpoints which relate to a specific platform.

Figure 4.1 describes the common process according to MDA [6], [175]–[179]. The description of the system is divided in three aspects: structural, behavioral and functional aspects. Structural (static) aspects describe time independent statements about the system. Behavioral (dynamic) aspects describe the plan of execution for the system. Functional aspects describe the purpose of the system. The CIM describes the business process to be supported by the ICT solution, while the PIM describes the ICT system, and PSM its implementation.

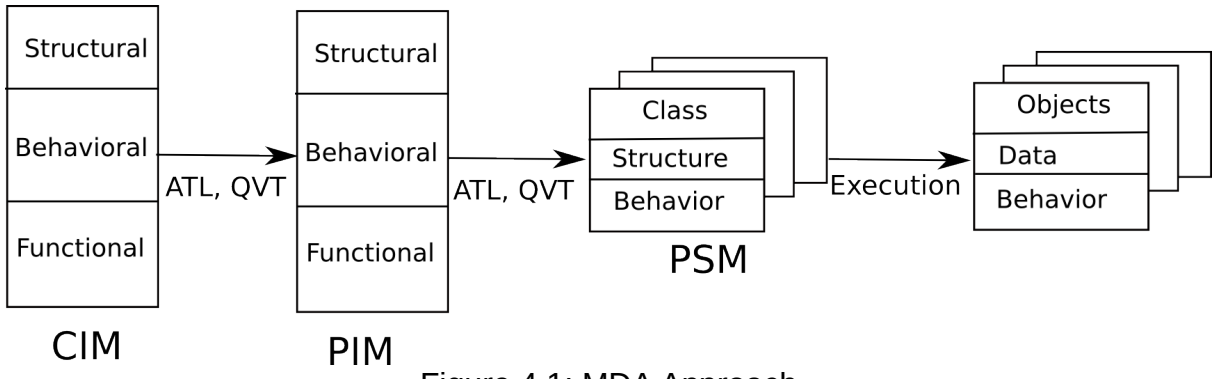

Figure 4.1: MDA Approach

UML structural diagrams are used for describing the structure of the business and the related ICT system. UML activity diagrams and BPMN models are used to represent the behavioral aspect of the real system. However, BPMN is preferred due its rich semantics beyond the ICT world. For the description of the ICT system, UML behavior diagrams are used. UML use case diagrams are frequently deployed to represent functional aspects.

The full MDA approach shows difficulties to complete the automatic transformation between models, especially because automatic transformation is highly dependent of the source and target models. For example, a change in the CIM model requires a change in the subsequent transformations and in the definition of the target models. This feature makes the MDA approach less flexible. Furthermore, the languages used for the system modelling are semi-formal which entails weak semantics and lack of reasoning capabilities. Accordingly, the logic deductions that the system is capable to perform are reduced and most of the logic is hard-coded; affecting flexibility,

adaptability and reuse.

4.1.2 Semantic Web Approach

The semantic web approach is based on the technologies stack presented in Figure 4.2. Ontologies have become the key element for the development of intelligent systems in the web. Ontology-based systems are often combined with the definition of rules in order to achieve a formal description of the system and its service-functional requirements [6], [171], [172].

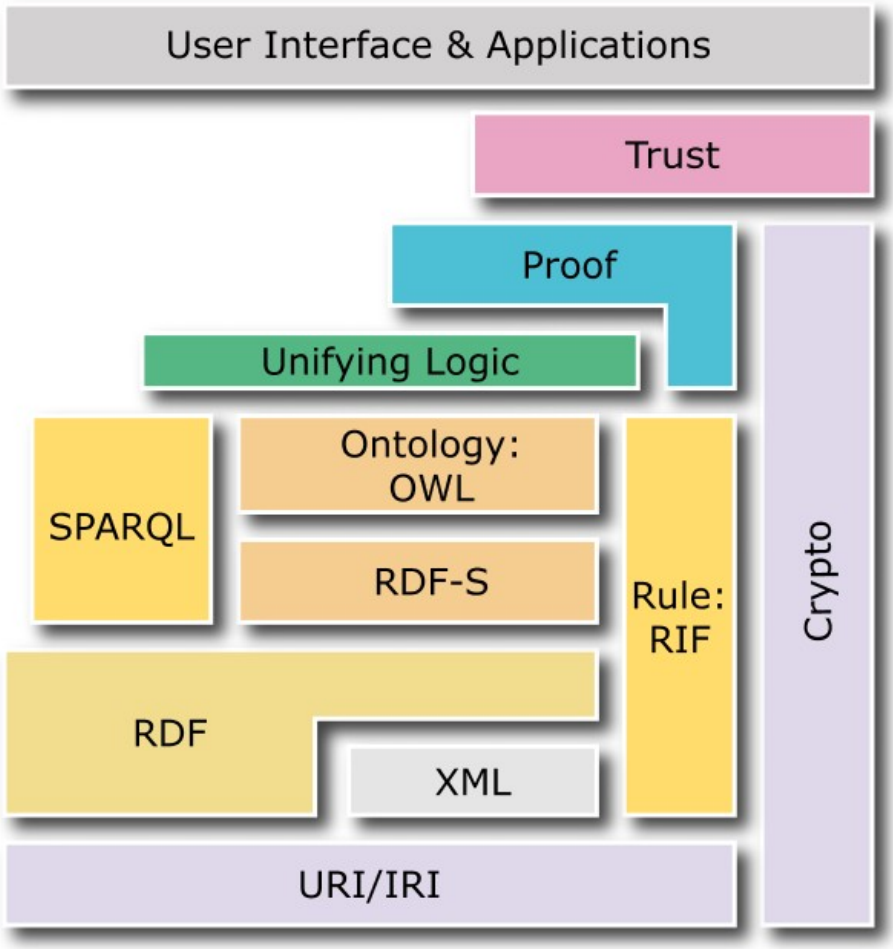

Figure 4.2: Semantic Web Stack, after [200]

All the logic of these applications is managed by queries, ontologies and rules. This approach has strong logic formalization, and the developed systems are able to perform intelligent deductions. However, this approach shows difficulties in representing behavioral aspects [179]. There are many alternatives to RIF/SWRL for

defining rules as presented in Section 1.5.5, and the SPIN language is a standardized alternative working with SPARQL, OWL and RDFS.

4.1.3 Hybrid Approaches

There are many ways to combine MDA and the Semantic Web approach [179]. A strong trend is to combine BPMN and ontologies to overcome the aforementioned weaknesses of the representation languages, e.g. [138]–[142], [144], [148], [149], [181]–[183]. The integration of ontologies and business process modeling is often called semantic business process. In this field, several important works have been provided, e.g., Process Specification Language (PSL) [136], semantic case management [137], or semantic computer-interpretable guidelines [138]. BPMN-based solutions are wider accepted due to their ability of representing the process with graphical diagrams and their standardization level.

The aforementioned solutions remain behind our approach, as they start from the ICT process, thereby ignoring the real world system architecture. Therefore, existing real world domain ontologies haven't been mapped according to the architectural systems' requirements. Even more, domain ontologies have been partially inconsistently and from scratch developed, ignoring existing approved domain and top-level ontologies.

4.2 Description of the Development Approach

The development process proposed in this Thesis combines the BPMN and ontologies framed into the GCM principles to transform the initial models into executable models. For each GCM viewpoint, the models are adapted according to some inputs required for the development process (Figure 4.3).

In the approach, the Business View corresponds with the architectural description presented in Chapter 2 and 3. This description includes formal models using OWL and SPIN languages, and semi-formal models of the behavior using BPMN languages. In the next sections, details of the other views are presented.

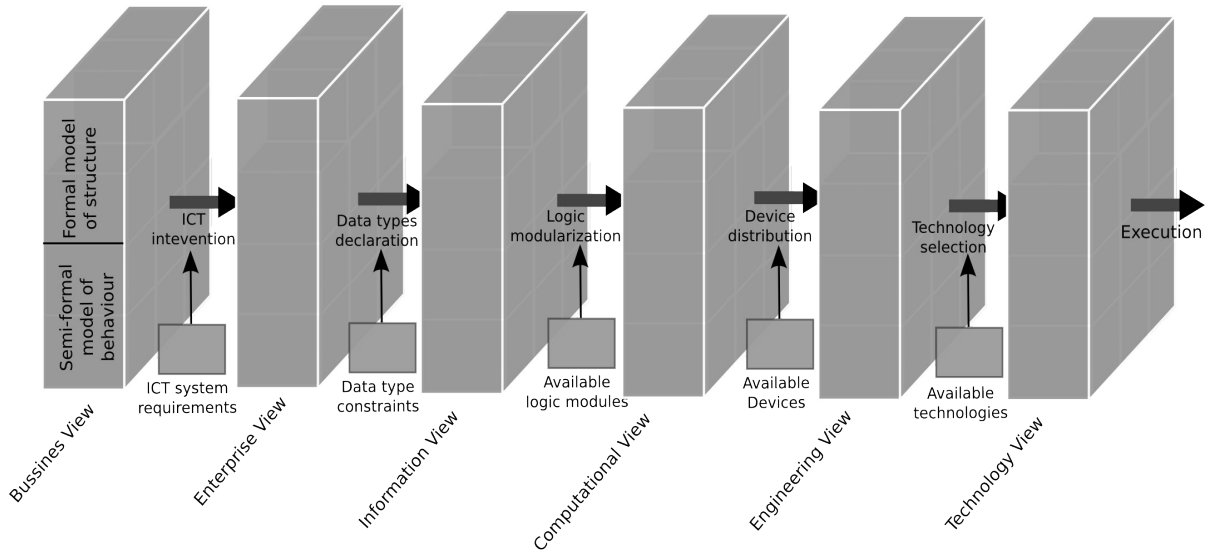

Figure 4.3: Proposed approach

4.2.1 Enterprise View

The Enterprise View defines the roles, activities and policies statements of the specified system [184]. The actors' roles into the system can be classified with the following classes: health organization staff, self-care actor, organizational administrator, and resource chief. The health organization staff was already presented in Section 2.4. Actors with this role perform the medical discipline processes as shown in the use case diagram of Figure 4.4. The specific process and policies for each individual role are defined in the system's rules. Self-care actor class represents the actors that are involved in the self-care task, e.g. the patient or the caregiver.

Organization administrator defines rules governing the organization where the medical processes are performed. Resource chief includes all the actors in charge to perform the resource management, i.e. location chief, human resource chief, software chief, device chief, equipment chief, and pharmacist. As is shown in Figure 4.5, each subclass of “Resource Chief” is in charge of managing the entities represented in the resource domain (see Sections 3.1.1 and 3.1.2).

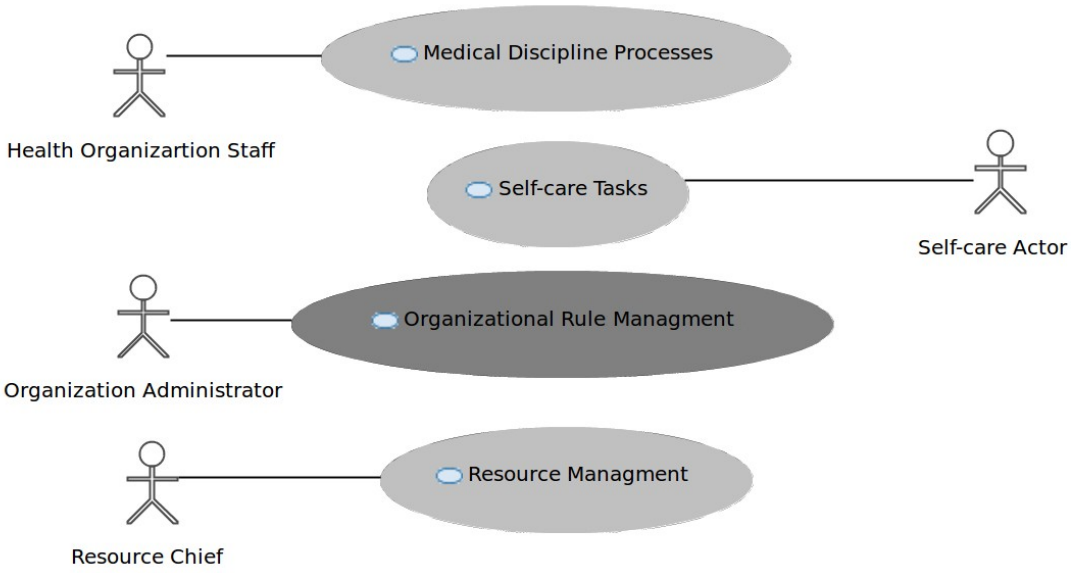

Figure 4.4: Use case diagram of the implemented system

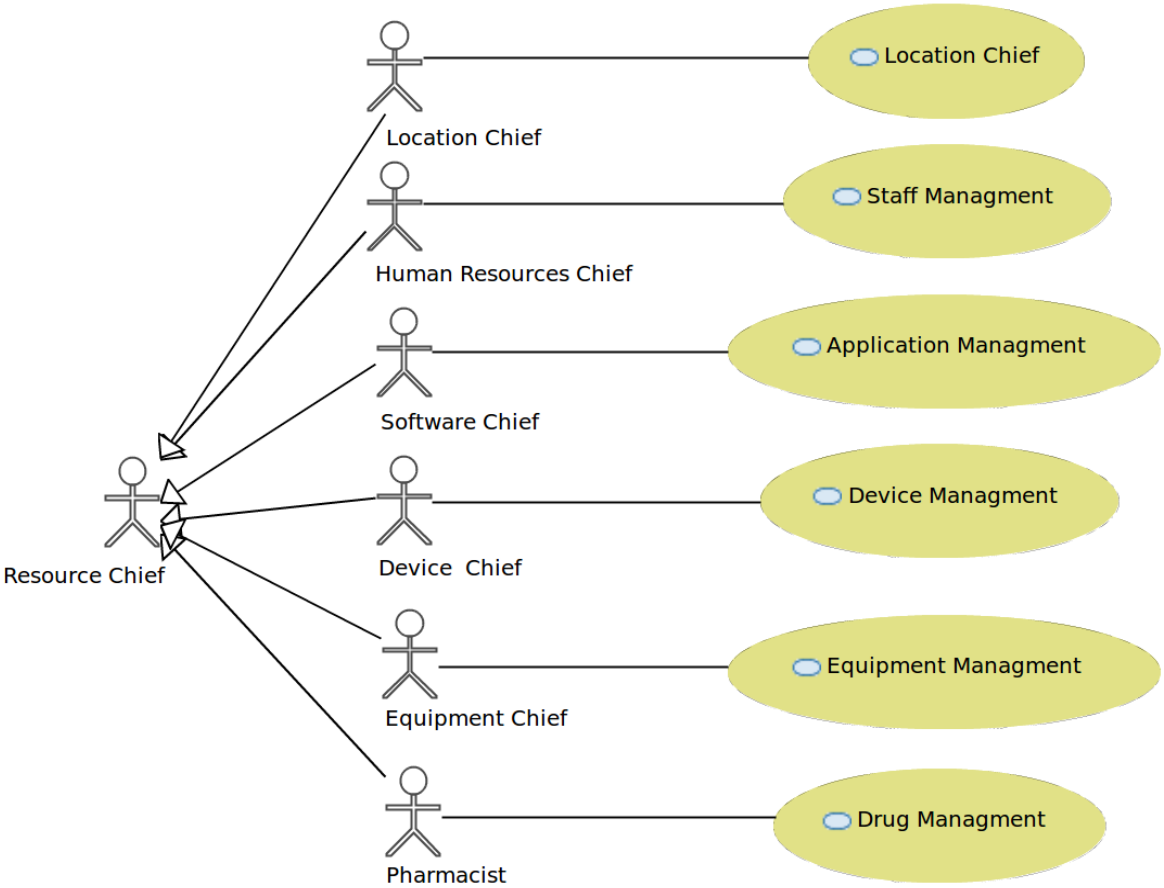

Figure 4.5: Use case diagram for resource chief

4.2.2 Informational View

Information View defines the semantics of information [184], this was already defined in the ontology. The description provided was computation independent. Therefore, the datatypes of the information were ignored. In our approach, the entities representing data are btl2:InformationObject individuals. These entities are the only ones that use datatypes in a computation sense. The next table shows some examples of datatypes constraints.

Class	Datatype Property	Range
dm2co:BloodGlucoseMeasurementResult	dm2co: hasValueIn_mg_dL	xsd:float
	dm2co:hasValueIn_mmol_L	xsd:float
btl2:InformationObject	dm2co:hasValue	xsd:string
btl2:represents some dm2co:Age	dm2co:hasValueIn_years	xsd:positiveInteger
dm2co:BloodPressureMeasurementResult	dm2co:hasValueIn_mmHg	xsd:float

Table 4.1: Datatype constraints

4.2.3 Computational View

Computation View corresponds with the functional decomposition of the system [184]. A first functional decomposition can be performed based on the information cycle given in any collaboration [10], [185]. Figure 4.6 shows the cycle and the functional components of the system (in italics and gray).

In the information cycle, the data is interpreted to get information, based on the information, a decision is made and then the corresponding actions are performed. Finally, the actions are observed in order to obtain new data. All the cycle is based on the knowledge of the executor. Computational systems that support collaboration need to implement that cycle. The DataRepository component is in charge of the data storage. The DataInterpreter is in charge to perform the interpretation of the data, obtaining the information according to the knowledge formalized. The DataMapper component maps the information to the knowledge of other actors involved in the current process. The Planner component is in charge of the decision-making process. This functional component creates an execution plan based on the information. The ExecutionController takes as input the plan, proceeds to assist the actors in the execution of that plan and performs the actions that he is able to do. Finally, the ExecutionListener is in charge of the observation of the process execution in order to get new information relevant in the collaboration.

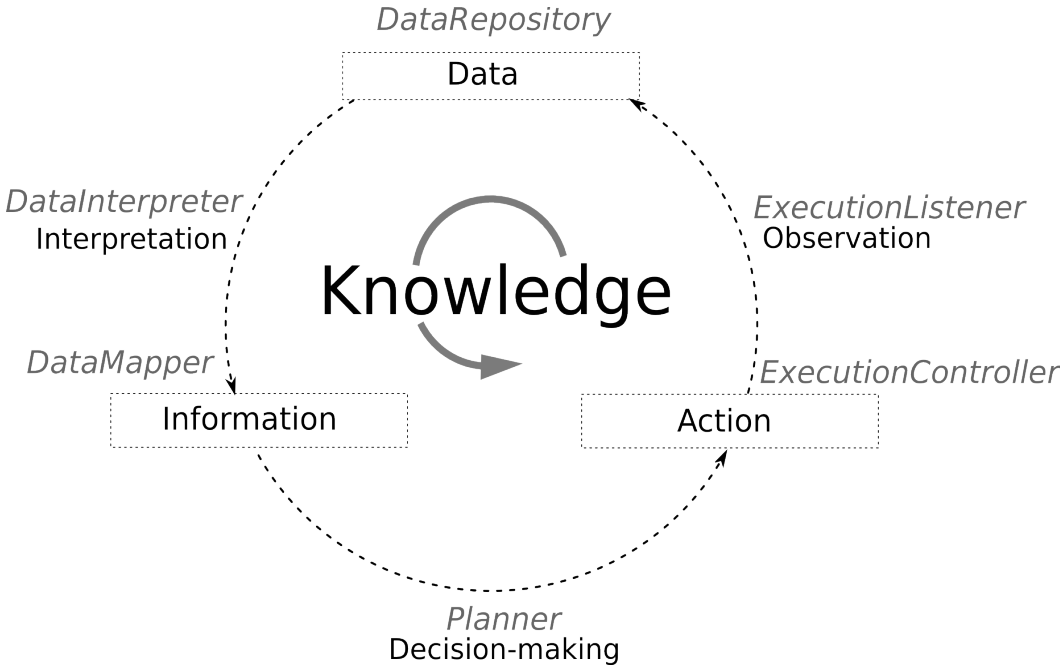

Figure 4.6: Functional components based on the information cycle, after [185] (changed)

Other functional decomposition can be made according with the dependencies between the components and detecting some functional components defined in related works. This decomposition corresponds with the layer representation of Figure 4.7.

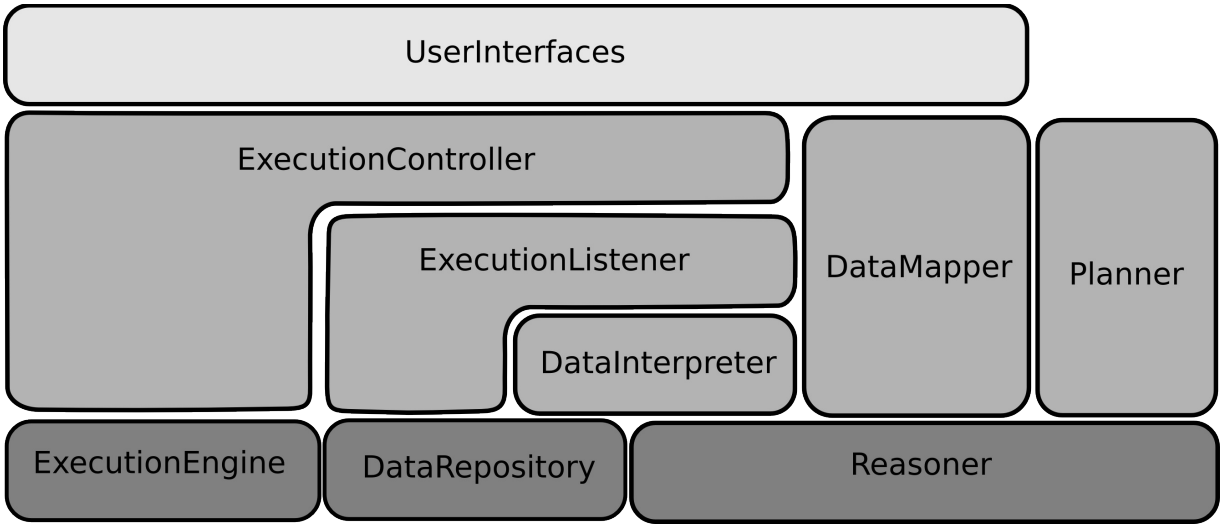

Figure 4.7: Layer diagram of functional components

There are three components of baseline, i.e. ExecutionEngine, DataRepository and Reasoner. Reasoner component is the component in charge of executing the inference rules in order to obtain new axioms. DataMapper, DataInterpreter and Planner work using the reasoner component. The DataRepository component is used only by the ExecutionListener and DataInterpreter components. The ExecutionEngine is the component in charge of interpreting the BPMN models and controlling the flow over the model elements. The ExecutionController is the unique component depending on the ExecutionEngine. However, the ExecutionController component also depends on the ExecutionListener and the DataInterpreter components. Finally, the UserInterfaces component offers usable interfaces to the actors in order to access/add the information and to perform some actions needed in the collaborative process.

4.2.4 Engineering View

Engineering View enables the modelling of the service machine that supports the execution of the computational specification [184]. This model is usually provided to identify the distributed nodes (devices) in the system that supports the computational view. Figure 4.8 shows the distribution for the implemented system.

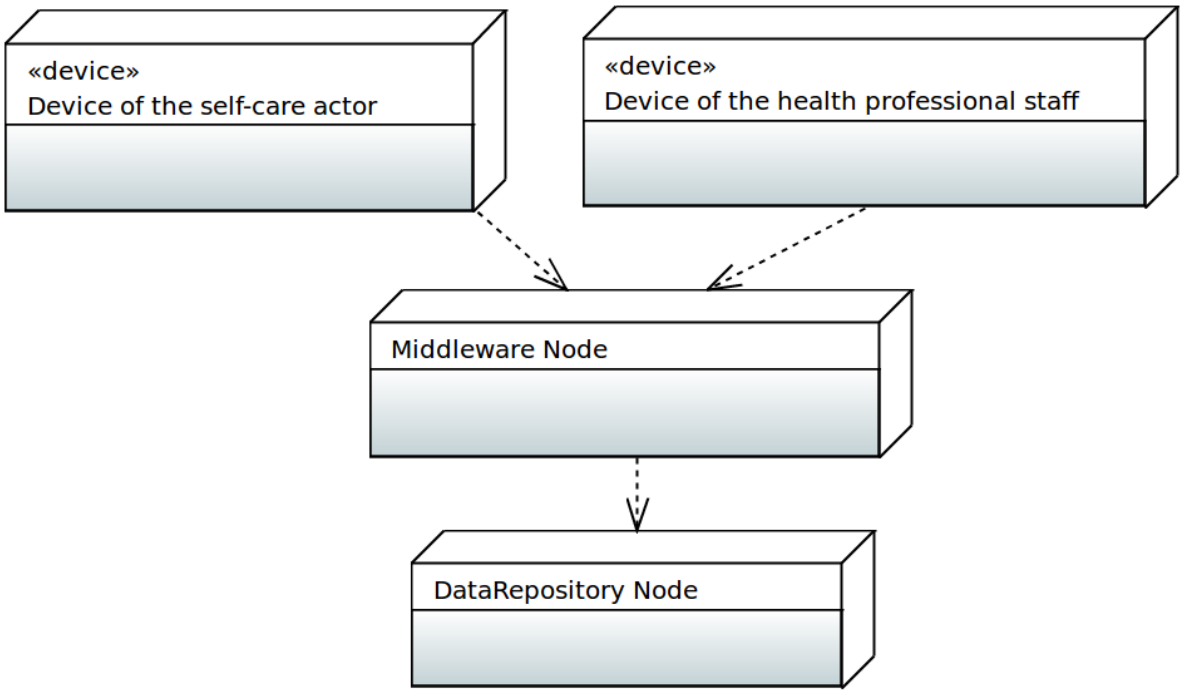

Figure 4.8: Nodes distribution in the system

“Device of the self-care actor” and “Device of the health professional staff” are clients of the UserInterfaces computational component. Each actor, can only use the interfaces to enable its corresponding contributions. The “Middleware Node” includes most of the computational components except the DataRepository that is allocated in its corresponding node.

4.2.5 Technology View

Technology View describes the implementation of the system in terms of a configuration of technology objects representing the hardware and software components of the implementation [184]. In this view, the technologies used to implement the functional components are selected. Figure 4.9 shows the components and its technologies in the implemented system.

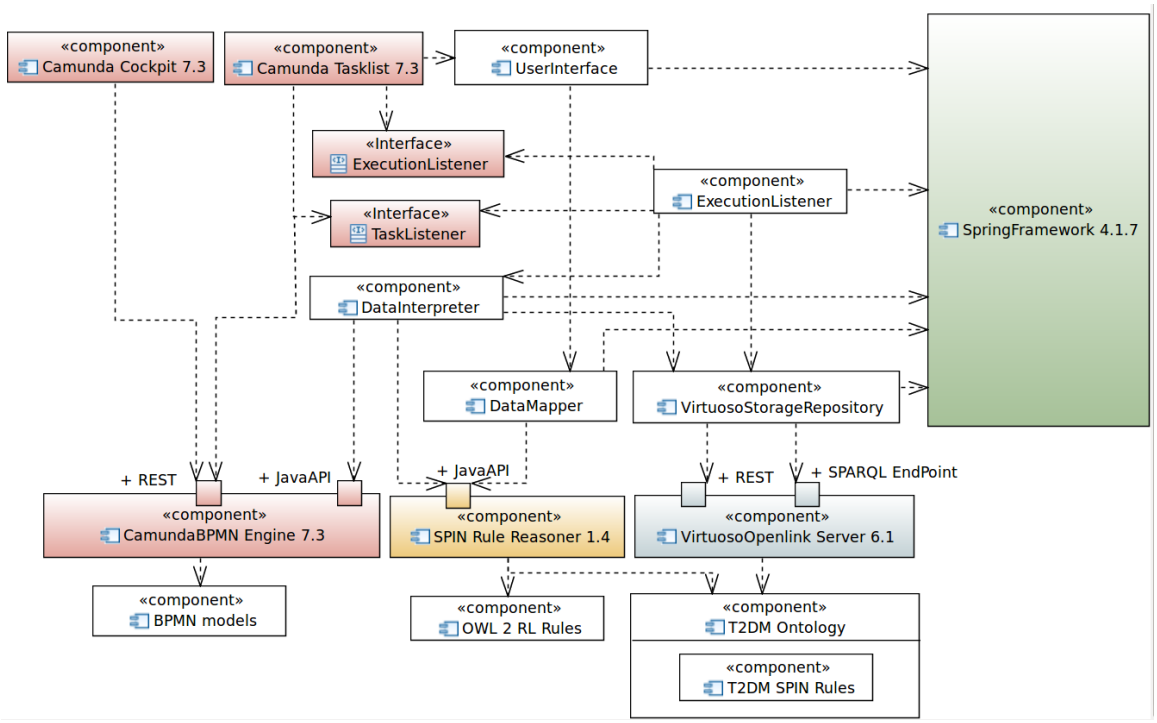

Figure 4.9: Components with their technologies selected

The ExecutionEngine functional component is implemented using the CamundaBPMN Engine in its version 7.3 [91]. The ExecutionController functional component is realized by web applications, also provided by the Camunda platform. It is possible to control the execution of the process by external applications using the

JavaAPI or the REST interface provided by the CamundaBPMN engine [186]. This engine - through extensions to the BPMN 2.0 specification - allows the relationship between the models and Java components. For example, the Camunda Tasklist 7.3 component consumes the ExecutionListener and TaskListener interfaces. The ExecutionListener component offers these interfaces and is therefore able to listen the operation performed over the engine. Reasoner functional component is realized by the SPIN Rule Reasoner in its version 1.4. This software parses the rules defined in SPIN languages and performs the inference process. There are two groups of rules: those defined within the ontology including the policies of the system and the OWL 2 RL [150] rules that define the semantics of this OWL profile. The DataRepository functional component is realized by two components: VirtuosoOpenlink Server (version 6.1) [187] and VirtuosoStorageRepository. The first one is a complete solution for data access and is able to manage RDF-based data repositories. This component offers a REST port to insert data, and a SPARQL EndPoint to query the data. All the components in Figure 4.9 colored in white are the components developed during the progress of the PhD program. Most of the components are developed in Java using the Spring Framework version 4.1.7 [188].

4.3 Testing Scenarios

In this section, four desired features of the system are tested. These features are: adaptability, flexibility, intelligence and interoperability. Adaptability and flexibility are highly related concepts. In the present work adaptability refers to the ability to adjust to new conditions [189]. The adaptation process could imply some configuration of the system, and covers usually long-term changes. The flexibility of a system refers simplicity of modifications [190]. In our context, this means an automatic or assisted re-configuration, and corresponds usually with short-term changes.

4.3.1 Adaptability

Many adaptations can be performed in the system by defining appropriated rules. For example, in the implemented pilot, rules for the adaptation of the alert/emergency messages to the language of the medical doctor are defined. The selection of the preferred language is made at the starting point of the T2DM care process as shown in Figure 4.10.

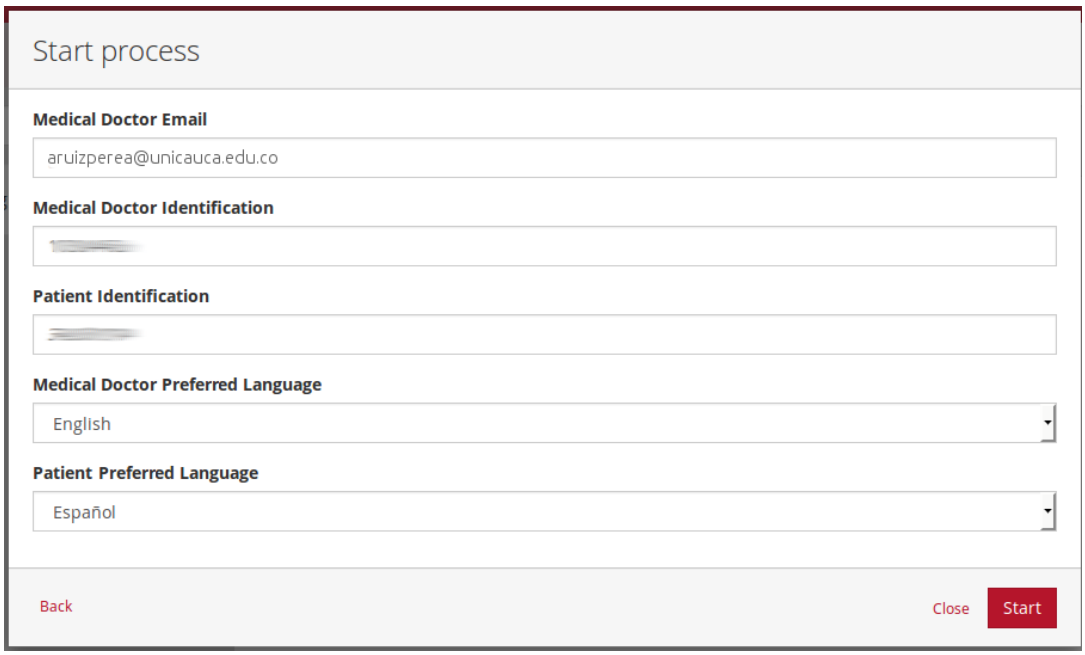

Figure 4.10: T2DM care process configuration

The language transformation is performed by maintaining the labels of the entities in the different languages supported. Currently, automatic translations are not supported. Table 4.2 shows the optional messages delivered to the medical doctor.

Adaptations by rules are limited to the knowledge described in the ontology. Currently, the technology context is not described in the ontology, therefore, adaption to different screen resolutions, devices, etc., is not possible.

Language	Message
English	The patient Gustavo Andrés Uribe Gómez identified with ID ********** has a hyperglycemia medical emergency, fasting blood glucose measurement result value = 320.0 mg/dL
Spanish	El paciente Gustavo Andrés Uribe Gómez identificado con el número ********** presenta un emergencia médica por hiperglucemia, resultado de la medición de la glucosa en sangre en ayunas valor = 320.0 mg/dL

Table 4.2: Language adaptation

4.3.2 Flexibility

The flexibility of the system is also provided by the definition of rules. For example, the rules defined in table 3.7 change the predefined behavior of the system without introducing any configuration at runtime. Other flexibility example is the map of the numeric value of measurement results to a qualitative scale for a better patient understanding. Test for these functionalities are available in the Github repository [191].

4.3.3 Intelligence

The intelligence feature refers to the ability to acquire and apply new knowledge by using inference rules. The system is able to classify findings according to some measurements provided. An example of this functionality is represented in Figure 4.11. In this case, a blood glucose measurement result of 320 mg/dL is sent to the system in the context of a self-observation task.

The ExecutionListener component receives the data and creates the corresponding triples. In the next step, the DataInterpreter component is delegated to follow the process. This module retrieves the data from the DataRepository component and runs the inference process.

The Reasoner provides as result a set of new triples. The Reasoner concludes that the measurement provided corresponds to a Hyperglycemia finding with a MedicalEmergency situation. Based on the policy defined by the medical doctor, preferring email messages for emergency notifications, the system creates a SendMessageByEmailPlan entity.

The DataInterpreter component identifies this plan and starts its execution over the ExectionEngine component. This plan includes ICT tasks and runs at the software system. Therefore, the execution engine executes the JavaBean corresponding to that plan.

The SendMessageByEmail bean uses the Gmail rest service in order to send the corresponding message. The emergency message sent is presented in Table 4.2. Detection of other findings and an example of assisted drug prescription are available in the Github repository [191].

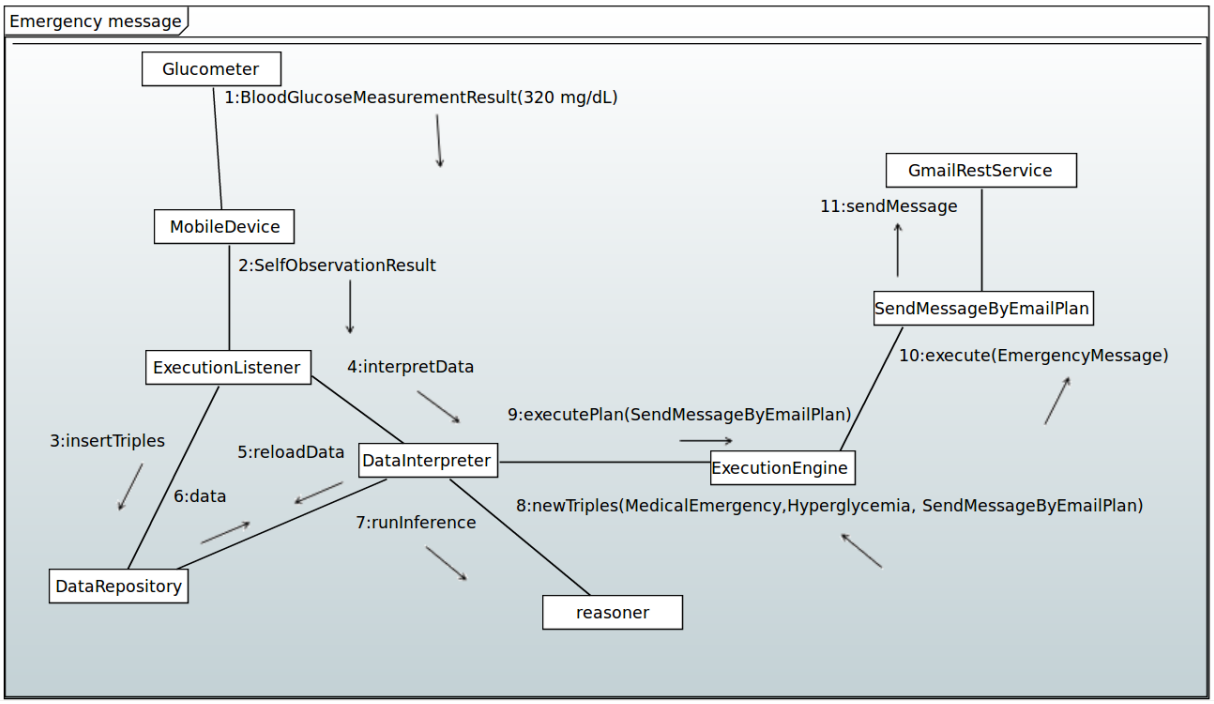

Figure 4.11: Sending an emergency message

4.3.4 Interoperability

Interoperability is the primary outcome of the proposed solution. As mentioned in Chapter 1, interoperability in a practical sense can be defined as the successful collaboration between actors to achieve a common business goal [10]. The business goal in our case is to keep the blood glucose levels as normal as possible. This is only possible if the actors perform the correct actions assisted by the software system. In order to evaluate the interoperability feature of the system the observations, recommendations and prescription of a medical expert are used as gold standard, and then compared with the observations, recommendations and prescription provided by the implemented software. The aforementioned experimental evaluation is presented in Chapter 5.

Furthermore, it is important to highlight that the software implemented support the cross-domain interoperability connecting entities from the medical, policy and resource domains. The domains are interconnected through the rules defined, for example, the table 3.6 corresponds with policies governing the medical behaviour. A special entity is the *HumanOrganism* because is mapped in the resource domain as Person, therefore these two entities can be used indistinctly in the definition of rules.

Then, most of the rules defined in chapter 3 includes persons (resource domain), medical procedures (medical domain) and the rule itself (policy domain).

4.4 Discussion

Traditional development processes like Unified Process (UP) [192] start from user requirements, identify use cases and implements the solution based on those use cases. In this way, the development team is in charge of modelling the system having in mind the types of information generated and shared during the business process. The generated models are semi-formal description of the system and are not intended to describe the system's domain in a logic way. Therefore, at least the following problems arise:

- The models are highly dependent on the development team knowledge. Heterogeneous models from heterogeneous development teams are obtained, without a clear way of harmonization.
- The models ignore essential parts of the business domain because domain experts usually are not part of the team.
- The models cannot guarantee correct inferences using logic rules.
- Most parts of the models are specific for the correspondent business process, limiting the re-usability of components and reducing the chance of interoperability.
- There is no a clear separation between the business domain description and the description of the information objects. That makes interoperability between information models difficult.

As mentioned in Section 4.1, MDA and the semantic web approach solve partially some of these problems. But a complete solution does not yet exist. The presented approach solves the aforementioned problems as follows:

- It uses top-domain and domain ontologies in order to avoid heterogeneous descriptions and allows the harmonization with related models. The ontologies are models verified by domain experts. Therefore they support the correctness of the description.
- It uses formal languages in order to enable reasoning over the models.
- It follows an architectural approach, which offers a generic description,

enabling high re-usability of components and increasing the chance of interoperability.

- The viewpoints separation allows a clear distinction between business and information aspects. This is essential to provide smooth information model interoperability.

The proposed implementation process generates software solutions demanding high processing capabilities. Therefore, a large-scale evaluation is needed. Such evaluation is out of scope of the present work and is part of the proposed future work.

4.5 Conclusions

After studying different alternatives of implementing software intensive systems according to the GCM principles, it was found that a hybrid method combining the MDA principles, the Semantic Web and the Business Process description is more appropriated. This method solves some problems present in traditional development processes and helps to build high quality systems. The proposed method was used to build a system that implements the models proposed in Chapters 2 and 3. The implemented system satisfies the GCM principles and supports the collaboration between actors involved in a glycemic control use case. The features of the system were tested demonstrating adaptability, flexibility, intelligence and interoperability. The evaluation of the proposed method in large-scale application is proposed as future work.

Chapter 5

Evaluation of Interoperability

This chapter shows an experimental evaluation of the interoperability supported by the developed system in the context of a pharmacological glycemic control use case. Next section describes the methodology applied.

5.1 Methods

As was mentioned in Chapter 1, interoperability is a generic concept defined as the relation between/among objects, concretely, a mutual capability necessary to ensure successful and efficient interoperation, supporting cooperation [9] or the successful collaboration between actors to achieve a certain business goal [10]. In the context of the presented work the collaboration/cooperation is supported by the developed software solution. The software provides interoperability at least in the following three ways:

- Controls the execution of the healthcare process according to policies and national medical guidelines and organizational protocols.
- Supports the actors in the decision making process.
- Maps the information considering the heterogeneous qualities of the actors.

The scope of the presented evaluation is limited to the support of the actors in decision making process.

According to DESMET [193], three empirical methods for the evaluation of software are identified: formal experiments, case studies and surveys. The quantitative formal experiment was selected using the criteria in the method selection table provided by

the DESMET methodology, which includes the evaluation context, the nature of the research object, the impact, maturity and learning curve of the service and the researchers capability undertaking the evaluation. The experiment design is described in the following section, following the recommendations of the method for software engineering planning described by Wohlin et al.[194].

5.2 Experimental Design

The objective of the experiment is to evaluate the interoperability of the proposed system by analyzing the effectiveness of the recommendations offered by the system to the users (actors) in order to support their decision making process. Figure 5.1 outlines the experiment.

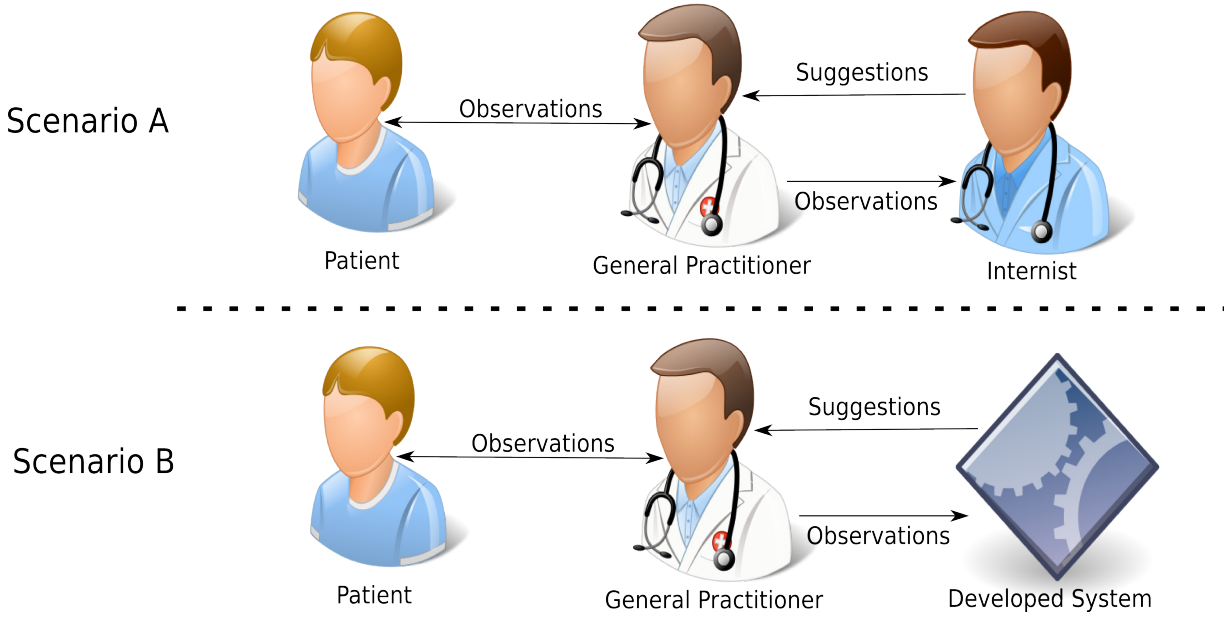

Figure 5.1: Description of the experiment design

The experiment compares the outcome of two different scenarios. Scenario A is the collaboration between a general practitioner and an internist. This is a common scenario in the Colombian context where the endocrinologist (medical specialist in charge of caring diabetes patients) is replaced by a physician specialized in internal medicine (internist) due to the lack of endocrinologists. In this scenario, the general practitioner performs general observations and the internist offers suggestions to the general practitioner in order to take the appropriate decisions in the caring process. In scenario B, the internist is replaced by our developed system suggesting the

appropriate actions. The effectiveness of the scenario B is evaluated using the outcome of the scenario A as gold standard. Therefore, the effectiveness is quantified using the F-measure metric [195], defined as:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \quad (5.1)$$

This metric requires the calculation of the precision and recall [196], which are defined as:

$$precision = \frac{|I|}{|P|} \quad (5.2a)$$

$$recall = \frac{|I|}{|R|} \quad (5.2b)$$

Where I is the set of correct or relevant suggestions provided by the designed system, P is the set of all the suggestions provided by the designed system and R is the set of suggestions provided by the internist (gold standard).

The experiment's elements are described in the following subsections:

5.2.1 Hypothesis

The efficiency of the system's recommendation, measured through the F-measure, is higher than 0.71 using as gold standard the suggestions provided by an internist.

The threshold of 0.71 corresponds with the F-measure average of the algorithms C4.5 and CART evaluated for the diagnosis of diabetes [197].

5.2.2 Experimental Subjects

The system of reference includes a medical internist working in a private health care institution of Popayán, Colombia. This internist is also professor at the University of Cauca. The internist provided 20 anonymized medical records including its observations and decisions made for these patients. The decisions made by the internist are considered equivalent to the suggestions given by him to a general practitioner.

5.2.3 Experimental Objects

The experimental objects are the observation results, findings, diagnosis,

prescriptions and recommendations included in the 20 medical records and the recommendations resulting from the developed system after introducing the observation results of the 20 medical records (Appendix C). The medical records are anonymized, but correspond to real patients.

5.2.4 Treatment and Control Treatment

The control treatment corresponds to the scenario A in Figure 5.1. In this scenario the patients are attended by an internist and a general practitioner. The results of this collaboration are the medical records of the patients. These medical records contain medical findings, diagnosis, prescriptions and recommendations provided by the internist based on the input observations and findings provided by the general practitioner. The treatment corresponds to the scenario B, which uses the observations provided by the general practitioner as input to the developed system. The outcomes of this scenario are the medical diagnoses suggestions provided by the developed system.

5.3 Results

For the scenario B the medical observations were manually introduced into the system using the user interfaces available (in English language), e.g. as demonstrated in Figure 5.2.

After entering all the 20 medical records (Appendix C) the system provides as outcome, some diagnosis suggestions, e.g. diagnosis suggestion as shown in the Figure 5.3.

Following, the suggestions of the system were compared with those provided by the internist. An example is shown in table 5.1. The underlined diagnosis are not asserted by the system and the bold diagnosis corresponds to irrelevant diagnosis.

The not asserted diagnosis are mainly due to the missing inference rules for those diagnoses, for example, the cases of diabetic complications that are not in the scope of the glycemic control pilot. However, those diagnoses were included in the calculation of the F-measure.

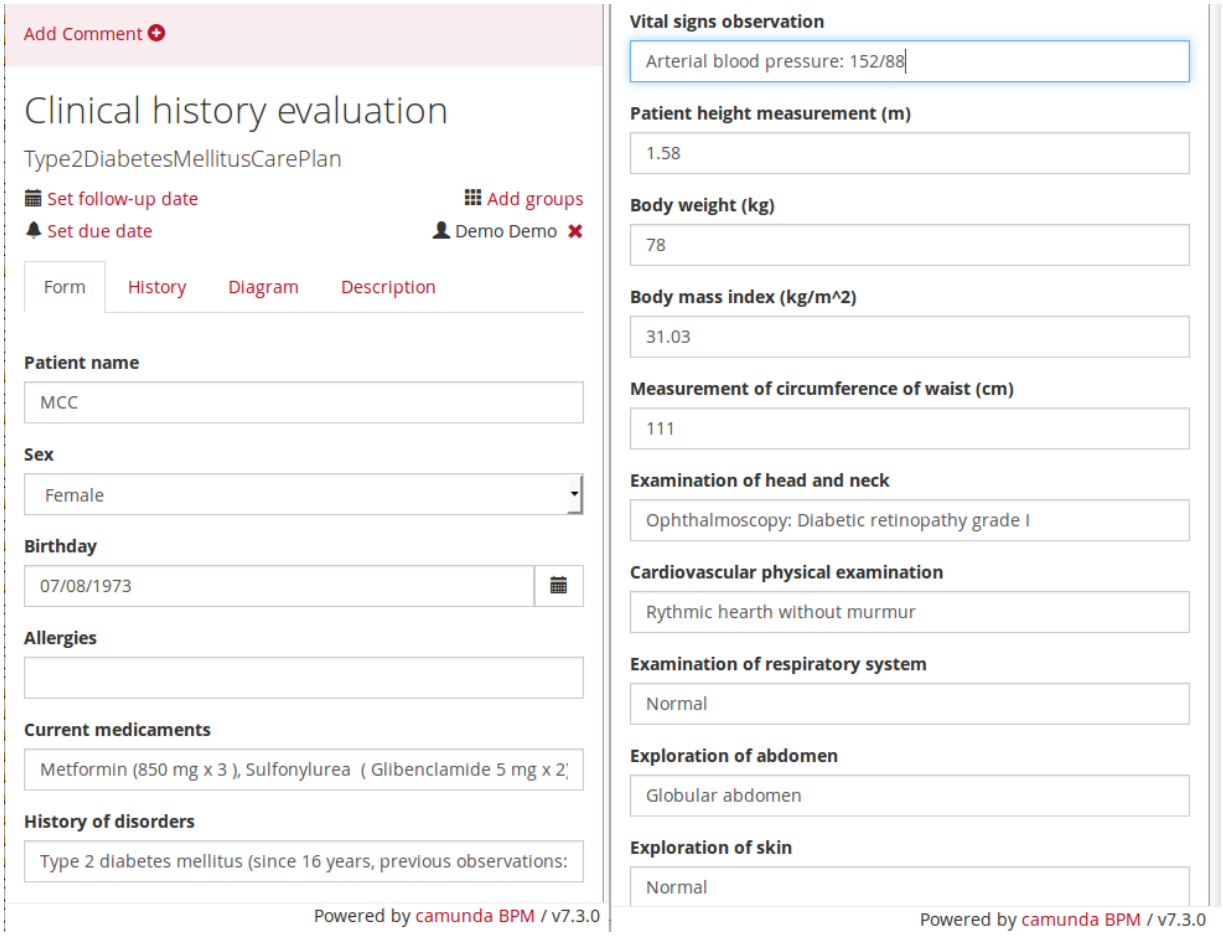

Figure 5.2: Screenshots introducing the medical records

Some diagnoses however, correspond to not-diabetic complications therefore are not asserted and not included in the calculation of the F-measure (e.g. Chondromalacia of patella, Mild malnutrition). Other diagnoses have not been asserted due to the difficulty to infer them using rules (e.g. No chronic complications, uncomplicated diverticular disease colon, probable primary hypothyroidism).

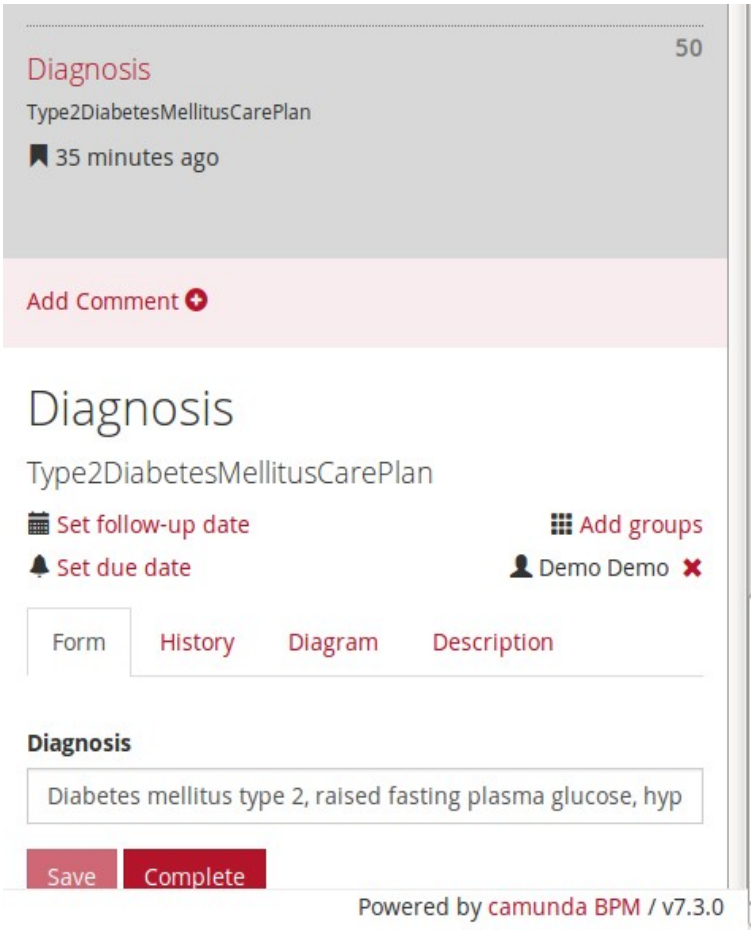

Figure 5.3: Diagnosis suggestions provided by the system

Medical doctor diagnosis	Developed system diagnosis
Type 2 diabetes mellitus	Type 2 diabetes mellitus
Peripheral diabetic neuropathy	Peripheral diabetic neuropathy
Hypertension stage 2	Hypertension stage 2
Overweight	Overweight
Scleral and hypertensive cardiopathy	Metabolic syndrome
Congestive heart failure stage II – C	Hypertriglyceridemia
Coronary artery disease	Raised fasting plasma glucose
Hypertriglyceridemia	***Surasiatic central obesity***
Metabolic syndrome	***Decreased ankle reflex***
Raised fasting plasma glucose	***Medical alert – hyperglycemia***
	Hypoesthesia
	Medical alert – hypertension

Table 5.1: Comparison of medical doctor and developed system diagnosis

The irrelevant diagnoses generated by the system correspond to real states of the patient, however, those diagnoses were considered medically irrelevant the context of the Colombian health systems. One reason is that those diagnoses, generally, are not included in the ICD10, which is used to classify the relevant medical diagnosis in Colombia.

Based on the aforementioned comparison method, the F-measure was calculated using the formula presented in 5.1. The F-measure was calculated for each medical record and for the total of suggestions provided for the 20 medical records. The results of these operations are summarized in Table 5.2 and Figure 5.4.

		Precision	Recall	F-Measure
N	Valid	20	20	20
	Missing	0	0	0
Mean		0.72	0.82	0.74
Std. Deviation		0.15	0.19	0.11
Range		0.50	0.60	0.42
Minimum		0.50	0.40	0.57
Maximum		1.00	1.00	1.00

Table 5.2: Descriptive statistics of the experiment

The total F-measure obtained was 0.74, with a minimum value of 0.57 and a maximum value of 1. This value in the sample confirms our hypothesis obtaining a F-measure over 0.71, and a minimum of 0.57. The F-measure has a standard deviation of 0.11. Therefore, it is possible to conclude that the behavior of the system is stable independently of the differences between the patients. The precision is 0,72 and Recall is 0,82, therefore the system is precise enough compared to similar systems (close to 0,7 threshold), but it is in favor of suggesting mainly relevant results. The minimum precision is 0,5 or 50% (Frequency =1) and the maximum precision was 1 or 100% (Frequency =3). The minimum was 0,4 or 40% (Frequency =1), while the maximum precision was 1 or 100 (Frequency =8). This demonstrates that 40% of the data presents a 100% recall.

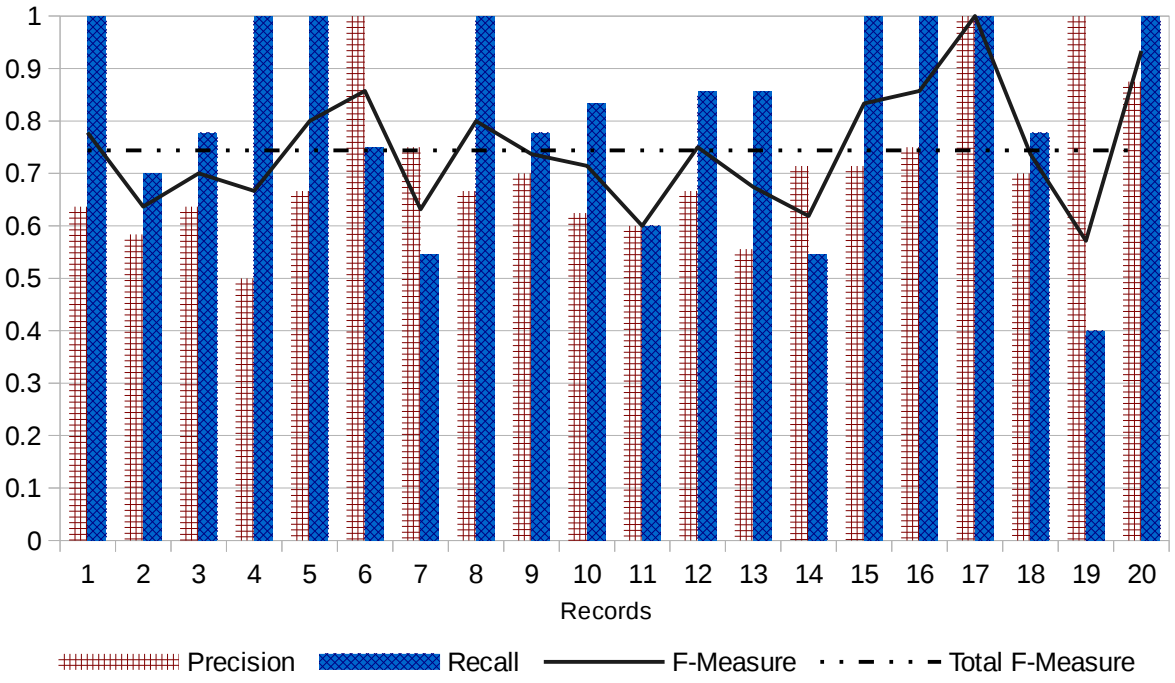

Figure 5.4: F-measure results

The significance of the results was also evaluated with a one-sample T-Test using IBM SPSS Statistics software. Table 5.3 and 5.4 presents the results of the significance test.

First a normality test is performed to the Precision, Recall and F-Measure variables. Only the F-Measure is normal ($p>0.05$). However, the one-sample T-Test is applied to all three variables. The results of the normality test are presented in Table 5.3.

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
F-Measure	0.082	20	0.200	0.969	20	0.739
Precision	0.210	20	0.021	0.875	20	0.014
Recall	0.231	20	0.006	0.863	20	0.009

a. Lilliefors Significance Correction

Table 5.3: Tests of Normality

Table 5.4 presents the results of the one-sample T-test. The one sample T-Test has a result that only Recall is significantly higher than the threshold value (0,71) with a p value of 0.02.

	Test Value = 0.71					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
F-Measure	1.36	19	0.19	0.03	-0.02	0.09
Precision	0.22	19	0.83	0.01	-0.06	0.08
Recall	2.66	19	0.02	0.11	0.02	0.20

Table 5.4: One-Sample Test

In order to improve the precision of the system is possible to add a new rule asserting only diagnoses included in the ICD-10. Applying this rule the results are shown in Table 5.5 and Figure 5.5. The new mean F-measure obtained was 0.88, with a minimum value of 0,57 and a maximum value of 1. The F-measure has a standard deviation of 0.12. The mean precision is 1 and Recall is 0,82, therefore the precision of the system was increased to 100%, and the recall is stable.

	Precision	Recall	F-Measure
N			
Mean	1.00	0.82	0.89
Std. Deviation	0.00	0.19	0.13
Minimum	1.00	0.40	0.57
Maximum	1.00	1.00	1.00

Table 5.5: Descriptive statistics with improved precision

The significance of the results with the new rule was also evaluated with a One-Sample T-Test using the IBM SPSS Statistics software. Table 5.6 and 5.7 presents the results of the significance test. First a normality test is performed to the Precision, Recall and F-Measure variables. The normality of Precision is not calculated because

it is a variable with constant values. No single variable is normal but, the one-sample T-Test can be applied to all three variables. The results of the normality tests are presented in Table 5.6.

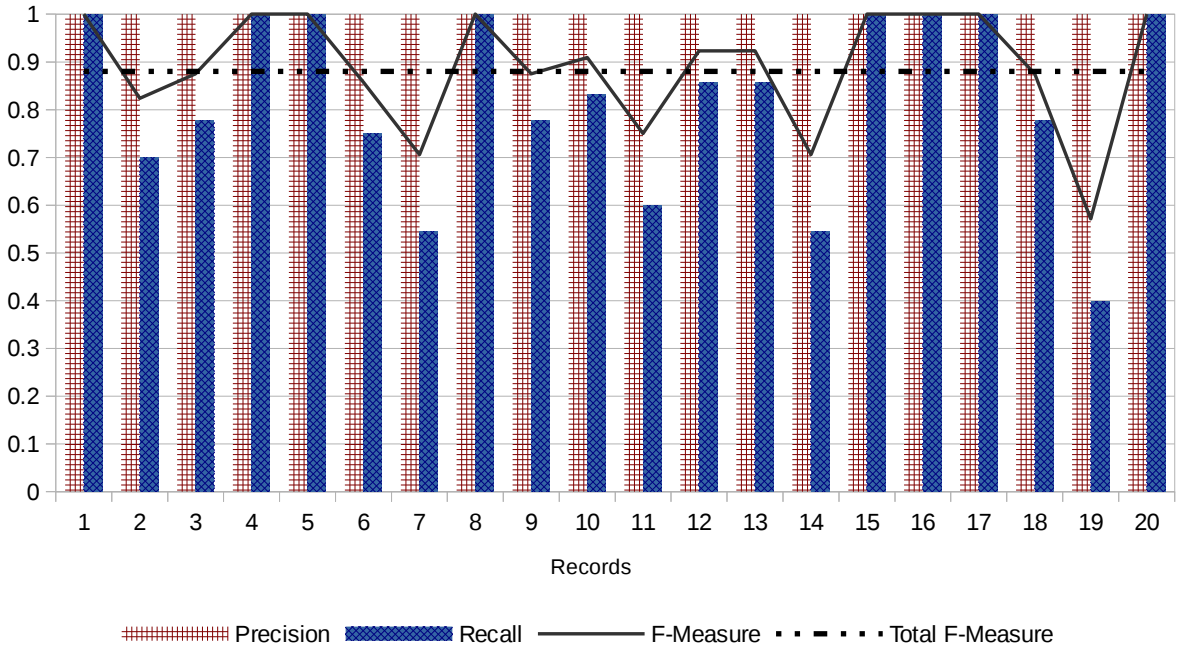

Figure 5.5: F-measure results with improved precision

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Recall	0.231	20	0.006	0.863	20	0.009
F-Measure	0.212	20	0.019	0.840	20	0.004

a. Lilliefors Significance Correction

Table 5.6: Tests of Normality

Table 5.7 presents the results of the one-sample T- test. The T-test cannot be computed to the precision variable, because the standard deviation 0. The one sample T-Test has a result that applying the new rule, the F-measure and the recall are signifi-

cantly higher than the threshold value (0,71) with a value of p=0,00 and p=0,02 respectively.

	Test Value = 0.71					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference Lower	Upper
F-Measure	6.46	19	0.00	0.18	0.12	0.24
Recall	2.66	19	0.02	0.11	0.02	0.20

Table 5.7: One-Sample Test Improved Precision

5.4 Discussion

The efficiency of the system's recommendation, measured through the F-measure, is significantly higher than 0.7 (mean = 0,88) using as gold standard the suggestions provided by an internist. This is true in the second test scenario when the rule considering as valid only ICD-10 coded diagnosis. Therefore, it can be concluded that the suggestions provided by the system are true assertions about the patient and the quality of the suggestions is unlikely to occur by chance. The precision (mean = 1) and recall (mean =0,82) are also significantly higher than the threshold value in the second test scenario. Therefore, the very high precision means that the system returned substantially more relevant recommendations than irrelevant, and the relatively high recall means that the system returned most of the relevant recommendations.

In order to improve even more the F-measure is possible to add the entities and rules corresponding with the diabetic complications and all the related findings. The definition of these entities and rules is only limited by the logic used (description logics - expressivity $\mathcal{SI}(\mathcal{D})$) and the SPIN language. However, the implementation of that entities and rules is out of scope of the present work.

Two special diagnoses found by the doctor are “No chronic complications” and “Uncomplicated diverticular disease colon” because correspond with the absence of one medical condition. Currently is unknown the mechanism to assert the medically relevant diagnoses about absence conditions, probably a machine learning algorithm

can play a better role in this task.

The study had some limitations as the number of samples used, due to the difficulties to get access to patient data. A larger study with a larger number of data is recommended.

5.5 Conclusions

The results of the experiment demonstrates that the system is useful to support the actors in its decision making process, which is a key factor in order to achieve interoperability and the expected goals. The F-measure is directly proportional to the completeness of the domain's description. Having obtained a mean F-measure value = 0,88 with a precision of 100% and recall of 82,1% demonstrates that the suggestions provided by the system are exact and relevant. It was demonstrated that, the development of a very effective system is feasible, but larger study with a larger number of data is recommended in order to demonstrate the quality of the system.

Chapter 6

Conclusions and Future Work

In this chapter, a summary of the dissertation conclusions and future research works are presented.

6.1 General Conclusion

A health information system was developed using the General Component Model. It demonstrated, in a glycemic control use case, cross-domain interoperability of the medical, policy and resource domains. Interoperability is also supported by policies and guidelines, decision support and knowledge mapping.

This result satisfies the hypothesis asserted in Section 1.3, suggesting that the methods applied enables cross-domain interoperability in diabetes care.

6.2 Other Conclusions

The following are the main conclusions of the dissertation:

- The description of the system using the GCM principles enables comprehensive interoperability, also integrating the computer independent aspects that have been ignored in most alternative solutions.
- The architecture-centric approach considers the compositional nature of the real world system and its functionalities in the sense of a system-theoretical White Box approach, and therefore, guarantees coherence of the system model also under the perspectives of multiple different domains.
- The consideration of the top-domain and standardized ontologies facilitates the

harmonization between the different domains involved in the system and enables correct inferences for running the information cycle inherent to any collaboration.

- The level of generality used in the generic description facilitates the adaptive nature of the system and the components re-usability.
- The methodology and models used in the architecture design facilitate the inter-disciplinary communication and allows the development of intelligent systems taking into account the experts' knowledge and relevant policies.
- The methodology allows considering relevant factors in order to improve the health of the T2DM patient such as clinical guidelines, alert conditions, patient safety, and emergency management.
- The ability to perform inferences facilitates the creation of decision support systems. These types of systems are relevant for providing health services in underserved areas, where often qualified health care personal is not available.
- A method combining principles of the MDA, the Semantic Web and the Business Process description was proposed, to implement the principles of the GCM in a software solution. This method solves some problems present in traditional development processes and helps to build high quality systems.
- The proposed method was used to build a system working according to the models provided. The implemented system supports the collaboration between actors involved in the glycemic control use case.
- The implemented system was tested, demonstrating adaptability, flexibility, intelligence, and interoperability.

6.3 Future Work

The following research or development projects are suggested as future work:

6.3.1 Evaluation of the system

The developed system should be evaluated in a large-scale environment, evaluating its response with a high number of patients and health professionals. Furthermore, the medical and financial impact of the solution needs to be evaluated, for example, in rural areas.

6.3.2 Data models mapping

The mapping between information models standards (e.g. HL7 and OpenEHR) using mapping rules over the ontology is feasible and has been demonstrated [130], [198]. This feature was not included in the present thesis due the Colombian context where very few institutions have adopted international health standards.

6.3.3 Automatic Language Transformations

Currently, the translation is only available in the statements presented in the ontology and not in the individuals. In order to extend the multi-language support an automatic translation, new algorithms need to be implemented. Also here, some work has been provided based on the principles used in this Thesis [199].

6.3.4 Automated Planner Composer and Service Discovery

A desired feature in the Planner functional module is the automatic composition of plans, discovering services according to some business goals. These features require the semantic description of the goals and the services. Methodologies for these descriptions are under research.

6.3.5 Development of a Framework for the Proposed Development Process

The proposed development process combines many technologies. Therefore, several tools need to be used separately. It is desired to have a tool integrating the development environment. The tool can also include additional features like a SPIN rules debugger.

References

[1] World Health Organization, “WHO | Diabetes,” 2011. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs312/en/index.html. [Accessed: 13-Jun-2011].

[2] National Library of Medicine, “PubMed Health A. D. A. M. Medical Encyclopedia: Diabetes [Internet]. Bethesda, MD, USA: NLM,” 27-Jun-2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0002194/. [Accessed: 13-Jan-2014].

[3] B. Blobel, “Knowledge Representation and Management Enabling Intelligent Interoperability–Principles and Standards,” *Studies in Health Technology and Informatics*, no. 186, pp. 3–18, 2013.

[4] B. Blobel and P. Pharow, “A model-driven approach for the German health telematics architectural framework and the related security infrastructure,” *Studies in health technology and informatics*, vol. 116, pp. 391–396, 2005.

[5] S. M. Tessier, “Ontology-based approach to enable feature interoperability between CAD systems,” Georgia Institute of Technology, Georgia, 2011.

[6] N. Chungoora, R. I. Young, G. Gunendran, C. Palmer, Z. Usman, N. A. Anjum, A.-F. Cutting-Decelle, J. A. Harding, and K. Case, “A model-driven ontology approach for manufacturing system interoperability and knowledge sharing,” *Computers in Industry*, vol. 64, no. 4, pp. 392–401, 2013.

[7] S. Sonsilphong and N. Arch-int, “Semantic Interoperability for data integration framework using semantic web services and rule-based inference: A case study in healthcare domain,” *Journal of Convergence Information Technology*, vol. 8, no. 3, 2013.

[8] M. Heywood, G. Paterson, M. Shepherd, and S. S. Abidi, “An Ontology-Based Electronic Medical Record for Chronic Disease Management,” 2011.

[9] S. Munk, “An analysis of basic interoperability related terms, system of interoperability types,” *Academic and Applied Research in Military Science*, vol. 1, pp. 117–132, 2002.

[10] B. Blobel, “Intelligent security and privacy solutions for enabling personalized telepathology,” *Diagnostic pathology*, vol. 6, no. Suppl 1, p. S4, 2011.

[11] B. Blobel and P. Pharow, “Analysis and evaluation of EHR approaches,” *Methods Inf Med*, vol. 48, no. 2, pp. 162–169, 2009.

[12] N. Guarino, “Formal ontology, conceptual analysis and knowledge representation,” *International journal of human-computer studies*, vol. 43, no. 5, pp. 625–640, 1995.
[13] T. Hofweber, “Logic and Ontology,” in *The Stanford Encyclopedia of Philosophy*, Spring 2013., E. N. Zalta, Ed. 2013.
[14] M. Ehrig, *Ontology alignment: bridging the semantic gap*. Springer, 2007.
[15] K. Munn and B. Smith, *Applied ontology: an introduction*, vol. 9. ontos verlag, 2008.
[16] W. Kuśnierczyk, “Nontological engineering,” in *Proceedings of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, 2006, pp. 39–50.
[17] S. Schulz and L. Jansen, “Formal ontologies in biomedical knowledge representation,” *Yearbook of medical informatics*, vol. 8, no. 1, p. 132, 2013.
[18] B. Smith, “Beyond concepts: ontology as reality representation,” in *Proceedings of the third international conference on formal ontology in information systems (FOIS 2004)*, 2004, pp. 73–84.
[19] T. R. Gruber and others, “Toward principles for the design of ontologies used for knowledge sharing,” *International journal of human computer studies*, vol. 43, no. 5, pp. 907–928, 1995.
[20] A. Chakravartty, “Scientific Realism,” in *The Stanford Encyclopedia of Philosophy*, Spring 2014., E. N. Zalta, Ed. 2014.
[21] F. Baader, *The description logic handbook: theory, implementation, and applications*. Cambridge Univ Pr, 2003.
[22] W3C, “OWL Web Ontology Language Overview,” Feb-2004. [Online]. Available: http://www.w3.org/TR/owl-features/. [Accessed: 06-Oct-2011].
[23] B. Blobel, “Ontologies, Knowledge Representation, Artificial Intelligence-Hype or Prerequisites for International pHealth Interoperability?,” *Studies in health technology and informatics*, vol. 165, p. 11, 2011.
[24] H. Stenzhorn, E. Beisswanger, and S. Schulz, “Towards a top-domain ontology for linking biomedical ontologies,” in *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, 2007, p. 1225.
[25] B. Blobel, W. Goossen, and M. Brochhausen, “Clinical modeling—A critical analysis,” *International journal of medical informatics*, vol. 83, no. 1, pp. 57–69, 2014.
[26] S. Schulz, D. Seddig-Raufie, N. Grewe, J. Röhl, D. Schober, M. Boeker, and L. Jansen, “Guideline on Developing Good Ontologies in the Biomedical Domain with Description Logics,” 2012.
[27] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, and others, “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration,” *Nature biotechnology*, vol. 25, no. 11, pp. 1251–1255, 2007.
[28] I. Niles and A. Pease, “Towards a standard upper ontology,” in *Proceedings of the international conference on Formal Ontology in Information Systems-*

Volume 2001, 2001, pp. 2–9.
[29] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, “Sweetening ontologies with DOLCE,” in *Knowledge engineering and knowledge management: Ontologies and the semantic Web*, Springer, 2002, pp. 166–181.
[30] H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek, “General formal ontology (GFO),” *Part I: Basic Principles. Onto-Med Report*, vol. 8, 2006.
[31] G. Maiga, “An evaluation framework for large-scale ontology-based biomedical data integrated systems,” 2009.
[32] Z. C. Khan and C. M. Keet, “Addressing issues in foundational ontology mediation,” 2013.
[33] V. Mascardi, V. Cordì, and P. Rosso, “A Comparison of Upper Ontologies.,” in *WOA*, 2007, pp. 55–64.
[34] Z. C. Khan and C. M. Keet, “The foundational ontology library ROMULUS,” in *Model and Data Engineering*, Springer, 2013, pp. 200–211.
[35] S. Schulz and M. Boeker, “BioTopLite: An Upper Level Ontology for the Life Sciences. Evolution, Design and Application,” presented at the Workshop on Ontologies and Data in Life Sciences, Koblenz, Germany, 2013, pp. 19–20.
[36] J. Morbach, A. Wiesner, and W. Marquardt, “OntoCAPE—A (re) usable ontology for computer-aided process engineering,” *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1546–1556, 2009.
[37] L. V. Bertalanffy, *General System Theory: Foundations, Development, Applications*, Revised edition. George Braziller Inc., 2013.
[38] R. L. Ackoff, “Towards a system of systems concepts,” *Management science*, vol. 17, no. 11, pp. 661–671, 1971.
[39] R. L. Ackoff, *Creating the corporate future: Plan or be planned for*. Wiley New York, 1981.
[40] B. Blobel, “Architectural approach to eHealth for enabling paradigm changes in health,” *Methods Inf Med*, vol. 49, no. 2, pp. 123–134, 2010.
[41] B. Blobel, “Translational medicine meets new technologies for enabling personalized care” *Studies in health technology and informatics*, vol. 189, p. 8, 2013.
[42] H. Völz, *Information*. Akademie-Verlag, 1982.
[43] T. ISO, “ISO/TC 215 Technical Report,” *International Organization for Standardization, Health Informatics, ISO TS*, vol. 18308, 2003.
[44] I. ANSI, *TR 20514 Health informatics–Electronic health record—Definition, scope and context*. ISO, 2005.
[45] R. D. Cebul, T. E. Love, A. K. Jain, and C. J. Hebert, “Electronic health records and quality of diabetes care,” *New England Journal of Medicine*, vol. 365, no. 9, pp. 825–833, 2011.
[46] P. J. O’Connor, J. M. Sperl-Hillen, W. A. Rush, P. E. Johnson, G. H. Amundson, S. E. Asche, H. L. Ekstrom, and T. P. Gilmer, “Impact of electronic health record clinical decision support on diabetes care: a randomized trial,” *The Annals of Family Medicine*, vol. 9, no. 1, pp. 12–21, 2011.
[47] R. Ran, C. Zhao, X. Xu, and G. Yao, “Improving perfect electronic health records

and integrated health information in china: a case on disease management of diabetes," in *Health Information Science*, Springer, 2013, pp. 232–243.
[48] S. Wang, "Development of the Diabetes Complication Surveillance System (DCSS)," University of Toronto, 2010.
[49] J. Herrin, B. Graca, D. Nicewander, C. Fullerton, P. Aponte, G. Stanek, T. Cowling, A. Collinsworth, N. S. Fleming, and D. J. Ballard, "The effectiveness of implementing an electronic health record on diabetes care and outcomes," *Health Services Research*, vol. 47, no. 4, pp. 1522–1540, 2012.
[50] M. Reed, J. Huang, I. Graetz, R. Brand, J. Hsu, B. Fireman, and M. Jaffe, "Outpatient electronic health records and the clinical care and outcomes of patients with diabetes mellitus," *Annals of internal medicine*, vol. 157, no. 7, pp. 482–489, 2012.
[51] S. Santana, "Diabetes population management with an electronic health record," *Online Journal of Nursing Informatics (OJNI)*, vol. 17, no. 1, 2013.
[52] K. Tchuitcheu and G. Berenger, "Development and evaluation of a conceptual model with an electronic medical record system for diabetes management in Sub-Saharan Africa," PhD. Thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, Göttingen, 2011.
[53] A. Agrawal, J. Bhattacharya, N. Baranwal, S. Bhatla, S. Dube, V. Sardana, D. R. Gaur, D. Balazova, and S. K. Brahmachari, "Integrating Health Care Delivery and Data Collection in Rural India Using a Rapidly Deployable eHealth Center," *PLoS medicine*, vol. 10, no. 6, p. e1001468, 2013.
[54] J. Kupersmith, J. Francis, E. Kerr, S. Krein, L. Pogach, R. M. Kolodner, and J. B. Perlin, "Advancing evidence-based care for diabetes: lessons from the Veterans Health Administration," *Health Affairs*, vol. 26, no. 2, pp. w156–w168, 2007.
[55] L. D. Booker and H. Trabulsi, "Project Control for Healthcare Information Systems Initiatives," in *Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on*, 2009, pp. 143–151.
[56] P. Fahey, "The Obstacles and Enablers to implementing a Patient Held Prescribing Record in Ireland," PhD. Thesis, University of Dublin, Dublin, 2012.
[57] C. Y. Osborn, L. S. Mayberry, S. A. Mulvaney, and R. Hess, "Patient web portals to improve diabetes outcomes: a systematic review," *Current diabetes reports*, vol. 10, no. 6, pp. 422–435, 2010.
[58] C. C. Quinn, M. D. Shardell, M. L. Terrin, E. A. Barr, S. H. Ballew, and A. L. Gruber-Baldini, "Cluster-randomized trial of a mobile phone personalized behavioral intervention for blood glucose control," *Diabetes Care*, vol. 34, no. 9, pp. 1934–1942, 2011.
[59] J. L. Schnipper, T. K. Gandhi, J. S. Wald, R. W. Grant, E. G. Poon, L. A. Volk, A. Businger, D. H. Williams, E. Siteman, and L. Buckel, "Effects of an online personal health record on medication accuracy and safety: a cluster-randomized trial," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 728–734, 2012.
[60] D. J. Wake and S. G. Cunningham, "'Digital Diabetes'-Looking to the Future," *The British Journal of Diabetes & Vascular Disease*, vol. 13, no. 1, pp. 13–20,

2013.
[61] N. Nijland, J. E. van Gemert-Pijnen, S. M. Kelders, B. J. Brandenburg, and E. R. Seydel, “Factors influencing the use of a Web-based application for supporting the self-care of patients with type 2 diabetes: a longitudinal study,” *Journal of medical Internet research*, vol. 13, no. 3, 2011.
[62] N. Segall, J. G. Saville, P. L’Engle, B. Carlson, M. C. Wright, K. Schulman, and J. E. Tcheng, “Usability evaluation of a personal health record,” in *AMIA Annual Symposium Proceedings*, 2011, vol. 2011, p. 1233.
[63] J. S. Wald, R. W. Grant, J. L. Schnipper, T. K. Gandhi, E. G. Poon, A. C. Businger, E. J. Orav, D. H. Williams, L. A. Volk, and B. Middleton, “Survey analysis of patient experience using a practice-linked PHR for type 2 diabetes mellitus,” in *AMIA Annual Symposium Proceedings*, 2009, vol. 2009, p. 678.
[64] S. G. Cunningham, D. J. Wake, A. Waller, A. D. Morris, and J. Walker, “My Diabetes My Way: an electronic personal health record for diabetes,” *The British Journal of Diabetes & Vascular Disease*, vol. 13, no. 3, pp. 143–149, 2013.
[65] T. Chomutare, L. Fernandez-Luque, E. Arsand, and G. Hartvigsen, “Features of mobile diabetes applications: review of the literature and analysis of current applications compared against evidence-based guidelines,” *Journal of medical Internet research*, vol. 13, no. 3, 2011.
[66] C. M. Costa, D. D. Gondim, D. D. Gondim, H. B. Soares, A. G. Ribeiro, I. Silva, E. Winkler, L. Celi, A. M. Guerreiro, and C. R. Leite, “S2DIA: A Diagnostic System for Diabetes mellitus using SANA platform,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 6078–6081.
[67] A. Dohr, J. Engler, F. Bentley, and R. Whalley, “Gluballoon: an unobtrusive and educational way to better understand one’s diabetes.,” in *UbiComp*, 2012, pp. 665–666.
[68] *Specification for Continuity of Care Record (CCR)*. ASTM International.
[69] S. Castano, A. Ferrara, and S. Montanelli, “Ontology-based Interoperability Services for Semantic Collaboration in Open Networked Systems,” *Springer London*, pp. 135–146, 2006.
[70] “Extended Common Logic Interchange Format (ECLIF) reference.” in: D ocument supplied with Highfleet Integrated Ontology Development Environment (IODE), 2010.
[71] Stanford University, “Protégé.” [Online]. Available: http://protege.stanford.edu/. [Accessed: 14-Sep-2015].
[72] N. Archer, U. Fevrier-Thomas, C. Lokker, K. A. McKibbon, and S. E. Straus, “Personal health records: a scoping review,” *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 515–522, 2011.
[73] T. Snyder and A. P. Honey, “Semantic Interoperability System for Medicinal Information (US Patent),” 20130030827, Jan-2013.
[74] G. A. Uribe, “Ontology-based Interoperability Service for EHR Systems -- Semantic Interoperability of Clinical Information between two Legacy EHR Systems in the Diabetes Context,” Master Thesis, University of Cauca,

Popayán, 2013.
[75] E. Rahm, P. Arnold, H.-H. Do, and D. Aumüller, “COMA 3.0,” *COMA 3.0 | Abteilung Datenbanken Leipzig*. [Online]. Available: http://dbs.uni-leipzig.de/Research/coma.html. [Accessed: 05-Feb-2013].
[76] D. M. Lopez and B. G. Blobel, “A development framework for semantically interoperable health information systems,” *International Journal of Medical Informatics*, vol. 78, no. 2, pp. 83–103, 2009.
[77] B. Blobel, “Introduction into advanced eHealth–the Personal Health challenge.,” *Studies in health technology and informatics*, vol. 134, p. 3, 2008.
[78] M. Lankhorst and others, *Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series)*. Springer, 2nd ed. edn.(Sep 2009), 2009.
[79] B. Blobel, M. Brochhausen, C. González, D. M. Lopez, and F. Oemig, “A system-theoretical, architecture-based approach to ontology management.,” *Studies in health technology and informatics*, vol. 180, p. 1087, 2012.
[80] A. Akerman and J. Tyree, “Using ontology to support development of software architectures,” *IBM Systems Journal*, vol. 45, no. 4, pp. 813–825, 2006.
[81] The Open Group, *Service-Oriented Architecture Ontology, Version 2.0*. 2014.
[82] International Organization for Standardization, “ISO HL7 21731 Health informatics -- HL7 version 3 -- Reference information model -- Release 1.” Geneva: ISO, 2006.
[83] OMG, “Unified Modeling Language (UML).” [Online]. Available: http://www.uml.org/. [Accessed: 19-Nov-2014].
[84] OMG, “Business Process Model and Notation.” [Online]. Available: http://www.bpmn.org/. [Accessed: 19-Nov-2014].
[85] B. Blobel, M. Davis, and P. Ruotsalainen, “Policy Management Standards Enabling Trustworthy pHealth” *Studies in health technology and informatics*, vol. 200, pp. 8–21, 2013.
[86] International Organization for Standardization, “ISO/IEC 19510:2013 Information technology - Object Management Group Business Process Model and Notation.” Geneva:ISO, Nov-2013.
[87] BonitaSoft, *BonitaBPM*. 2014.
[88] Alfresco, *Activiti*. 2014.
[89] Alfresco, “Interprocess communiaction - Activity Forum,” *Interprocess communication [Message flow] | Activiti Forums*. [Online]. Available: http://forums.activiti.org/content/interprocess-communication-message-flow. [Accessed: 07-Nov-2014].
[90] Camunda, *Camunda modeler*. 2014.
[91] Camunda, *Camunda BPM Platform*. 2014.
[92] jBoss Community, *jBPM 6.0*. 2014.
[93] Princeton University, “WordNet Search - Rule,” *WordNet Search - 3-1*. [Online]. Available: http://wordnetweb.princeton.edu/perl/webwn?s=rule. [Accessed: 14-Jul-2014].
[94] L. Morgenstern, C. Welty, H. Boley, and G. Hallmark, “RIF Primer (Second

Edition).” 05-Feb-2013.
[95] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.”
[96] H. Knublauch, J. A. Hendler, and K. Idehen, “SPIN-overview and motivation,” *W3C Member Submission*, 2011. [Online]. Available: http://www.w3.org/Submission/spin-overview/. [Accessed: 20-Nov-2014].
[97] M. Kifer and H. Boley, “RIF Overview (Second Edition).” W3C Working Group, Feb-2013.
[98] jBoss Community, “Drools - Business Rules Managment System.” [Online]. Available: http://www.drools.org/. [Accessed: 20-Nov-2014].
[99] Sandia National Laboratories, “Jess, the Rule Engine for the Java Platform”. 2008.
[100] The International Business Machines Corporation, “IBM Operational Decision Manager.” .
[101] W3C, “OWL 2 Web Ontology Language Document Overview,” *OWL 2 Web Ontology Language Document Overview (Second Edition)*, 2012. [Online]. Available: http://www.w3.org/TR/owl2-overview/. [Accessed: 27-May-2014].
[102] R. Cyganiak, D. Wood, and M. Krötzsch, “RDF 1.1 Concepts and Abstract Syntax.” W3C Recommendation, 25-Feb-2014.
[103] W3C, “Implementations - RIF.” [Online]. Available: http://www.w3.org/2005/rules/wiki/Implementations. [Accessed: 20-Nov-2014].
[104] D. R. Sutton and J. Fox, “The syntax and semantics of the PROforma guideline modeling language,” *J Am Med Inform Assoc*, vol. 10, no. 5, pp. 433–443, Oct. 2003.
[105] R. A. Jenders, R. Corman, and B. Dasgupta, “Making the standard more standard: a data and query model for knowledge representation in the Arden syntax,” *AMIA Annu Symp Proc*, pp. 323–330, 2003.
[106] A. Seyfang, S. Miksch, and M. Marcos, “Combining diagnosis and treatment using ASBRU,” *Int J Med Inform*, vol. 68, no. 1–3, pp. 49–57, Dec. 2002.
[107] A. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. A. Greenes, and E. H. Shortliffe, “GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines,” *J Biomed Inform*, vol. 37, no. 3, pp. 147–161, Jun. 2004.
[108] S. W. Tu, J. R. Campbell, J. Glasgow, M. A. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, J. G. Mansfield, M. A. Musen, and R. M. Abarbanel, “The SAGE Guideline Model: achievements and overview,” *J Am Med Inform Assoc*, vol. 14, no. 5, pp. 589–598, 2007.
[109] B. Blobel, “Knowledge representation and management enabling intelligent interoperability-principles and standards.,” *Data Knowl Med Decis Support.*, pp. 3–18, 2013.
[110] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, and others, “Comparing computer-interpretable guideline models: a case-study approach,” *Journal of the American Medical Informatics Association*, vol. 10, no. 1, pp. 52–68, 2003.

[111] G. A. Uribe, D. M. Lopez, and B. Blobel, “Towards automated biomedical ontology harmonization.,” *Studies in health technology and informatics*, vol. 200, pp. 62–68, 2013.
[112] G. A. Uribe, B. Blobel, D. M. López, and S. Schulz, “A generic architecture for an adaptive, interoperable and intelligent type 2 diabetes mellitus care system,” *Stud Health Technol Inform*, vol. 211, pp. 121–131, 2015.
[113] G. A. Uribe, B. Blobel, D. M. López, and A. A. Ruiz, “Specializing architectures for the type 2 diabetes mellitus care use cases with a focus on process management,” *Stud Health Technol Inform*, vol. 211, pp. 132–142, 2015.
[114] B. Blobel, P. Ruotsalainen, C. Gónzales, and D. M. López, “Policy-Driven Management of Personal Health Information for Enhancing Interoperability,” *Studies in Health Technology and Informatics*, vol. 205, pp. 463–467, Mar. 2014.
[115] International Organization for Standardization, *ISO TS 22600 Health Informatics - Privilege Management and Access Control*, Geneva:ISO. Part, 2006.
[116] B. Blobel, R. Nordberg, J. M. Davis, and P. Pharow, “Modelling privilege management and access control,” *International Journal of Medical Informatics*, vol. 75, no. 8, pp. 597–623, 2006.
[117] M. Makins, *Collins compact English dictionary*. Glasgow: Harper Collins Publishers, 1994.
[118] B. Blobel, “Ontology driven health information systems architectures enable pHealth for empowered patients,” *International journal of medical informatics*, vol. 80, no. 2, pp. e17–e25, 2011.
[119] Regenstrief Institute, “Logical Observation Identifiers Names and Codes (LOINC),” 2014. [Online]. Available: http://www.loinc.org. [Accessed: 21-Aug-2014].
[120] World Health Organization, “International classification of diseases (ICD),” 2012.
[121] OBO Foundry, “The Open Biological and Biomedical Ontologies,” 2015. [Online]. Available: http://www.obofoundry.org/. [Accessed: 02-Oct-2015].
[122] IHTSDO, “SNOMED Clinical Terms Overview.” Sep-2008.
[123] G. A. Uribe, D. M. López, and B. Blobel, “Architectural Analysis of Clinical Ontologies for pHealth Interoperability.,” *Studies in health technology and informatics*, vol. 177, pp. 176–182, 2012.
[124] International Labour Organization, “International Standard Classification of Occupations (ISCO),” *ISCO - International Standard Classification of Occupations*, 2007. [Online]. Available: http://www.ilo.org/public/english/bureau/stat/isco/. [Accessed: 06-Apr-2014].
[125] International Organization for Standardization, “Health informatics—Functional and structural roles.” 2008.
[126] B. Blobel, G. A. Uribe, M. Brochhausen, D. M. López, and S. Schulz, “Formalization of Basics and Principles for Interoperable System Architectures,” (in submission).
[127] M. Brochhausen and B. Blobel, “Architectural approach for providing relations in biomedical terminologies and ontologies.,” *Studies in health technology and informatics*, vol. 169, pp. 739–743, 2011.

[128] F. Oemig and B. Blobel, “A formal analysis of HL7 version 2. x.,” *Studies in health technology and informatics*, vol. 169, p. 704, 2011.
[129] F. Oemig and B. Blobel, “An ontology architecture for HL7 V3: Pitfalls and outcomes,” in *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*, 2009, pp. 408–410.
[130] F. Oemig and B. Blobel, “Establishing semantic interoperability between HL7 v2. x and V3: a Communication Standards Ontology (CSO),” *Journal of Health Informatics*, vol. 3, no. 4, 2011.
[131] Healt Level 7 International, “HL7 Version 3 Standard: Security and Privacy Ontology, Release 1.0,” *Ann Arbor: HL7 International*, 2013.
[132] W. Goossen, A. Goossen-Baremans, and M. van der Zel, “Detailed Clinical Models: A Review,” *Healthcare Informatics Research*, vol. 16, no. 4, p. 201, 2010.
[133] M. Vida, L. Stoicu-Tivadar, B. Blobel, and E. Bernad, “Interoperability Evaluation Case Study: An Obstetrics-Gynecology Department and Related Information Systems,” *Stud Health Technol Inform.*, vol. 186, pp. 177–181, 2013.
[134] M. Vida, L. Stoicu-Tivadar, B. Blobel, and E. Bernad, “Modeling the Framework for Obstetrics-Gynecology Department Information System,” *European Journal for Biomedical Informatics*, vol. 8, no. 3, pp. en57–en64, 2012.
[135] B. Blobel and F. Oemig, “The importance of architectures for interoperability.,” *Studies in health technology and informatics*, vol. 211, pp. 18–56, 2014.
[136] M. Gruninger and C. Menzel, “The process specification language (PSL) theory and applications,” *AI magazine*, vol. 24, no. 3, p. 63, 2003.
[137] L. Boaro, “Formal methods for semantic case management,” 2013.
[138] D. Riaño, F. Real, J. A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, and C. Caltagirone, “An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients,” *Journal of biomedical informatics*, vol. 45, no. 3, pp. 429–446, 2012.
[139] C. Ghidini, M. Rospocher, and L. Serafini, “A formalisation of BPMN in description logics,” Technical report TR 2008-06-004, FBK-irst, 2008.
[140] C. Natschläger, “Towards a BPMN 2.0 ontology,” in *Business Process Model and Notation*, Springer, 2011, pp. 1–15.
[141] L. Penicina, “Choosing a BPMN 2.0 Compatible Upper Ontology,” in *eKNOW 2013, The Fifth International Conference on Information, Process, and Knowledge Management*, 2013, pp. 89–96.
[142] W. Yao and A. Kumar, “CONFlexFlow: Integrating flexible clinical pathways into clinical decision support systems using context and rules,” *Decision Support Systems*, vol. 55, no. 2, pp. 499–515, 2013.
[143] M. Cortes-Cornax, I. Ciuciu, S. Dupuy-Chessa, D. Rieu, and others, “Towards the Integration of Ontologies with Service Choreographies,” in *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, 2013, pp. 343–352.
[144] M. Born, F. Dörr, and I. Weber, “User-friendly semantic annotation in business process modeling,” in *Web Information Systems Engineering–WISE 2007 Workshops*, 2007, pp. 260–271.

[145] L. E. Tello, J. A. Carreón, and M. L. Castillo, “Enfoque para la gestión de procesos de negocio semánticos utilizando ontologías,” *Revista Ingenierías USBMed*, vol. 4, no. 1, pp. 56–62, 2013.
[146] B. Wetzstein, Z. Ma, A. Filipowska, M. Kaczmarek, S. Bhiri, S. Losada, J.-M. Lopez-Cob, and L. Cicurel, “Semantic Business Process Management: A Lifecycle Based Requirements Analysis.,” in *SBPM*, 2007.
[147] European Union, “Super Home Page,” *Super Integrated Project - Home*, 2009. [Online]. Available: http://www.ip-super.org/. [Accessed: 25-Jul-2014].
[148] N. Hashemian and S. S. R. Abidi, “Modeling clinical workflows using business process modeling notation,” in *Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on*, 2012, pp. 1–4.
[149] F. Smith and M. Proietti, “Rule-based Behavioral Reasoning on Semantic Business Processes.,” in *ICAART (2)*, 2013, pp. 130–143.
[150] World Wide Web Consortium, “OWL 2 Web Ontology Language Profiles,” 12-Nov-2012. [Online]. Available: http://www.w3.org/TR/owl2-profiles/. [Accessed: 30-Jul-2014].
[151] M. Krötzsch, *OWL 2 Profiles: An introduction to lightweight ontology languages*. Springer, 2012.
[152] N. Drummond, M. Horridge, R. Stevens, C. Wroe, and S. Sampaio, *Pizza ontology*. The University of Manchester, 2007.
[153] B. Smith, “Applied ontology: a new discipline is born,” *Philosophy Today*, vol. 12, no. 29, pp. 5–6, 1998.
[154] R. Rabasa-Lhoret and M. B. Stuart Ross, “Targets for Glycemic Control,” *Canadian Journal of Diabetes*, vol. 37, pp. S31–S34, 2013.
[155] M. D. Robyn Houlden, M. D. David Miller, and M. D. Vincent Woo, “Monitoring Glycemic Control,” *Canadian Journal of Diabetes*, vol. 37, pp. S35–S39, 2013.
[156] A. E. Pinilla, L. Lancheros, and D. F. Viasus, “Guía de atención de la diabetes mellitus tipo 2,” *Guías de promoción de la salud y prevención de enfermedades en la salud pública. Bogotá*, pp. 361–439, 2007.
[157] W. Harper, A. Hanna, V. Woo, K. G. Dawson, J.-F. Yale, L. MacCallum, C. Maureen, S. Scot, and M. Hopkins, “Pharmacologic Management of Type 2 Diabetes,” *Canadian Journal of Diabetes*, vol. 32, pp. S53–S61, 2008.
[158] K. Harno, T. Paavola, C. Carlson, and P. Viikinkoski, “Patient referral by telemedicine: effectiveness and cost analysis of an Intranet system,” *J Telemed Telecare*, vol. 6, no. 6, pp. 320–329, 2000.
[159] C. D. A. C. P. G. E. Committee and others, “Nutrition therapy,” *Can J Diabetes*, vol. 32, no. Suppl 1, pp. S40–S45, 2008.
[160] R. J. Sigal, G. P. Kenny, D. H. Wasserman, and C. Castaneda-Sceppa, “Physical activity/exercise and type 2 diabetes,” *Diabetes care*, vol. 27, no. 10, pp. 2518–2539, 2004.
[161] World Medical Association and et al., “World Medical Association Declaration of Helsinki. Ethical principles for medical research involving human subjects.,” *Bulletin of the World Health Organization*, vol. 79, no. 4, p. 373, 2001.
[162] Colombian Congress, “Ley 1581 de 2012.” 2012.

[163] Colombian Congress, “Ley 23 de 1981.” .
[164] C. Técnico, “Unidad Sectorial de Normalización en Salud,” *Guía Técnica“ Buenas Prácticas” para la seguridad del paciente en la atencion en salud. Bogotá: Ministerio de la Protección Social, USN-Unidad Sectorial de Normalización en Salud*, pp. 77–81, 2010.
[165] R. C. Turner, R. R. Holman, C. A. Cull, I. M. Stratton, D. R. Matthews, V. Frighi, S. E. Manley, A. Neil, K. Mcelroy, D. Wright, and others, “Intensive blood-glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes (UKPDS 33),” *lancet*, vol. 352, no. 9131, pp. 837–853, 1998.
[166] D. Control, C. T. R. Group, and et al., “The relationship of glycemic exposure (HbA1c) to the risk of development and progression of retinopathy in the Diabetes Control and Complications Trial,” *Diabetes*, vol. 44, no. 8, pp. 968–983, 1995.
[167] I. M. Stratton, A. I. Adler, H. A. W. Neil, D. R. Matthews, S. E. Manley, C. A. Cull, D. Hadden, R. C. Turner, and R. R. Holman, “Association of glycaemia with macrovascular and microvascular complications of type 2 diabetes (UKPDS 35): prospective observational study,” *BMJ*, vol. 321, no. 7258, pp. 405–412, 2000.
[168] G. A. Uribe, B. Bernd, D. M. López, and S. Schulz, “Architecture for an Adaptive, Interoperable and Intelligent Type 2 Diabetes Mellitus Care System,” *Studies in Health Technology and Informatics*, (submitted).
[169] J. O. López, *Constitución política de Colombia*. Plaza y Janes Editores Colombia sa, 2004.
[170] C. de la República, “Ley 1122 de 2007: por la cual se hacen algunas modificaciones en el Sistema General de Seguridad Social en Salud y se dictan otras disposiciones,” 2007.
[171] C. dCongreso, “Ley 100 de 1993, diciembre 23, por la cual se crea el sistema de seguridad social integral y se dictan otras disposiciones,” 1993.
[172] Congreso de Colombia, “Ley 1438 del 19 de Enero de 2011: Por medio de la cual se reforma el sistema general de seguridad social en salud y se dictan otras disposiciones.” 2011.
[173] J. Kramer, “Is abstraction the key to computing?,” *Communications of the ACM*, vol. 50, no. 4, pp. 36–42, 2007.
[174]
[175] B. Brahim, E. B. Omar, and G. Taoufiq, “A methodology for CIM modelling and its transformation to PIM,” *Journal of Information Engineering and Applications*, vol. 3, no. 2, pp. 1–21, 2013.
[176] A. Kriouile, N. Addamssiri, T. Gadi, and Y. Balouki, “Getting the static model of PIM from the CIM,” in *Information Science and Technology (CIST), 2014 Third IEEE International Colloquium in*, 2014, pp. 168–173.
[177] N. Silega, T. T. Loureiro, and M. Noguera, “Model-driven and ontology-based framework for semantic description and validation of business processes,” *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 12, no. 2, pp. 292–299, Mar. 2014.

[178] A. Rodríguez, E. Fernández-Medina, J. Trujillo, and M. Piattini, “Secure business process model specification through a UML 2.0 activity diagram profile,” *Decision Support Systems*, vol. 51, no. 3, pp. 446–465, Jun. 2011.
[179] F. S. Parreiras, *Semantic Web and Model-Driven Engineering*. John Wiley & Sons, 2012.
[180] B. Blobel and F. Oemig, “Ontology-driven health information systems architectures,” *this volume*, 2009.
[181] N. Lasierra, A. Alesanco, and J. Garcia, “Designing an Architecture for Monitoring Patients at Home: Ontologies and Web Services for Clinical and Technical Management Integration,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 896–906, May 2014.
[182] L. Subirats, L. Ceccaroni, C. Gómez-Pérez, R. Caballero, R. Lopez-Blazquez, and F. Miralles, “On Semantic, Rule-Based Reasoning in the Management of Functional Rehabilitation Processes,” in *Management Intelligent Systems*, Springer, 2013, pp. 51–58.
[183] A. Daniyal and S. S. R. Abidi, “Semantic Web-based modeling of Clinical Pathways using the UML Activity Diagrams and OWL-S,” in *Knowledge representation for health-care. Data, processes and guidelines*, Springer, 2010, pp. 88–99.
[184] ISO, “Information technology — Open Distributed Processing — Reference model: Overview.” [Online]. Available: http://www.itu.int/rec/T-REC-X.901-199708-I/en. [Accessed: 11-Jun-2014].
[185] B. Blobel, “New Approaches to Privacy and Security,” Cambridge, U.S.A., 25-Oct-2013.
[186] Camunda, “REST API | camunda BPM docs.” [Online]. Available: http://docs.camunda.org/latest/api-references/rest/. [Accessed: 06-Sep-2015].
[187] OpenLink, “OpenLink Virtuoso Home Page.” [Online]. Available: http://virtuoso.openlinksw.com/. [Accessed: 07-Sep-2015].
[188] Pivotal Software, “Spring Framework.” [Online]. Available: http://projects.spring.io/spring-framework/. [Accessed: 07-Sep-2015].
[189] Oxford University Press, “adaptability - definition of adaptability in English from the Oxford dictionary.” [Online]. Available: http://www.oxforddictionaries.com/definition/english/adaptability. [Accessed: 07-Sep-2015].
[190] Oxford University Press, “flexibility - definition of flexibility in English from the Oxford dictionary.” [Online]. Available: http://www.oxforddictionaries.com/definition/english/flexibility. [Accessed: 07-Sep-2015].
[191] G. A. Uribe, “gaurgo/DiabetesCare,” *GitHub*. [Online]. Available: https://github.com/gaurgo/DiabetesCare. [Accessed: 07-Sep-2015].
[192] I. Jacobson, G. Booch, J. Rumbaugh, J. Rumbaugh, and G. Booch, *The unified software development process*, vol. 1. Addison-wesley Reading, 1999.
[193] B. Kitchenham, S. Linkman, and D. Law, “DESMET: a methodology for evaluating software engineering methods and tools,” *Computing & Control*

Engineering Journal, vol. 8, no. 3, pp. 120–126, 1997.
[194] C. Wohlin, *Experimentation in software engineering: an introduction*, vol. 6. Springer Netherlands, 2000.
[195] C. Van Rijsbergen, *Information Retrival*, 2th edition. London; Boston: Butterworth, 1979.
[196] D. Grigori, J. C. Corrales, and M. Bouzeghoub, “Behavioral matchmaking for service retrieval,” in *Web Services, 2006. ICWS’06. International Conference on*, 2006, pp. 145–152.
[197] D. S. Kumar, G. Sathyadevi, and S. Sivanesh, “Decision support system for medical diagnosis using data mining,” *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 147–153, 2011.
[198] F. Oemig and B. Blobel, “An Ontological Approach to Automate Interface Configurations,” *European Journal for Biomedical Informatics*, vol. 8, no. 3, pp. en18–en21, 2012.
[199] F. Oemig and B. Blobel, “Natural Language Processing Supporting Interoperability in Healthcare,” in *Text Mining*, C. Biemann and A. Mehler, Eds. Springer International Publishing, 2014, pp. 137–156.
[200] W3C, “Semantic Web, and Other Technologies to Watch: January 2007 (24).” [Online]. Available: http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24). [Accessed: 10-Sep-2015].