Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

ON CONTEXT-AWARE SOFTWARE SYSTEMS AND INTEROPERABILITY: A DISCUSSION GROUNDED IN DATA

Rebeca Campos Motta

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro
Junho de 2016

ON CONTEXT-AWARE SOFTWARE SYSTEMS AND INTEROPERABILITY:
A DISCUSSION GROUNDED IN DATA

Rebeca Campos Motta

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

____

Prof. Guilherme Horta Travassos, D.Sc.

____

Prof. Toacy Cavalcante de Oliveira, D.Sc

____

Profª. Sheila dos Santos Reinehr, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2016

Motta, Rebeca Campos

On Context-Aware Software Systems and Interoperability: a Discussion Grounded in Data / Rebeca Campos Motta – Rio de Janeiro: UFRJ/COPPE, 2016.

XIV, 150 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos.

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2016.

Referências Bibliográficas: p. 98-106.

1. Software Engineering. 2. Interoperability. 3. Context-Aware Software Systems. I. Travassos, Guilherme Horta II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

“Mas, buscai primeiro o reino de Deus, e a sua justiça, e todas estas coisas vos serão acrescentadas.”

Mateus 6:33

Agradecimentos

Agradeço a Deus, aquele que é poderoso para fazer - e que fez - infinitamente mais do que tudo quanto pedi ou pensei, me ajudando até aqui.

Agradeço aos meus pais, Ronaldo e Santafé, pelo amor e confiança depositados em mim, fundamentais para me tornar a pessoa que sou hoje. Aos meus avós, que me ensinaram, sendo exemplos de vida. À minha irmã Débora e ao meu primo Rafael, que me fazem enxergar o quanto cresci e como sempre posso melhorar. Ao Angelo, pela paciência, companheirismo e compreensão em todos os momentos, melhores que uma Brastemp.

Aos amigos de Campos, aos amigos do Rio, e aos amigos de Campos no Rio, em especial Ana, Alessandra, Jandira, Marianna e Mario Jorge, pois onde quer que eu esteja sempre me incentivam e me inspiram.

Desde jujubas a me consolar nas muitas vezes em que precisei, passando por e-mails de madrugada e lições de vida, agradeço ao professor Guilherme Horta Travassos por me orientar nesse trabalho. Seu incentivo constante e singular atuação foram essenciais para minha evolução acadêmica.

Agradeço aos pesquisadores Káthia Marçal de Oliveira e Victor Vidigal Ribeiro pela valiosa contribuição e pelo apoio durante esse trabalho. Aos companheiros do grupo de Engenharia de Software Experimental: Breno, Daniel, Hilmer, Jobson, Luciana, Paulo, Rafael, Talita e Thiago, pelo ambiente de trabalho, pelas conversas, pela companhia e por me ajudarem sempre que precisei. E também a Fábio, Felyppe e Hélvio, que tanto contribuíram e me aconselharam nesse período, pela amizade construída além do laboratório.

À Dani, Evandro, Brunno, Israel, os Renans e todos os amigos da liga, por fazerem essa jornada mais leve e divertida.

Aos professores Sheila dos Santos Reinehr e Toacy Cavalcante de Oliveira, por aceitarem o convite de participar da banca e por suas considerações.

Agradeço ao *Linnaeus Palme International Exchange Programme* pela oportunidade de intercâmbio, e aos amigos do *Blekinge Institute of Technology*. Agradeço também à equipe da UFRJ e da COPPE por todo o suporte, e ao CNPq e ao Projeto CAcTUS pelo apoio financeiro.

E a todos que colaboraram de alguma forma para esse trabalho, a minha mais profunda gratidão.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SOBRE SISTEMAS SENSÍVEIS AO CONTEXTO E INTEROPERABILIDADE: UMA DISCUSSÃO FUNDAMENTADA NOS DADOS

Rebeca Campos Motta

Junho/2016

Orientador: Guilherme Horta Travassos
Programa: Engenharia de Sistemas e Computação

Sistemas sensíveis ao contexto devem ser capazes de interagir com diferentes atores para completar suas tarefas e agir de acordo com o contexto, independentemente das suas diferenças de desenvolvimento ou organização. Sendo assim, a interoperabilidade é um desafio no desenvolvimento de sistemas desse tipo. Neste trabalho busca-se entender como a interoperabilidade é tratada em sistemas sensíveis ao contexto, reforçando as bases científicas para a compreensão da interoperabilidade e seus conceitos.

Um estudo secundário em trabalhos da literatura técnica foi conduzido para observar como a interoperabilidade é abordada em tais sistemas. Com base nos resultados da revisão da literatura, os princípios da Teoria Fundamentada nos Dados foram aplicados para analisar, organizar e relacionar os dados obtidos.

Como resultado, construiu-se um conjunto de conceitos e preocupações relacionados com questões de interoperabilidade em sistemas sensíveis ao contexto. O aprofundamento e utilização dos conceitos apresentados podem ser organizados e aplicados em forma de ferramenta de suporte para decisões de planejamento do sistema e avaliação da interoperabilidade.

O corpo de conhecimento gerado tem intenção de contribuir para o progresso da área de interoperabilidade em sistemas sensíveis ao contexto, e todas as atividades conduzidas bem como os resultados obtidos na pesquisa são detalhados nesta dissertação.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ON CONTEXT-AWARE SOFTWARE SYSTEMS AND INTEROPERABILITY:
A DISCUSSION GROUNDED IN DATA

Rebeca Campos Motta

June/2016

Advisor: Guilherme Horta Travassos
Department: Computer Science and Systems Engineering

Context-aware systems must be able to interact with different devices to complete their tasks and act according to the context, regardless of their differences in development or organization. Therefore, interoperability is a challenge in the development of systems of this kind. This work seeks to understand how interoperability is addressed in context-aware systems, strengthening the scientific basis for the understanding of interoperability and its concepts.

A secondary study was undertaken in technical literature to observe how interoperability is addressed in such systems. Based on the results of the literature review, the principles of Grounded Theory were applied to analyze, organize and relate the findings.

As a result, we constructed a set of concepts and concerns related to interoperability issues in context-aware systems. Deepening and using the concepts presented can be arranged and applied as a support tool for decisions related to systems planning and assessing interoperability.

The organized body of knowledge intends to represent progress in the interoperability in context-aware systems, and all activities conducted as well the results obtained in the research are detailed in this dissertation.

INDEX

INDEX OF FIGURES

INDEX OF TABLES

1 Introduction

This chapter introduces the work, presenting motivations and the problem addressed, as well the objectives. Furthermore, the research methodology applied is also presented in the chapter, ending with the dissertation organization.

1.1 Context

In his seminal work Weiser (WEISER, 1991) imagined a future where technology will be embedded in people's everyday life leading to the Ubiquitous Computing research area. In the last years, significant technological evolution made possible advances in several domains like medicine, manufacturing, and education. Advances in the communication field, like the wireless network, enable the development of new systems, smart devices and the Internet of Things. With the popularization of such devices, they become increasingly present in ordinary tasks of our daily lives. Reading the news, running or simply sending a message are now activities supported by technology, very similar to the future imagined by Weiser.

With popular and accessible technology more users are engaged with technical solutions, generating new demands and concerns. To address this growing demand, it is necessary to increase communication, awareness, and functionalities in such systems and devices.

The ubiquity concept is to exist or be everywhere, at the same time. To that end, ubiquitous computing aims to develop computing technologies integrated with objects in a seamless manner, so they become indistinguishable (WEISER, 1991). It seeks to create useful systems to be present in different situations encountered in the real world and observe its impact on the user perspective (ABOWD, 1999; TANG *et al*., 2011).

Various devices (sensors, computers, mobile technologies) and services should interact automatically to achieve the envisioned objectives. Aside the diversity of interacting devices, ubiquitous systems have particular features, like context sensitivity and transparent interaction, which require differentiated development approaches. It is necessary to develop new research in software engineering to deal with the diversity and particularity of ubiquitous computing to support its development (ABOWD 1999; NIEMELÄ and LATVAKOSKI 2004; SPÍNOLA, MASSOLLAR, and TRAVASSOS 2007; ATZORI *et al.*, 2010; ROSE *et al.*, 2015).

1.2 Motivation

In technical literature, works like (CONSOLVO; ARNSTEIN; FRANZA, 2002; O'NEILL et al., 2005; SPÍNOLA; PINTO; TRAVASSOS, 2009; ROCHA et al., 2011) discuss open challenges to develop, assure quality and evaluate such systems. The concerns reported in these studies cast light on complexity when dealing with ubiquitous systems, and the research gaps must be addressed to meet particular features such as context sensitivity, transparent interaction, and adaptive behavior. Dealing with ubiquitous systems and devoted effort to this research is not a trivial task since these systems encompass characteristics of distributed, heterogeneous and user-centered software.

To establish a particular approach to ensure quality and allow improvement towards usefulness in ubiquitous systems the CNPq Project called CAcTUS – Context-Aware Testing for Ubiquitous Systems – was started. Two major interests of the project were considered in this work: actor-computer interaction and context-awareness.

In the project perspective, the systems relations are evolving alongside with technology and surpass the human-computer interaction. The project introduces the idea of actor-computer interaction, where an actor is not only a human but any other computer, device, object or application present in a ubiquitous system.

Context-awareness is a characteristic of ubiquitous systems (SPÍNOLA, 2010). This feature enables to use information about the user, his/her behavior and the environment they are inserted in to provide services and enhance the user experience. Context is any information that can be used to characterize the situation of an entity (DEY, 2001). An entity can be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user himself or herself and the application itself. To capture and be aware of the context, systems use distinct sensors and applications, and this configures the interactions when dealing ubiquitous systems. Different types of actors, interactions and relevant context should be considered in the systems' quality evaluation.

The CAcTUS Project aims to define a strategy for planning and executing testing procedures considering systems' context. The project concerns to tests performed to ensure the best actor-computer interaction taking into account the possible contexts of use as well as the methods for designing these tests. To investigate this matter, the researchers initiate the project undertaking secondary studies aiming to characterize the state of the art of the field. At present two other *quasi*-systematic literature reviews, one addressing test strategies (RODRIGUES; MATALONGA; TRAVASSOS, 2014) and other regarding test case design (SANTOS; ROCHA; ANDRADE, 2015) were also undertaken.

The CAcTUS Project plan to characterize the state of the art for context-aware software testing can be observed in Figure 1.1.

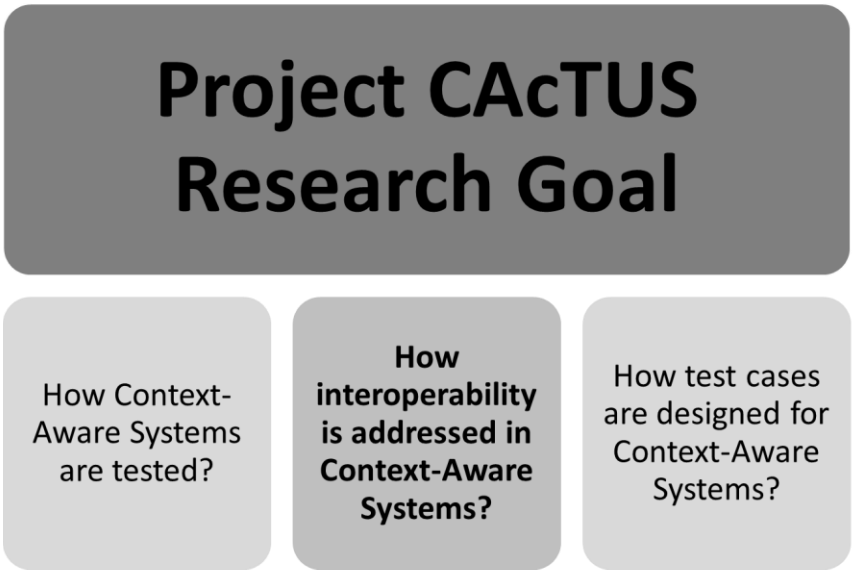

Figure 1.1 Project CAcTUS Research Goal

One of the drivers for this dissertation was derived inside the CAcTUS Project, considering that systems should be able to interact with different actors, achieve their purposes, to complete their tasks and act according to the context regardless their differences in development or organization.

1.3 Problem

Interoperability is long seen as an important systems' property (IEEE, 1990) and is currently referred as a significant technical challenge in the development of ubiquitous systems (COSTA; YAMIN; GEYER, 2008; ROCHA *et al*., 2011). The issue often reported is that current practices are not suitable to address all interoperability particularities in such systems (STRANG; LINNHOFF-POPIEN, 2003; GRACE; FLORES; BLAIR, 2009) due to some key aspects:

- **Heterogeneity:** society's relation with systems is trickier nowadays because the computer is no longer closed in a room. It consists of a specialized set of components like sensors, embedded devices, computers, mobile phones, web services and others. These components are developed and connected using broad range solutions, going from different programming languages to transport protocols creating complex systems-of-systems (NIEMELÄ; LATVAKOSKI, 2004; COSTA; YAMIN; GEYER, 2008; GRACE; FLORES; BLAIR, 2009; ATZORI *et al.*, 2010).

- **Dynamicity:** the world itself is in constant change, everyday situations are dynamic and diverse. Computing is not only about computers anymore, as technology is present in several manners in people's life and also integrated with the environment around them. To handle such dynamicity, it is necessary to be aware and deal with context variance, particularly in the runtime (BLAIR; BENCOMO; FRANCE, 2009; GRACE; FLORES; BLAIR, 2009; TANG *et al*., 2011; ROSE *et al*, 2015).

The main problem addressed in this work is derived from these aspects. Interoperability is an essential characteristic and comes as a key requirement to cope with heterogeneity issues, aiming at making systems compatible and harmonious. Context-aware is the feature that enables adaptive or reactive behavior in systems. In consideration of this matter, this work attempts to observe the relation of these concepts considering the interoperability impact when a context-aware system should interact with others. The research is based on the expectation that considering a new factor, as context variance, systems interoperability is affected, adding complexity and overlooked difficulties. To observe whether the expectation is confirmed, the research question derived from the problem stated is: **How interoperability is addressed in context-aware software systems?** This issue drives this research and all the activities performed.

1.4 Objective

The interoperability is a well-established research topic, studying issues related to lack of interoperability and innovative solutions for problems in several areas for almost 40 years (FORD *et al*., 2007). However, despite years of research, the majority of solutions to address interoperability issues are limited to standards. Standards are usually domain specific concerning particular areas and in some cases allow different interpretations (FANG; HU; HAN, 2004). Standards are necessary to achieve interoperability, but it does not evolve at the same pace as technologies, so more research is required to deal with emerging technologies such as context-aware systems.

This work aims to enhance the scientific foundations of interoperability to contribute to the development of the field. Different studies have been performed to present approaches to achieve interoperability, as well interoperability attributes and how to measure them (LEITE, 1998; FORD *et al*., 2007; REZAEI *et al*., 2014). Interoperability topic can be seen along with technical, syntactic, semantic and organizational levels which increase the complexity and also research opportunities.

Considering these interoperability aspects (approaches, attributes, and levels), joining them with the particularities of context-aware systems, the research activities of this work **have the objective to build a set of concepts and concerns related to interoperability issues derived from technical literature**, as following:

- **General Concepts:** an examination of fundamental interoperability concepts, principles, and ideas, introductory strategies to describe issues related to it, observe patterns, similarities, and differences between them.
- **Particular Concepts:** development and refinement of the general interoperability concepts in the light of context-aware systems.
- **Mechanisms:** necessary decisions and significant concerns to translate these concepts into actions to achieve interoperability.

1.5 Methodology

Considering the objectives of this research, the method used for this work followed the steps below and are presented on a timeline in Figure 1.2. Each step is detailed and justified later on their specific chapters.

1. Selection of the research topic: the CAcTUS Project envisioned the starting point of this investigation. The initial setting was carried out in discussions among project participants regarding the pertinence and viability of the research.

2. Definition of the research problem: this step had the purpose of delimiting the problem and identifying initial concepts related to interoperability in context-aware systems. An initial *ad-hoc* review of the technical literature was necessary to align the research perspective, as well to support the planning in the next step.

3. q*uasi-S*ystematic Literature Review: the review was performed following the suitable to prepare the protocol as well its execution. It was chosen to execute this secondary study in order to proper address the research question and to characterize the current state of research regarding interoperability.

4. Data analysis: the analysis process was performed using the guidance of the Grounded Theory methodology. The conceptual set represents interoperability aspects retrieved from the literature, how they relate and decisions towards interoperable systems.

5. Interpretation of findings: discussions held and the results observed after the data analysis. The results led to the theoretical framework and answers to the research questions.

6. Final report: document the whole procedure in a dissertation format.

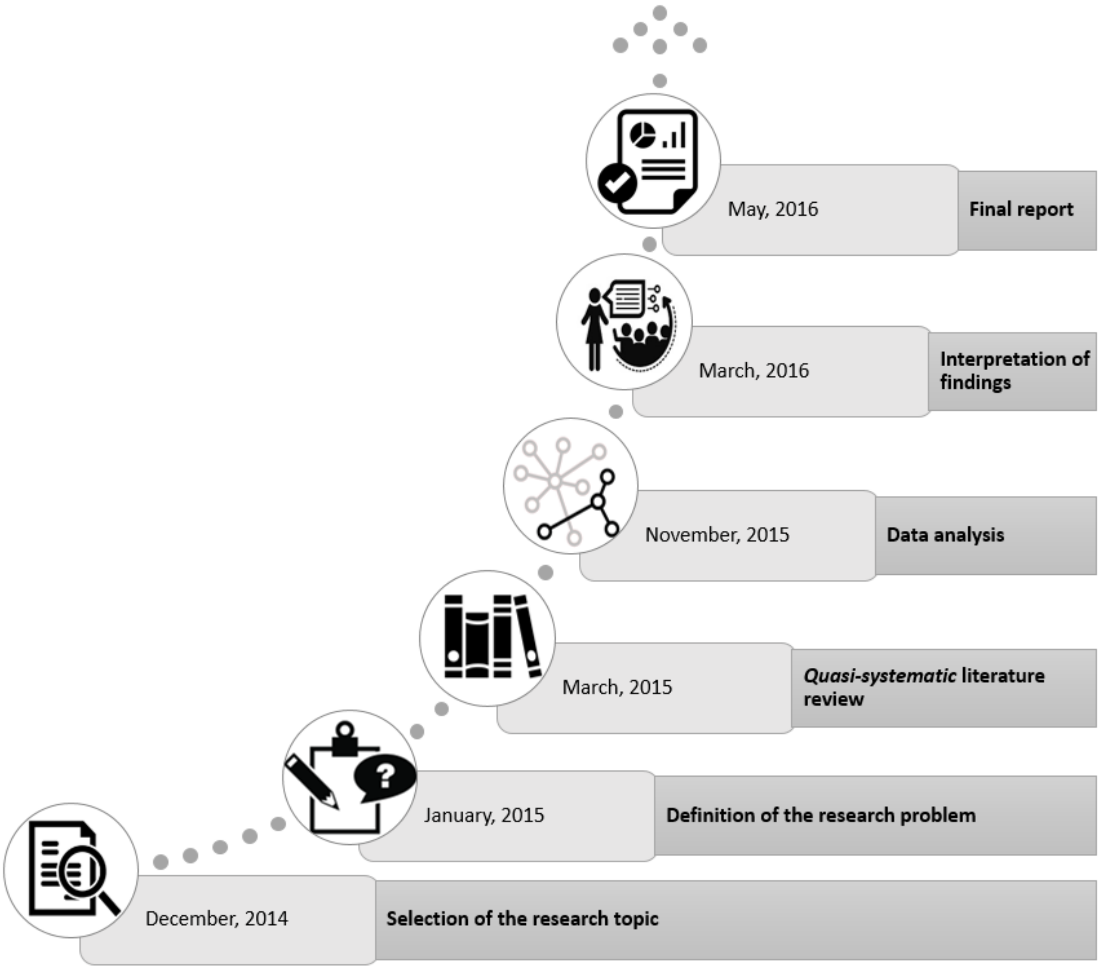

Figure 1.2 - Steps of the research method applied

1.6 Organization

The dissertation is organized into five chapters, including this introductory chapter, as follows:

- **Chapter 1 – Introduction:** this first chapter introduces this work, presenting motivations and the problem addressed, as well the objectives and research methodology applied.
- **Chapter 2 – Concepts and Definitions:** the basics used during the development of this research are described in this chapter. This chapter present theoretical basis of Ubiquitous Computing, Context-Aware Systems, and Interoperability.
- **Chapter 3 – *Quasi*-systematic Literature Review:** this chapter presents the process and results of the secondary study undertaken to characterize interoperability in Context-Aware Systems.
- **Chapter 4 – Qualitative Analysis: t**his chapter reports principles and usage of Grounded Theory, to analyze the data retrieved from the *quasi*-systematic

literature review aiming to inter-link concepts and relate them into higher abstractions synthesizing the results.

- **Chapter 5 – Interoperability Characterization:** the findings of the literature review turned into concepts after the analysis performed. These concepts are discussed in this chapter, characterizing interoperability and answering the research questions that drives this work
- **Chapter 6 – Conclusions:** the conclusion of this work is presented in this chapter. We outline the main contributions of this research and some limitations. Finally, some possible paths for future work are described to continue the research herein presented.

2 Concepts and Definitions

Described in this chapter are the basics concepts used during the development of this research. This chapter presents theoretical basis of Ubiquitous Computing, Context-Aware Systems, and Interoperability. The definitions and ideas presented here intend to drive the reader into a common perspective and understanding of this work.

2.1 Introduction

> *"Sal awakens: she smells coffee. A few minutes ago her alarm clock, alerted by her restless rolling before waking, had quietly asked "coffee?", and she had mumbled "yes." "Yes" and "no" are the only words it knows. Sal looks out her windows at her neighborhood. Sunlight and a fence are visible through one, but through others she sees electronic trails that have been kept for her of neighbors coming and going during the early morning. Privacy conventions and practical data rates prevent displaying video footage, but time markers and electronic tracks on the neighborhood map let Sal feel cozy in her street. Glancing at the windows to her kids' rooms she can see that they got up 15 and 20 minutes ago and are already in the kitchen. Noticing that she is up, they start making more noise. At breakfast Sal reads the news. She still prefers the paper form, as do most people. She spots an interesting quote from a columnist in the business section. She wipes her pen over the newspaper's name, date, section, and page number and then circles the quote. The pen sends a message to the paper, which transmits the quote to her office. Electronic mail arrives from the company that made her garage door opener. She lost the instruction manual, and asked them for help. They have sent her a new manual, and also something unexpected -- a way to find the old one. According to the note, she can press a code into the opener and the missing manual will find itself."* (WEISER, 1991)

The excerpt above represents the daily life of Sal in Mark Weiser's vision of a future where Ubiquitous Computing is a reality (WEISER, 1991). Since Weiser's seminal work much progress has been made to make ubiquitous computing real. Nowadays it is possible to imagine several other scenarios and to develop systems that are aware of the current context and act according to the information perceived. Nevertheless, as we can see in this chapter, there are still open challenges to address.

2.2 Ubiquitous Computing

2.2.1 Definition

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." (WEISER, 1991). Weiser considered the writing as the first information technology, and he observed how it has became ubiquitous – natural and ready for use, at a glance. With this comparison, in the ubiquitous computing paradigm computers are not limited to desktops or windows, but present in every place and everything. It is present in ordinary objects that support people in everyday activities, without the perception of using technology, just a fluid, and natural interaction.

After Weiser's first proposition, the research in the area and the technology itself have evolved together with new concepts and definitions for ubiquitous computing like the definitions presented below (JEON *et al*., 2007):

Table 2.1 - Ubiquitous Computing Definitions (JEON *et al*., 2007)

Author	Definition
(SAKAMURA, 1987)	*Ubiquitous computing is being able to use computers anywhere at any time.*
(WEISER, 1993)	*Ubiquitous computing has as its goal the enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user.*
(MATTERN, 2001)	*Everyday objects will become smart, and they will all be interconnected.*
IBM (JEON *et al*., 2007)	*Pervasive computing delivers mobile access to business information without limits from any device, over any network, using any style of interaction. It gives people control over the time and the place, on demand.*
(LEEM *et al*., 2005)	*Ubiquitous computing is a technology, in which invisible computers are embedded and connected with all things so that anyone can communicate, exchange and share information anywhere anytime.*

In their definition, Jeon *et al.* (2007) consider ubiquitous computing as "technology environment in which computers are invisibly embedded in every object and linked with each other so that information sharing and communication are possible in anywhere, anytime and anyplace."

Regardless of a standard definition and despite the differences between the ones presented above, ubiquitous computing is becoming a reality, and the similarities among the research can guide to a shared understanding of what makes a system.

To illustrate this world where connectivity and computing extend to objects, sensors, and daily elements we have the new term of "Internet of Things" (*IoT*). The idea is that *IoT* allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service (GUILLEMIN *et al.*, 2010; ROSE *et al.*, 2015). Things, in turn, are any smart object or device that provide services or information, changing the perspective of static to active participation in this new internet paradigm. This progress releases the computing from the traditional boxes and windows and embeds it into everyday life objects – like the proposal of ubiquitous computing. These objects can be seamlessly integrated into the environment closing the distance from virtual e physical reality. The particularities of each object, the environment of their use, preferences of the users and so on are relevant information in this scenario that can lead to new ways of interacting with objects; that can be somewhat unpredictable (ROSE *et al.*, 2015).

2.2.2 Characteristics

To limit and categorize ubiquitous systems the work of Spínola, Massollar and Travassos (2007), through a systematic literature review, presents ten characteristics of a ubiquitous system:

- **Adaptable behavior:** the ability to, dynamically, adapt and offer services according to the environment in use, within its limitations.
- **Context sensitivity:** the ability to collect information from the current environment.
- **Experience capture:** capacity to capture and register experiences for later use.
- **Fault tolerance:** capacity to adapt itself when facing environment's faults (for example, on-line/off-line availability).
- **Function composition:** the ability to, based on basic services, create a service required by the user.
- **Heterogeneity of devices:** provides application mobility among heterogeneous devices. That is, the application could migrate among devices and adjust itself to each one of them.

- **Invisibility:** the ability to be present in objects of daily use, weakening, from user's point of view, the sensation of explicit use of a computer and enhancing the perception that objects or devices provide services or some "intelligence". With that, it is possible to find proper alternatives for traditional graphical interfaces used on desktop solutions for more natural ways of data input in such a way that the user will minimally perceive the interface itself.
- **Service discovery:** pro-active discovery of services according to the environment where it is being used. The application has to interact with the environment and allow the user to do the same, to find new services or information to achieve the desired purpose.
- **Service omnipresence:** it allows users to move around with the sensation of carrying services and devices with them.
- **Spontaneous interoperability:** ability to change partners during its operation and according to its movement.

This work (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007) also states possible differences between projects, leading to presence or absence of some of the characteristics, but it remains a ubiquitous system, nonetheless. The presented characteristics are at a high abstraction level to define their significance and be a starting point for further research. Still, it is possible to see all the features in a more concrete way, because each one has its attributes. For instance, context sensitivity is a characteristic of ubiquitous systems and having it has its attributes and implications, as presented in the next section.

2.2.3 Examples

Returning to this section introduction, in the life of Sal in Mark Weiser's vision, we see daily activities supported by ubiquitous technology. Initiatives as the ones presented here are working towards making ubiquitous computing a reality (AITENBICHLER *et al*., 2007; RODUNER *et al*., 2007).

In the work of Aitenbichler *et al.* (2007) a regular off-the-shelf coffee machine was adapted for RFID readers, and it can detect users' actions to start processes automatically. With five major components, the general architecture introduced in this work enables the user to get his favorite coffee automatically, once the coffee mug is under the coffee dispenser. The validation performed had 11 subjects for user testing purposes.

Another interesting study provides an observation of users caring out everyday tasks using a mobile phone (RODUNER *et al*., 2007). The study conducted had 23 subjects using a prototype to perform 18 tasks in four different categories. Tasks like to adjust water hardness in a dishwasher, setting a clock on time on a radio and brewing coffee from a coffee maker were all made through a mobile phone and not with the traditional manual interaction.

"What starts on paper, doesn't have to stay there[1]." Chapman, Lahav, and Burgess (2009) conducted a series of studies - ethnographic interviews, usability laboratory studies, focus groups, field trials and corporate site visits - to investigate note-taking activity with a digital pen. This pen is used on a specially patterned paper to record the writings in a digital format that can be later downloaded, reviewed or printed. In a similar path, the work of Despont-gros *et al.* (2005) presents a pen with standard ink cartridge, camera, communication unit, pressure captor, image processing unit, storage unit, and battery. An operator used this pen and its technology in a clinical scenario, to collect and processes handwritten questionnaires related to mothers´ satisfaction after labor analgesia. The study used the pen to digitalize the forms automatically and tested in two different settings to evaluate reliability and acceptance. The reliability proved to be similar to the regular pen, and the advantage is the direct acquisition of information.

A technological assistance to find a lost object, as a manual for example, can be FETCH (Finding Everything using Technology Convenient and Handy). This solution was motivated by focus group and then evaluated in other two studies (KIENTZ *et al*., 2006). Developed to assist the visually impaired, the FETCH uses Bluetooth tags to implement a tracking system. These tags are attached to objects, and the system is controlled via mobile phone, and a tag makes a beep to locate an object. A related work (NAKADA; KANAI; KUNIFUJI, 2005) presents a device with two different position location methods used to enhance detection accuracy. In their system, the objects are tagged, and a list is made. If the object is lost, the user selects it from the list (with passive RFID) and "the moving light" is responsible for notifying the user the detected object by illuminating the location of the lost object in a room.

These examples address the initial scenario presented in the introduction, and a complete ubiquitous research can be seen in the Aware Home Research Initiative[2] at the Georgia Institute of Technology (KIENTZ; JONES, 2008). Most of the works presented here are academic published examples driven by research purposes, with no commercial objectives, but many of these solutions can be permeated into our lives in other ways.

[1] http://www.livescribe.com/pt/smartpen/

[2] http://www.awarehome.gatech.edu/

2.2.4 Challenges

Although this research area began in the 90´s, it is still active nowadays with major progress after the popularization of Internet and mobile phones. Several works presents research challenges in ubiquitous computing like the ones described below (EDWARDS; GRINTER, 2001; WANT; PERING, 2005; LOUREIRO *et al*., 2009):

- **Quality:** it is a concern comprising context quality and quality of services for example. More research should be conducted towards approaches to deal with conflicts, improve the identification of relevant information together with verification and validation procedures.
- **Sensing:** it is necessary to improve the sensing devices including the dynamic choice and inclusion of adequate context variables to each application. For significant context sensitivity, it is also important to consider context variation and the context source to assure the quality and apply standard semantics through all the process.
- **Reliability:** the new ubiquitous applications should be more reliable than regular desktop software systems and as functional as a refrigerator or television, for example. Considering differences in development culture, technological approaches, market expectations, and legal issues, these smart applications have to be designed for reliability, which requires resources and affects the system architecture.
- **Interoperability:** the premise is that a ubiquitous environment will be intelligible to components that may be from different vendors and appear in different moments. Whether it is called spontaneous or impromptu, true interoperability, in this case, wouldn't be just interaction itself, but interacting with no foreplanning or user intervention.

2.3 Context-Aware Software Systems

2.3.1 Definition

Context sensitivity is one of the ten characteristics of ubiquitous systems (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007) seen in the previous section and it is part of the scenario proposed by Weiser (1991).

The world has a complex multi-dimensional contextual space with different levels of importance that can change over time. To translate the world into computing technologies when considering context, it is necessary to state the contextual variables to be used. Also, it is important to consider the difference in context perception. Humans tend to respond unconsciously sometimes to context variance via sensation, cognition, and emotion. With this natural answer, one can underestimate the distinct and complex task of acquiring, storing, controlling and using contextual information through systems. As computing is no longer limited to computers and so integrated with the environment, the interaction with the user becomes almost imperceptible. More frequently, systems can adapt their behavior according to the information they receive about the environment or the user's, and this information is the context the systems should be aware.

The work of Bazire and Brézillon (2005) after collecting and analyzing 150 definitions of context they state that the definition depends on the research area. Despite many different definitions of the concept, researchers agree that (VIEIRA; TEDESCO; SALGADO, 2011):

- Context exists when related to another entity (e.g. person, place, object or interaction).
- Context is a set of items (e.g. concepts, rules, and propositions) associated with an entity.
- An item is considered as part of a context only if it is useful to support the problem at hand.

The present work is based on an embracing and well-referred definition provided by Dey (2001):

> *“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”*

It is necessary first to understand and delimit context to deal with context-aware systems. After this brief presentation of context concepts and its definition, we can now address context-aware systems. To delineate the field this work is inserted, we refer to the considered first work mentioning the “context-aware” term (SCHILIT; THEIMER, 1994). In their work, “context-aware systems discover and react to changes in the environment they are situated in”, and context information refers to the users’ location, identities of nearby people and objects, and modifications to those individuals and objects. Therefore, the systems are continuously monitoring or subscribing information about the world around them. Dey (2001) considers systems as context-aware if it “uses context to provide relevant information and services to the user, where relevancy depends on the user’s task”. Based on these and other definitions, a system can use context information to enhance systems’ natural behavior or as a trigger to its adaptation, and it is up to the developer to decide what information should be considered relevant for the system.

2.3.2 Characteristics

As context definitions can change depending on the research area, the context can change depending on the system. For most of the systems, context is more than a small set of retrieved information. A computer system requires entries to perform tasks and provide outputs. Users manually provided traditionally information, but with the use of context-aware approach the physically perceived world can be computationally incorporated.

Context-aware applications should be able to retrieve **who's, where's, when's** and **what's** related to an entity. From this information, it is possible to answer **why** a situation is happening (DEY; ABOWD, 1999). Identity (**who**), location (**where**), time (**when**) and activity (**what**) are necessary to characterize a particular situation. These context pieces of information are not limited to answering these questions but are sources of further relevant information. Take **who** for example. The user´s identity is not only his name, but from that, the system can acquire related information such as phone numbers, e-mail, and friends list.

Context-aware systems may have an elaborate architecture and several components, but in general, it consists of four steps to process context-awareness presented in Figure 2.1 (LEE; CHANG, 2010).

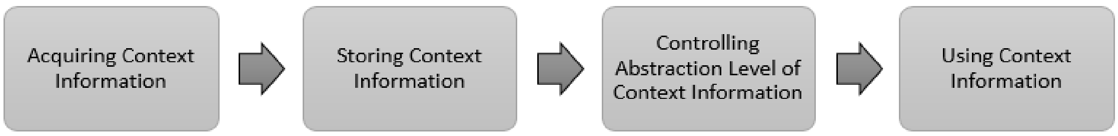

Figure 2.1 - General process of Context-aware Systems adapted from (LEE; CHANG, 2010)

- **Acquire:** context information is diverse and can be acquired in several ways, as from a person's profile, sensors or derived from historical data for example (LEE; CHANG, 2010; VIEIRA; TEDESCO; SALGADO, 2011).
- **Store:** to Lee and Chang (2010) is important what data model is used to represent context information. Dey, Abowd and Salber (2001) state that the information should be stored not just for current usage, but also for later retrieval, keeping historical data with relevant context information.
- **Control:** control in the abstraction level is related to interpreting or aggregating data (LEE; CHANG, 2010). Context aggregation means changing raw data to high-level information. Context interpretation takes context information and enhances it with semantics.

- **Use:** refers to the step that presents the context to the user, it uses context information to propose useful actions, to trigger a command in the application or change systems behavior according to context variance (SCHILIT; THEIMER, 1994; DEY, 2001).

It is of great importance to make it clear that not everything is context information, as it should be according to what is relevant to a particular system. The complexity to consider every variance in every possible piece of information would be overwhelming. Therefore, it is necessary to define what is seen as context information for a given system that helps to determine which behaviors the system will support when relating to context. Take the systems purpose, for instance, it can vary from a system to another, and the context should be according to it.

2.3.3 Examples

During the research for this work, we came across several articles that provide an extensive review of the area and several examples of context-aware systems in diverse applications. These works provide a glimpse of the use of context-aware characteristic as a key factor for ubiquitous computing development (CHEN; KOTZ, 2000; BALDAUF; DUSTDAR; ROSENBERG, 2007; HONG; SUH; KIM, 2009; LEE; CHANG, 2010).

The work of Cheverst *et al.* (2000) is considered to be one of the first context-aware applications. The GUIDE is a touristic application that provides information and support navigation to tourists in the city of Lancaster. As requirements, the system should have flexibility, support for dynamic information and interactive services, also be tailored to visitor´s context. They handled with context information in two forms: personal (interests, preferences, location, and others) and environmental (time of day, opening times of attractions, and the distance between attractions and others). The evaluation performed had an expert walkthrough (with four usability experts) and field trial (60 volunteers during four weeks). The conclusions let no doubt regarding the appreciation of the system’s benefits by the public.

Goularte; Pimentel and Moreira (2006) propose a model to describe documents (frames, videos, scenes, series) for interactive TV. Based on the Dey’s definition of context, each document is characterized by a set of contextual information implemented via an abstract type **PrimaryContext,** composed by **IdentityType**, **LocationType***,* **TimeType**, and **ActivityType** – which represent Dey´s dimensions of **who**, **where***,* **when** and **what**. This initiative is to present interactive content associated with metadata enabling relationships among objects with high-level semantics.

Context-aware systems are now present in other domains and with constant growth. An application like a smart agenda that can give valuable information based on the student identity and their current location in Cornell University Campus. Information like attractions in a specific building, test dates for a class are available (BURRELL; GAY, 2002). The system in question, called e-graffiti, was used by 57 students in a six-month period for evaluation purposes with constructive reviews from the students' part.

The work of Favela *et al.* (2004) introduces interactive displays in hospitals with context information. Through instant messaging, information as availability, location and status of patients or staff are updated, then extended to open view on the public display. The system was evaluated by 28 hospital staff members using an acceptance model and a case study where all participants report the sense of immediate and easy access to the medical information to be beneficial to a hospital environment.

Beyond these examples, there are many other not limited to academic research but in actual usage. Screen orientation changes due to the device movement, user's current location make a map application to be automatically updated, light adjustment of the phone, when used in a dark room are examples of systems aware of their environment and their context of use. Still, there are several issues to be handled regarding context-aware systems.

2.3.4 Challenges

Despite the benefits and usefulness of context-awareness, it is necessary to address some issues and investigate new solutions to make it a more viable and reliable technology:

- **Context definition:** among the challenges in this area, defining context is of significant importance when developing context-aware systems. If out of its context of validity, a system can lead to the incorrect use of knowledge. Also, with infinite contexts, it is not possible to confirm what is needed, bringing difficulties with terminology and characteristics related to context as well (BAZIRE; BRÉZILLON, 2005; VIEIRA; TEDESCO; SALGADO, 2011).
- **Security and privacy:** protecting user's privacy and context sensitive information should be a priority, and several current systems lack security mechanisms (BALDAUF; DUSTDAR; ROSENBERG, 2007; LEE; CHANG, 2010).
- **Context variation:** despite the use of different terms, the technical literature shows that context is dynamic and should be treated as so. Dealing with problems associated with the impact of variation and several context information at once is still an open challenge (DEY; ABOWD, 1999; VIEIRA; TEDESCO; SALGADO, 2011; MATALONGA; RODRIGUES; TRAVASSOS, 2015).

- **Interoperability:** the wide variety of devices, platforms, protocols and interfaces contributes to increasing the complexity of development and the heterogeneity among systems' components that can lead to incompatibility, which can impair system interactions (PASCOE; RYAN; MORSE, 1999; NIEMELÄ; LATVAKOSKI, 2004).

2.4 Interoperability

As seen throughout this chapter, interoperability appears as a challenge for context-aware systems and ubiquitous computing. With Weiser's' vision turning into reality, we see more often solutions like the examples presented in the previous sections, an increasing number of systems developed for TVs (GOULARTE; PIMENTEL; MOREIRA, 2006), universities (BURRELL; GAY, 2002), hospitals (FAVELA *et al*., 2004) and houses (KIDD; ORR; ABOWD, 1999; KIENTZ; JONES, 2008).

To deploy applications in such different fields for the most diverse purposes, it is necessary to support a variety of components with features and functions suited to the applications objectives. The difficulty is to achieve interoperability between things, components, and systems. This work is an attempt to address the open challenges in this topic as presented in the introduction. Therefore, the terms and concepts in the perspective of the author are presented in the following sections. This part of the chapter is to provide an overview of general interoperability and the gaps this work can fill when related to context-awareness.

2.4.1 Definition

Initially, the interoperability term was limited to information systems but with the progress of research and interest of several organizations, it became more complicated than imagined at first sight. Part of this work was to delimit the perspective of interoperability we would like to address. It is important to understand the general view of interoperability, starting by its definitions and is fascinating to observe differences and similarities between the ones we came across throughout this work. Terminology problems are current in this research area, and, as expected, there have been several definitions of interoperability proposed by researchers, standards bodies or governments over the years. The work of Ford *et al.* (2007) identified thirty-four distinct definitions for interoperability, and the oldest definition they found is from 1977 by the American Department of Defense:

The ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together.

Below we can see one of the most used and referenced definitions, presented by IEEE (IEEE, 1990):

The ability of two or more systems or components to exchange information and to use the information that has been exchanged

These definitions are the most popular when referring to interoperability. Therefore, they have been chosen to be used as ground for this work, with interoperability related to systems in a general view, and we can see some particularities in both of them:

- The first definition of the Department of Defense limits the object of interoperation to services; and being from the Department of Defense units and forces are from military context (not relevant to systems' perspective). **Provide and accept** have the idea of choice. Providing the service only is not enough. Accepting the service only is not sufficient. The service should be used to complete the proposed definition and to operate **effectively** together. However, what does **effectively** mean?
- The latter definition from IEEE is strictly of systems and their components, bearing more similarities with our perspective. In a systemic view, a system is a set of parts forming a unified whole; these parts are here named as components. The understanding of components includes different points of view, as it can go from small modules to services in an enterprise for example. This definition is limited to **information** exchange. One can argue that in the end everything is information at a **bit** format, and that is true. However, the complexity and evolution of today's systems being smarter and bigger can lead to a discussion whether it is only information when concerning to perception or recognition, and comprehension as well.

In the similarities, we can observe that interoperability does not refer to one system alone, it always relates to systems interaction (two or more). Also, only an exchange does not define interoperability, but the object of interoperation should be used for a purpose. At last, interoperability is domain specific. We know that systems are developed for different areas, but that are different business domains like industry, health or military, and each one of them has specific interoperability needs. These ideas permeate this work, and a single definition to cover current definitions should be flexible enough to adapt, embrace them and comprise the differences.

2.4.2 Characteristics

The definitions presented (FORD *et al*., 2007; IEEE, 1990) are widespread through other technical papers and based on these works we also show some interoperability characteristics as a starting point.

The original directives of the American Department of Defense could not be accessed. Therefore, the information presented here is based solely on the work of Ford et al. (2007). This work does not propose or list interoperability characteristics. Instead, it surveyed interoperability measurement models, and it is from these models that we can retrieve interoperability characteristics. From the survey, fourteen models are analyzed each one has a particular focus with their different benefits or limitations. To confirm the relevance of the models we used the work of (REZAEI et al., 2014), a systematic review on interoperability evaluation models, where ten models were analyzed and compared. The intersection of these two works has eight models in common. As this section does not intend to be an extensive review of interoperability evaluation models, we chose to summarize and present the interoperability characteristics considered in two of them, with high citations rate when searching using the Scopus Database and the that we could access the original work.

- **Interoperability Assessment Methodology (LEITE, 1998):** through the measurement and quantification of a set of interoperability components observed in systems that should interact the methodology proposes the interoperability assessment to be the sum of the characteristics of the components. As interoperability components, we identify the following characteristics:
    - **Data Elements:** the data itself and the structure of data elements into messages.
    - **Information Flow:** the volume of data considering operation time.
    - **Information Utilization:** provides data, correctly interprets it and verifies if the proper action was taken.
    - **Interpretation:** confirms the consistency of the transmitted data and examines how individual processors interpret the data received.
    - **Latency**: The elapsed time from the request to reception.
    - **Node Connectivity**: the ability to send and receive data. It implies that the transmitter and receiver are available at any time.
    - **Protocols:** protocols determine access to data and are based on the information exchange requirements.

    - **Requirements:** all systems where interoperability is desired must have shared operational requirements.
    - **Standards:** to be interoperable, systems should have a common implementation of standards to define transmission and data acceptance.
- **Levels Of Information Systems Interoperability (C4ISR ARCHITECTURE WORKING GROUP, 1998):** the assessment is in a questionnaire format and interoperability profiles. Interoperability is considered in four levels (**Isolated, Connected, Functional, Domain**, and **Enterprise**). In their proposal, each level may have influence factors which they organize into four key attributes:
    - **Procedures:** encompass the many forms of documented guidance and operational controls that affect system development, integration, and functionality, considering standards, management, security policy, and operations.
    - **Applications:** the fundamental purpose of which any system is built and the functional requirements specified.
    - **Infrastructure:** what supports the establishment of systems operations and interactions, considering network, hardware, security equipment and services.
    - **Data:** regarding the exchanged data, dealing with data format (syntax) and its content or meaning (semantics).

Based on the IEEE (1990) definition, the ISO standard (ISO/IEC WD, 2011) presents a form to observe "the degree to which two or more systems, products or components can exchange information and use the information that has been exchanged." They see interoperability in two parts: **Internal** (set of attributes for assessing the capability of the system product's interaction with designated systems) and **External** (the object transferred easily between the system and the other system connected). Considering this perspective, they present two attributes:

- **Connectivity with the external system**: how correctly the external interface protocols have been implemented, when comparing the number of correctly implemented interfaces with the total number of external interfaces.
- **Data exchangeability**: how accurately the implementation of data exchange format is, determined between linking systems when compared the number of data formats exchanged with the total number of data formats to be exchanged.

From this first attempt to list interoperability characteristics we can see that it is similar to its definition: it varies according to the domain, being hard to address this multifaceted concept at once.

There are other issues related to interoperability that is not directly addressed in these works. Like in the work of Ford *et al.* (2007), where they report a concept called **interoperability type**. Sixty-four different types of interoperability are presented in a table form where the authors briefly classify them as either technical or non-technical types but left it open to interpretation. From this initial review and other activities conducted in our work we understand **interoperability type** as **interoperability perspectives**, which are related to interoperability interpretation in diverse areas and can change the interoperability definition and its needs, further discussed in the following chapters. The Levels Of Information Systems Interoperability model (C4ISR ARCHITECTURE WORKING GROUP, 1998) introduces the concept that we address and call here as **interoperability level**, the level of support provided for interoperability in a given interaction. This model gives just a glimpse of the concept that we develop in this work, being the initial dimensions: **technical**, **syntactic** and **semantic**, and are broadened through this work.

2.4.3 Examples

As seen in definition, interoperability stands for systems interacting for a purpose and can be related to any system. Examples of applications and systems related to interoperability permeate this work. In this section, we return to two examples given in previous sections, because in these, while describing the technical background of their applications, the authors also present the concern with interoperability (DESPONT-GROS *et al*., 2005; GOULARTE; PIMENTEL; MOREIRA, 2006).

The clinical information system in (DESPONT-GROS *et al*., 2005) covers design, development, and support for the management of medical knowledge. The authors report details of the implementation where the architecture uses event-driven processes focused on interoperability with Web Services.

Goularte, Pimentel and Moreira (2006), while describing multimedia content report the use of standards as a way to deal with metadata semantics. The benefits of the decision of having standardized schemas is to maximize interoperability and "a practical way to use the standard is to make a selection of the description structures needed by the application".

2.4.4 Challenges

The challenges presented here are not just the result of an overview of the interoperability topic but represent practical issues that deserve the effort of bein develop if the envisioned future is not one of incompatibility but with simple, seamless and fluid interaction.

- **Extend interoperability concept:** most definitions of interoperability are limited to current technologies and do not comprehend the aimed progress with ubiquitous computing (ZDRAVKOVIĆ; NORAN; TRAJANOVIĆ, 2014). These definitions assume a previous agreement between the systems regarding their behaviors in an interaction. Removing these agreements and extending the concept is a step towards the next challenge of spontaneous interoperability (EDWARDS; GRINTER, 2001).
- **Spontaneous interoperability**: under continuous change with new users and devices being connected to systems at runtime, the reconfiguration cannot affect systems availability. Adapting to context changes without harming the functioning of the system or user activities is a major goal to achieve interoperability in context-aware systems and ubiquitous computing (EDWARDS; GRINTER, 2001; SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007).
- **Heterogeneity:** current systems have varying transport capability, quality and usage cost with different requirements, input/output methods and several interaction mechanisms such as speech, movement, screens, buttons, etc. This diversity of software structures, component models, interface technologies and languages is present not only in computers but a wide range of components. To turn spontaneous interoperability a reality, the heterogeneity aspect of the devices should be considered (NIEMELÄ; LATVAKOSKI, 2004; ATHANASOPOULOS; TSALGATIDOU; PANTAZOGLOU, 2008).
- **Evaluation:** it is necessary to refine interoperability measurement towards a complete set of interoperability models. Besides, it is also important to bring more quantitative measures of interoperability as part of the evolution for truly interoperable systems (FORD *et al*., 2007).

2.5 Chapter Considerations

This chapter presents the basic concepts on ubiquitous computing, context awareness, and interoperability that permeate this work. Each one of them with a general definition, enabling a shared understanding of the topic under discussion; also their characteristics representing attributed and properties by which the concepts can be described and observed.

The examples and challenges acted as motivation to this research and in the next chapter, a *quasi*-systematic literature review is presented retrieve from technical literature the current research state of the field as well as the progress and outcomes related to the concepts presented here.

This initial investigation was a significant activity to further development of the work as it was possible to understand the background of the area and how this work relates to previous ones.

3 Interoperability Concepts and Attributes in Context-Aware Systems: A Literature Review

This chapter presents the process, and initial results of the quasi-Systematic Literature Review conducted to characterize interoperability in Context-Aware Systems. The findings expose the current state of the research in literature and were used to support the development of this work.

3.1 Introduction

The *quasi*-Systematic Literature Review reported in this chapter was conducted to investigate interoperability in context-aware systems. The purpose is to characterize this research area and organize a body of knowledge for the further development of this work.

A systematic literature review is a type of secondary study (KITCHENHAM, 2004) performed focusing on a distinct research topic aiming to identify all relevant results that meet the research purpose. According to this purpose, the material retrieved should be evaluated and interpreted. This methodology differs from other reviews of literature because a formal and well-defined process has to be followed, which contributes to the findings reliability. With the organization, it is also possible to replicate the review to identify new results in later periods and audit the used protocol. These actions contribute to the reasons to perform a systematic literature review, together with:

- Summarize the publications related to interoperability in context-aware systems.
- Identify possible research gaps in current publications and observe which areas need further investigation.
- Expand the conceptual background of this investigation in an unbiased and replicable manner.

It is an initial review focused mainly on the characterization of this research topic with no comparison performed; therefore, we identify it as a *quasi*-Systematic Literature Review (*q*SLR) (TRAVASSOS *et al*., 2008). We executed all the phases of the *q*SLR with the same rigor as in a systematic literature review but with no means to compare outcomes or apply meta-analysis in principle.

The following sections of this chapter expose segments of the used protocol as well the data retrieved and the review contribution. The complete protocol of this review was documented as a technical report (MOTTA *et al*., 2015). The author of this work was responsible for organizing and conducting each step of the review. The others researchers acted as quality elements, to mitigate bias and reduce the validity threats.

A review is an important tool for this research, but it means to contribute beyond this work. It was an activity of the CAcTUS project that allowed further project progress and having similarities with other two reviews conducted in the same context (RODRIGUES; MATALONGA; TRAVASSOS, 2014; SANTOS; ROCHA; ANDRADE, 2015). It can also provide an overview of the current results in this area for other researchers and as a basis for future work.

3.2 Review Process

In a systematic review, a structured process should be followed (BIOLCHINI *et al*., 2007) divided into four different steps: planning, execution, result analysis and packaging. Figure 3.1 presents the review process with these steps. Figure 3.2 detail the activities performed in each step.

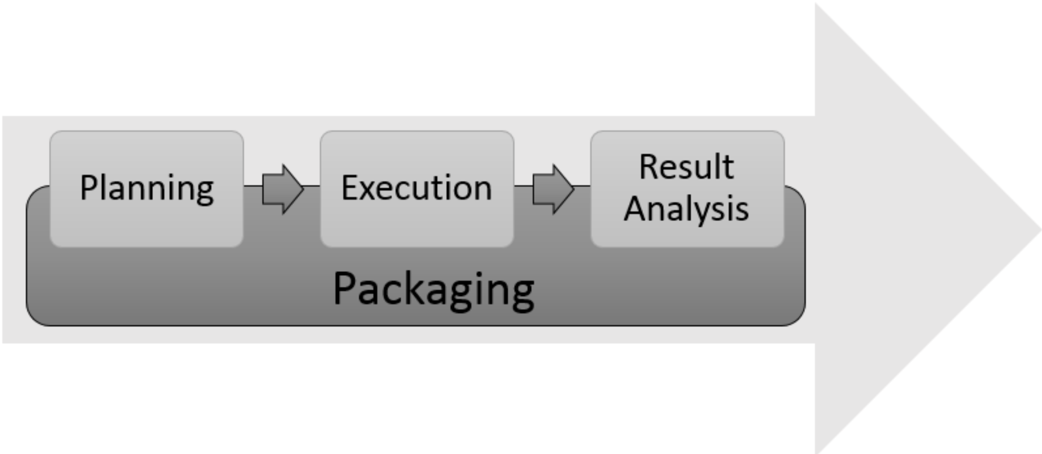

Figure 3.1 - Review process with four steps, adapted from (BIOLCHINI *et al*., 2007).

Planning: prepare the protocol based on the research questions. The search string should be formulated considering possible terms and synonyms. Studies selection and inclusion criteria are also decided. It is important the protocol approval to proceed to the execution step.

Execution: this step is carried in trials where the search string iteratively evolves aiming to improve precision and recall. Each trial involves reading and consensus from the readers' part on the studies retrieved. Decisions encompass whether to continue and include papers considering the criteria established in the planning step, or refine search string and perform a new trial. It is important the readers consensus proceed to the analysis step.

Result Analysis: the readers agree upon a set of candidate papers, considering the inclusion/exclusion criteria. After full reading the candidate papers, their data is extracted and passed through quality assessment. Based on the results the readers agree on the list of included papers with all the information synthesized, therefore enabling analyses focused on answering the research questions and review completion.

Packaging: this step is performed through all the review process, aiming to document every decision in each activity, as well as the information collected and analyzed.

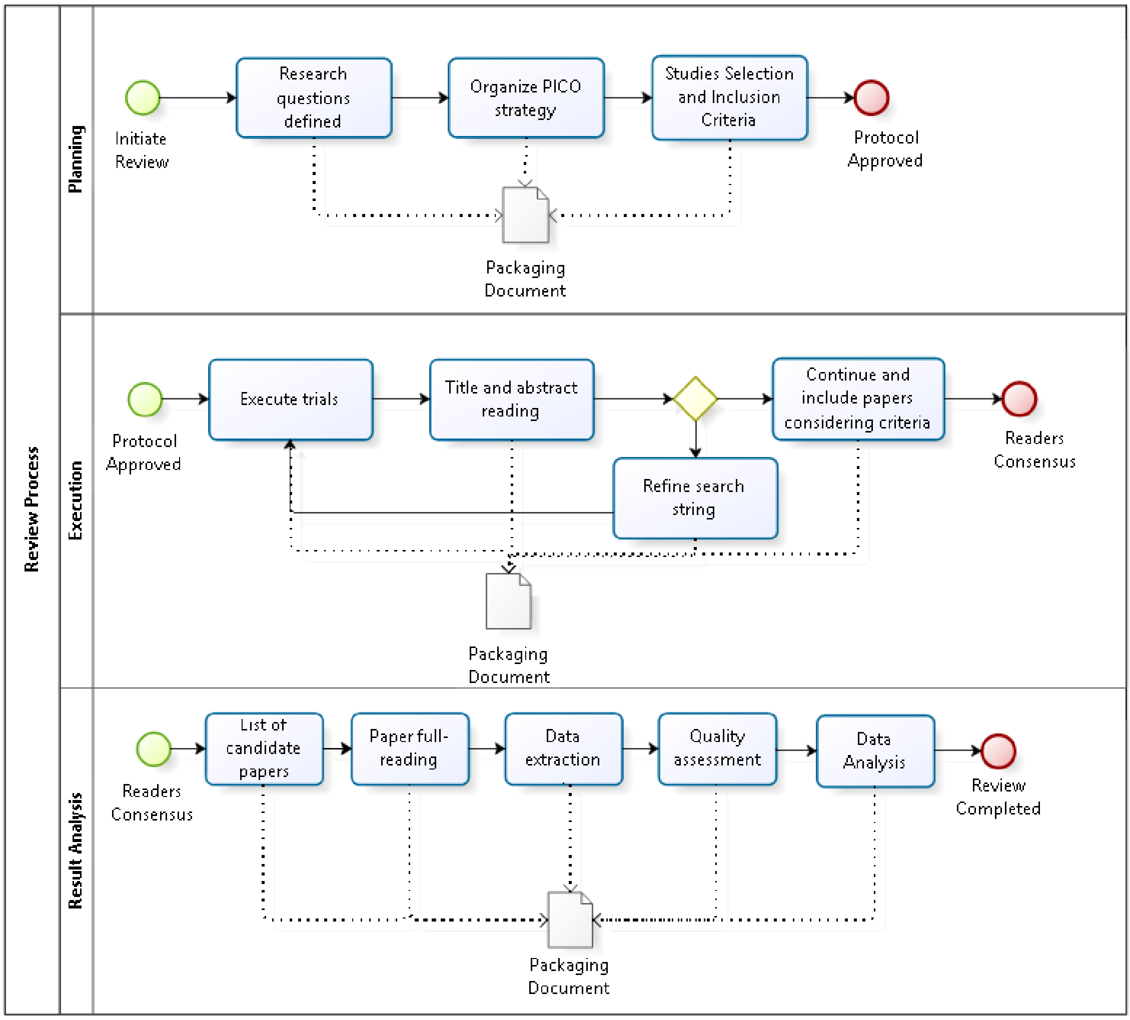

Figure 3.2 - The extension of the process as it was carried out in this review

3.3 Objectives

As presented in the Introduction, with widespread technology and interconnection of different systems and devices, interoperability has become critical. In context-aware systems, it is essential that multiple types of systems, devices, and services be able to interact with each other. Therefore, interoperability is a crucial requirement for this kind of system (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007). The need for this review was commissioned inside the CAcTUS project and concerns to observe interoperability in context-aware systems as a preliminary step to further project activities. The GQM paradigm (BASILI, 1992) was used to derive the goals of the *q*SLR.

Table 3.1 - Goals from the GQM paradigm, adapted from (BASILI 1992)

Goal	
Analyze	*Interoperability*
With the purpose of	*Characterizing*
Regarding	*The concepts and attributes that define it*
From the point of view of	*Researchers*
In the context of	*Context-aware software systems described in technical literature*

3.4 Planning

This section presents in details the planning activity of the *qSLR* conducted. Before the review execution, a search was performed looking for other reviews dealing with interoperability in context-aware systems. The method used was the title and abstract reading, searching the term "interoperability". Other terms searched were: **systematic review, literature review, systematic literature review, structured literature review, structured review** and **systematic mapping**. From the results, one work presents a review of context-aware systems and suggests a new classification framework of context-aware systems (HONG; SUH; KIM, 2009). Another interesting review is focused on interoperability evaluation models, that for its relevance was included in this work (REZAEI *et al*., 2014). After the initial research, this review attempts to fill the gap and merge these two research areas: interoperability and context-aware systems.

The planning step encompasses three different activities: define research questions, organize string using the PICO strategy and decide the selection criteria. It is necessary that the initial version of the protocol be established and approved. These activities are presented in Figure 3.3 and detailed in the following.

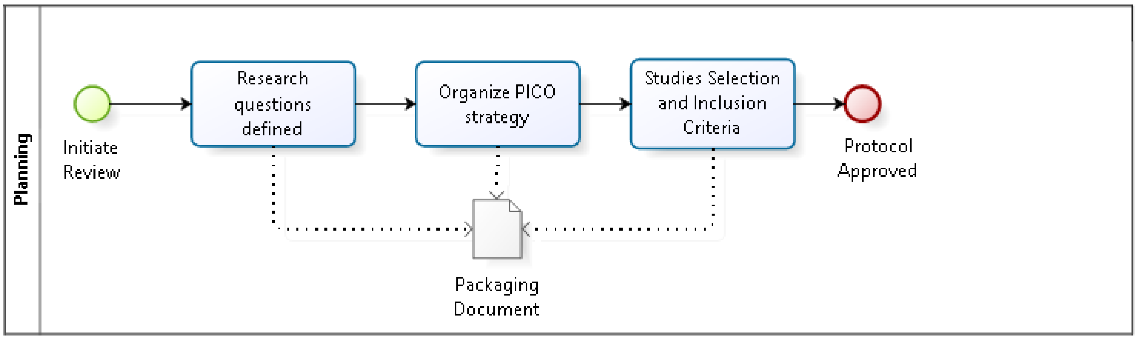

Figure 3.3. Review Process: Planning Activities

3.4.1 Research Questions

After observing the need for a systematic review, the next step is to formulate research questions (KITCHENHAM, 2004). Using the GQM paradigm, from the goal presented above, we derived research questions and metrics that answers the proposed questions.

Table 3.2 - Questions from the GQM paradigm, adapted from (BASILI 1992)

Goal	
Analyze	*Interoperability*
With the purpose of	*Characterizing*
Regarding	*The concepts and attributes that define it*
From the point of view of	*Researchers*
In the context of	*Context-aware software systems described in technical literature*
Questions	
How is interoperability addressed in context-aware software systems?	
Which are interoperability concepts and attributes used in context-aware software systems?	
How to evaluate interoperability in context-aware software systems?	

3.4.2 PICO Strategy

The **PICO** strategy was used to organize and structure the search string (PAI *et al*., 2004). This acronym organizes the search into four parts: **P**opulation, **I**ntervention, **C**omparison (*if applicable*) and **O**utcome.

- **Population**: context-aware systems.
- **Intervention**: interoperability.
- **Comparison**: *not applicable.*
- **Outcome**: definition, attributes, and evaluation methods.

3.4.3 Studies Selection and Inclusion Criteria

This step regards the organization and criteria related to the search of primary studies. These studies are articles, written in English language and available on the web, enabling their retrieval. The database and search engine used was Scopus Database[3]. It was chosen because it covers the majority of software engineering publications, presents stable results and was used in previous reviews, as (RODRIGUES; MATALONGA; TRAVASSOS, 2014) and (SANTOS; ROCHA; ANDRADE, 2015). Terms limitation was a decisive factor for not choosing IEEE Xplore Digital Library[4].

[3] http://www.scopus.com/

[4] http://ieeexplore.ieee.org/Xplore/home.jsp

For this selection, a set of inclusion and exclusion criteria was established to support the decision. Considering the nature of the research question and the characterizing goal of the review the criteria were set to retrieve as many relevant papers as possible. The criteria are presented below:

- **Inclusion Criteria:**
  - To talk about interoperability;
  - To discuss systems including context-awareness characteristics or to be applied to context-aware systems
- **Exclusion Criteria:**
  - Not being available for retrieval;
  - Published earlier than 1991[5];
  - Studies in duplicity;
  - Register of proceedings.

3.5 Execution

The iterative nature of the systematic review process can be observed in the execution steps because the search is performed using various combinations of search terms (different trials) derived from the research question, supported by the PICO strategy.

In every trial, the decision process and their respective partial results are fully detailed in the protocol (MOTTA *et al*., 2015) and partially described in Appendix A. The execution step encompass different activities: execute trial, read title and abstracts of the returned articles, refine string if necessary and include papers considering the criteria. There is a need to reach the reviewers consensus to proceed. These activities are presented in Figure 3.4 and detailed in the following.

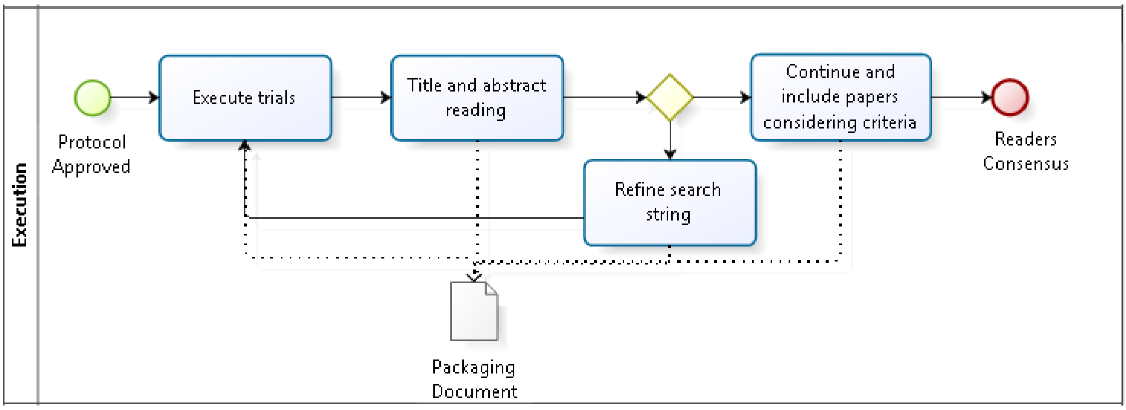

Figure 3.4. Review Process: Execution Activities

[5] Considering the seminal work of (WEISER, 1991)

The search string has evolved until the last trial and as presented:

- **Population**: context-aware systems.
    - (("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR pervasive OR ubiquitous OR "event based" OR "self adaptive" OR "self adapt" OR "ambient intelligence" OR "assisted living" OR "agents systems" OR "multiagent systems" OR "systems of systems" OR “software systems”)
- **Intervention**: interoperability.
    - (interoperability OR interconnection OR interoperation OR interaction OR integration OR exchange)
- **Outcome**: definition, attributes, and evaluation methods.
    - ("evaluation metric" OR "evaluation method" OR "evaluation model" OR "evaluation process" OR "evaluation methodology" OR "evaluation criteria" OR "evaluation approach" OR "evaluation strategy" OR "measurement method" OR "measurement model" OR "measurement process" OR "assessment method" OR "assessment model" OR "assessment strategy" OR "quality attributes" OR "quality properties" OR "quality characteristics" OR "quality features"))

After the search, the results pass through selection. The selection in the first step considers title, abstract and keywords of an article to be accepted and selected.

- **Acceptance Criteria:**
    - Four distinct readers evaluated each study. The readers were the author of this work, the advisor of this work and two external readers with experience in conducting systematic reviews and involved with context-aware research. The discussion setting includes the readers expressing and justifying the rationale behind the choice and reaching a consensus to include or exclude the study under consideration. The studies acceptance criteria organized as follows:

Table 3.3 - Acceptance criteria among readers

Reader 1	Reader 2	Reader 3	Reader 4	Final
Include	*Include*	*Include*	*Include*	Include
Include	*Include*	*Include*	*Doubt*	Include
Include	*Include*	*Doubt*	*Doubt*	Include
Include	*Doubt*	*Doubt*	*Doubt*	Discussion
Doubt	*Doubt*	*Doubt*	*Doubt*	Discussion
Doubt	*Doubt*	*Doubt*	*Exclude*	Discussion
Doubt	*Doubt*	*Exclude*	*Exclude*	Exclude
Doubt	*Exclude*	*Exclude*	*Exclude*	Exclude
Exclude	*Exclude*	*Exclude*	*Exclude*	Exclude
Exclude	*Exclude*	*Exclude*	*Include*	Exclude
Exclude	*Exclude*	*Include*	*Include*	Discussion
Exclude	*Include*	*Include*	*Include*	Include
Doubt	*Doubt*	*Include*	*Exclude*	Discussion
Include	*Include*	*Doubt*	*Exclude*	Discussion
Exclude	*Exclude*	*Doubt*	*Include*	Discussion

After reaching a consensus, all the selected papers go to analysis in the next step.

3.6 Result Analysis

The final trial was executed on July 14th of 2015. From this execution, 408 articles were recovered. These articles had to pass the inclusion, exclusion, and acceptance criteria. As result 16 articles remained for a full reading, they were considered relevant for the review objective as they passed all criterion. During the result analysis a series of activities are performed: select the candidate papers and full-read them, extract data according to the extraction form, assessing the quality according to the criteria in the quality assessment and analyze data. These activities are presented in Figure 3.5 and detailed in the following.

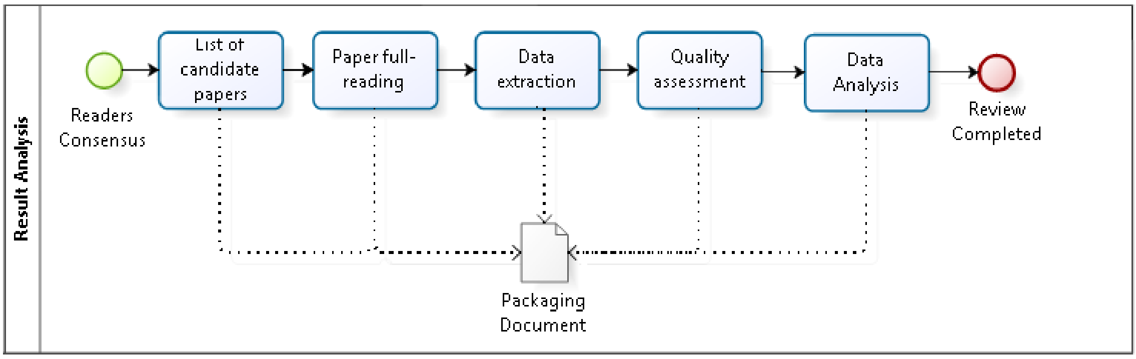

Figure 3.5. Review Process: Result Analysis Activities

From the candidate articles, the snowballing process was performed in two forms: backward and forward.

- **Backward**: one of the readers was responsible for checking the references presented in the articles, and if the title seemed appropriate for the review, it was considered.
- **Forward**: Using Scopus database, one of the readers searched if each article was cited by any other. If so, by reading the title and it appears suitable for the review, this new paper was considered. The possible new set of papers was then revised and approved by another reader. With this activity, nine new articles were included and enhanced the coverage of the review. Afterward, a total of 25 papers were selected. Table 3.4 presents the amount of articles in the last trial.

Table 3.4 - Amount of articles in the last trial

Papers	Total
Found	*408*
Duplicity	*6*
Proceedings	*42*
Kept after full reading	*16*
Snowballing	*9*
Final	**25**

3.6.1 Data Extraction

The extraction stage aims to capture information that can answer the research questions. A form was used to aid the process and collect the information of interest described, and it is presented in Table 3.5. All readers integrally read the papers after selection. The author of this work was responsible for extracting the information according to the proposed form. The data was then revised by other two readers that were free to add or remove any information. This step contributed to mutual understanding between the readers and to have a stronger belief in the selected papers as they fulfilled the required fields.

Table 3.5 - Data Extraction Form

Extraction Field - *Description*
Reference information - *Title, authors, year, source, abstract.*
Interoperability definition - *As presented in the article.*
Interoperability dimension - *The range variation of interoperability (e.g. Semantic, Syntactic) as presented in the article.*
Contextualized Interoperability - *Domain where interoperability is considered (e.g. Business Interoperability, Service Interoperability) as presented in the article.*
Interoperability attribute and theoretical basis - *Name or list of interoperability attributes presented in the paper and how it was derived.*
Evaluated interoperability attribute - *Name or list of interoperability attributes evaluated in the presented technique/article.*
Interoperability measures or method - *Type and value of the measurement system used to evaluate the presented interoperability attributes, or the model or method used to evaluate interoperability.*
Pre-existent approach - *If the article presents an adaption or evolution of an existent approach.*
Conditions or Restrictions - *Limitations for the applicability.*
Target Domain – *The main setting of software category, as presented in the article.*
Type of experimental study - *Types of empirical studies according to ESE group terminology[6].*
Experimental Study Data - *Include any information regarding design, variables and threats if available. Also, which interoperability evaluation method and attributes were used.*

3.6.2 Quality assessment

The data extracted was also submitted to a quality evaluation to weight the importance of individual studies and better detail each study. It is important to state that quality in this assessment is related to adequacy, or how much a paper attend the objectives of this research. For that, we defined questions based on the extraction form (Table 3.5) generating a set of quality assessment criteria. By answering the questions presented it is possible to observe each study according to the relevance for the characterization of interoperability in context-aware systems.

[6] http://lens-ese.cos.ufrj.br/wikiese/index.php/Experimental_Software_Engineering_-_Glossary_of_Terms

Table 3.6 - Quality Assessment Criteria

Quality Assessment Criteria	
Criteria related to interoperability concepts	*Is there any interoperability definition? (1 pt.)*
	Is there any description of the interoperability dimension? (1 pt.)
	Is there any description about the interoperability application? (1 pt.)
	Is there any definition regarding interoperability attributes (e.g. security, reliability)? (0.5 pt.)
	Is there any description about how the interoperability attributes have been derived? (0.5 pt.)
Criteria related to interoperability evaluation	*Is the interoperability evaluation described? (1 pt.)*
	Does the interoperability evaluation include the proposed attributes? (1 pt.)
	Is there an empirical evaluation of the interoperability approach? (1 pt.)
Criteria related to background theory or applicability	*Does the paper describe any adaptation/evolution of pre-existent interoperability approach? (1 pt.)*
	Is there any description of restrictions and conditions about the applicability of the interoperability approach? (1 pt.)
	Is it possible to identify for which types of system can the interoperability approach be used? (1 pt.)
Criteria related to interoperability approach generalization	*Is there any description of the interoperability approach application in additional settings? (0.5 pt. for each setting)*

The author of this work conducted the assessment that was later revised by another researcher. If any doubts or changes arises, both discussed to reach a consensus. It happened in an iterative manner until evaluating all papers.

After this evaluation, as a cut-off point was used the first quartile, calculated in 3 points. It was chosen to keep only papers with a score above the specified cut-point, strengthening the analysis as only the most relevant works were kept. With this decision, from the twenty-five initial selected papers, only seventeen were included. A summary of these seventeen articles is presented below.

1. Athanasopoulos, Tsalgatidou, and Pantazoglou 2008;
2. Chen 2005;
3. Chen, Doumeingts, and Vernadat 2008;
4. Fang, Hu, and Han 2004;
5. Gjoreski *et al*. 2015;
6. Kutvonen, Ruokolainen, and Metso 2007;
7. Kutvonen 2013;
8. Madni and Sievers 2013;

9. Memon *et al*. 2014;
10. Naudet *et al*. 2010;
11. Rezaei *et al*. 2014;
12. Ruokolainen and Kutvonen 2006;
13. Sullivan and Lewis 2003;
14. Ullberg, Lagerström, and Johnson 2008;
15. Vega-Barbas *et al*. 2012;
16. Wyatt, Griendling, and Mavris 2012;
17. Zutshi, Grilo, and Jardim-Goncalves 2012.

3.6.3 Studies Summary

Table 3.7 - Studies Summary

Summary of Studies in the Final Result Set	
Study	**Summary**
Athanasopoulos, Tsalgatidou, and Pantazoglou 2008	*It presents a generic service model focused on peer-to-peer services. Context is referred as conformance between what is provided and requested by the services and is considered an interoperability dimension. For the authors, a unified approach for interoperability regarding models, languages and tools can be achieved based on shared set of conceptual principles. A case study in the area of Crisis Management demonstrates the use of the model.*
Chen 2006	*This work proposes a framework to define the research domain of enterprise interoperability, where it relates the enterprise level with interoperability barriers and approaches to remove the barriers. No formal evaluation of the framework was performed.*
Doumeingts, and Vernadat 2008	*This study is a normative writing that reviews architectural developments for integration and interoperability since the beginning of the 1980s. They define and differentiate each concept, also presenting different frameworks that address these concepts, considering the application context. As the work is an overview of previous initiatives, no formal evaluation is performed.*
Fang, Hu, and Han 2004	*The paper analyzes different levels of interoperability in services interaction. It aims to assess interoperability using a formal model, but the model was not validated. In this work, context is any*

	information acquired by a sensor and considered an interoperability level.
Gjoreski *et al*. 2015	*It describes an initiative to evaluate ambient assisted living solutions focused in activity recognition, aiming to understand the user's situation and context. In a laboratory experiment, the solutions were assessed considering the following criteria: recognition accuracy, recognition delay, installation complexity, user acceptance, and interoperability. Eight different solutions were evaluated using a questionnaire.*
Kutvonen, Ruokolainen, and Metso 2007	*The middleware proposed in this work is designed focusing on collaboration establishment and management at business services in an organizational context. The approach is federated where services are independently developed, and the middleware ensures technical, semantic, and pragmatic interoperability. The idea is to seek out partners that can interoperate, using a network business model expressing the desired interaction. An evaluation of the approach is not presented.*
Kutvonen 2013	*The paper is a project that aims to develop a mature and dynamic ecosystem architecture to prevent interoperability mistakes and support collaboration between enterprise and services. As contributions, it addresses significant problems in inter-enterprise computing and proposes that services supported by software artifacts should be composed together, aware of context and preferences.*
Madni and Sievers 2013	*Inserted in systems of systems domain, this paper present considerations and challenges for these systems relations. A typology of systems of systems and an ontology of its integration is shown. The authors present two ways to achieve interoperability in these systems: design focusing on interoperability during systems conception or retrofitted in the systems after its completion, each strategy with pros and cons. The context is presented as characteristic of this type of systems, considering its evolutionary development and emergent behavior.*
Memon *et al*. 2014	*The authors report that the field of Ambient Assisted Living systems should be adaptive, requiring high quality-of-service to achieve interoperability, usability, security, and accuracy. This review*

	presents an overview of related concepts, current solutions and usage of them. Interoperability is considered a vital attribute, but they conclude that despite recent progress challenges as closed solutions and proprietary protocols are yet to be overcome.
Naudet *et al*. 2010	*An ontology of interoperability based on systemic decision model is the main contribution of this paper. A use case illustrates the appropriateness of the concepts. In this work, context is any particular environment or conditions where interoperability problems or solutions may arise, to which they cover in their proposal.*
Rezaei *et al*. 2014	*This review focuses on interoperability evaluation models. They presented ten different models and compared them regarding twelve distinct issues in four different granularity levels. From the main findings, we can observe that based on their analysis, all evaluation models cover the technical interoperability, but syntactic and semantic interoperability issues deserve greater priority and effort.*
Ruokolainen and Kutvonen 2006	*In the context of enterprise computing environments, the authors identified aspects and analyzed properties to achieving interoperability. Interoperability was analyzed using five different levels of abstraction, containing several issues to consider. The solution proposed is federated collaboration, providing support for heterogeneity via loose coupling of business applications and contract based operations. No formal evaluation is presented.*
Sullivan and Lewis 2003	*The work start presenting challenges to obtain* (WEISER, 1991) *vision of ubiquitous systems, such as new processors, sensors, and actuators involving a number and type of organizations that need to take part in achieving the seamless interoperability necessary. The proposed solution is to bridge dynamically two parties from different pervasive environments developed using different ontologies. The paper focused on semantic interoperability, and a prototype was presented to explore the proposed solution.*
Ullberg, Lagerström, and Johnson 2008	*The paper presents a framework to support the creation of architectures to service interoperability analysis and displayed as an influence diagram. They focus on service interoperability, divided at runtime and design time, each one with different attributes. The*

	framework was used as decision support for a large power distribution company in Sweden, as an example.
Vega-Barbas *et al*. 2012	*The paper presents S³OiA, a syntactic/semantic Service-Oriented Architecture that allows the integration of components considering several challenges in the Internet of Things scenarios, such as the dynamic and adaptive nature of the applications since they can evolve depending on the context. The validation process was performed in two parts: a survey to observe the elderly people real needs for their daily routine, and a prototype called Smart Diary, developed using S³OiA architecture concerning the survey findings.*
Wyatt, Griendling, and Mavris 2012	*The authors evaluated eight existent interoperability methods regarding: if they are qualitative or quantitative, the system type the model focus if they include non-materiel solutions and how to execute the measurement. The work is part of the military context, in the scope of acquisition of systems-of-systems. The evaluation results are to support decision-making during the early phases of the acquisition process.*
Zutshi, Grilo and Jardim-Goncalves 2012	*This paper proposes a model to measure interoperability in the business context. In the model, eight parameters represent different interaction levels that entities can engage. Its sub-parameters assess each parameter in a questionnaire form regarding its relevance and performance. Due to difficult to quantify some of the parameters the authors used a multi-criteria (Analytical Network Process) method as a way to evaluate the perception and opinions about parameters. Two case studies applied the model, with the results illustrating its usefulness and potential.*

3.6.4 Initial Results

It is possible to see in Figure 3.6 the studies information regarding citation type and publication year. Despite the seminal work on ubiquitous computing being published over twenty years ago (WEISER, 1991), from the perspective of this review, the publications related to context-awareness feature are more recent. This information may indicate the current interest and progress in the area in the past few years. This perception corroborates with Adam Greenfield (2010), reporting that with the internet, mobile technologies and smart devices popularization we are now living in the dawning age of ubiquitous computing.

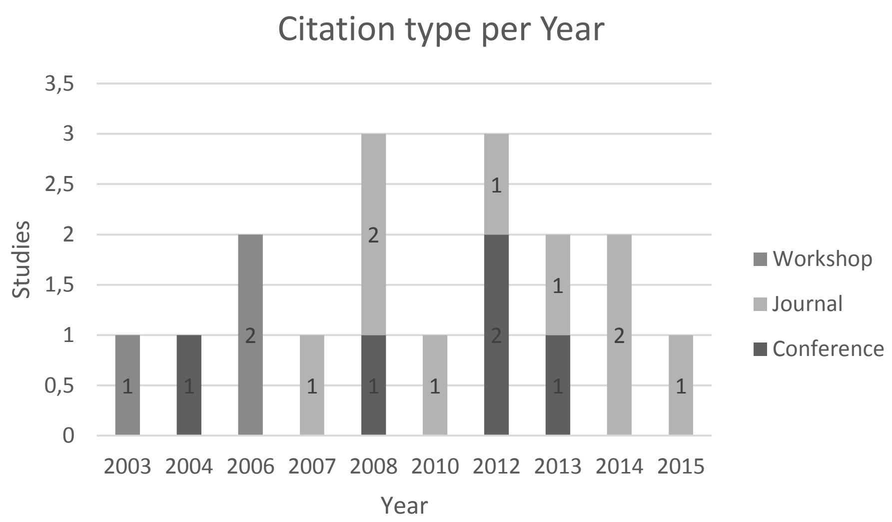

Figure 3.6 - Citation type per Year

Figure 3.7 shows that context–aware systems permeate different domains. Under system domain, we classified works related to ubiquitous systems, also systems-of-systems or any other systems type. Classified in the ubiquitous system domain are several studies (SULLIVAN; LEWIS, 2003; VEGA-BARBAS *et al*., 2012; MEMON *et al*., 2014; GJORESKI *et al*., 2015), a varied set representing the different concerns and opportunities in ubiquitous systems research. Systems-of-systems are often concerned with challenges of interoperable, distributed, adaptable and heterogeneous systems, like the challenges presented in (WYATT; GRIENDLING; MAVRIS, 2012; MADNI; SIEVERS, 2013). The government, business, and enterprises have their interests in interoperability, which supported by computing solutions can meet organizations needs for cooperation, classified under the organizational domain. Some examples are studies like (CHEN, 2006; NAUDET *et al*., 2010; ZUTSHI; GRILO; JARDIM-GONCALVES, 2012). We classified these works as services since their proposals are not restricted as they can be used in different heterogeneous or distributed environments considering services as a mean for systems interaction (FANG; HU; HAN, 2004; ATHANASOPOULOS; TSALGATIDOU; PANTAZOGLOU, 2008; ULLBERG; LAGERSTRÖM; JOHNSON, 2008).

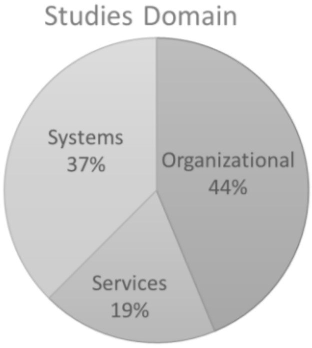

Figure 3.7 - Studies Domain

Based on the data extracted, we applied the quality assessment criteria (Table 3.6) and in Figure 3.8 it is possible to observe the comparative results among the evaluated studies. As reported in sub-section 3.6.2, the cut-off point was established in 3 points.

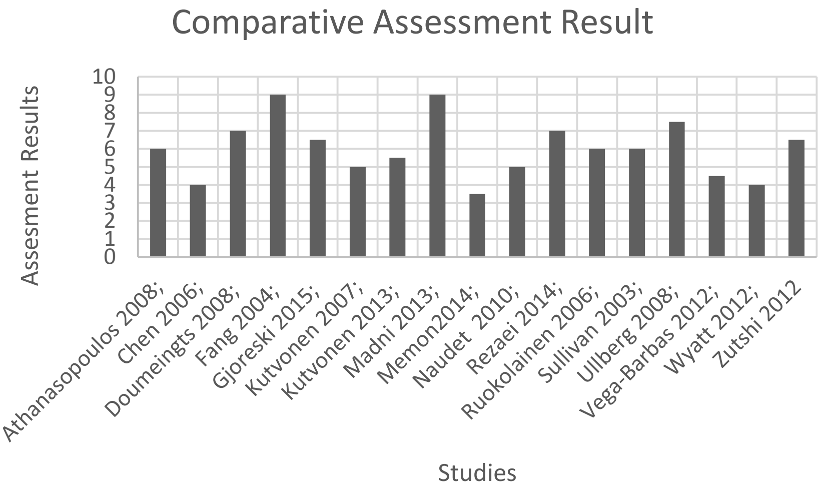

Figure 3.8 - Comparative Assessment Result

In Figure 3.9 we can see that none of the retrieved studies answered all the assessment questions and therefore did not acquire total points. All studies provided information regarding the target domain and approach application. Considering the general results of the evaluation (Figure 3.9), we can see that most of the studies (70.5%) present a definition of interoperability and its dimensions. It is interesting to observe that 76.4% of the studies present interoperability attributes but only 41.1% report some attributes evaluation. Furthermore, only 11.7% present an empirical evaluation of their proposal, from which we understand as lack of evidence in the findings, from the experimental perspective. The aim is to observe the adequacy and contribution of the study to answer our research questions.

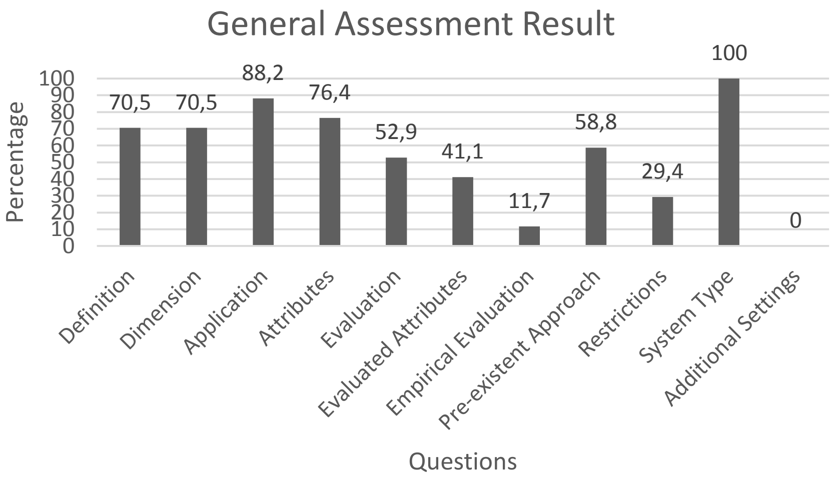

Figure 3.9 - General Assessment Result

This initial analysis was performed to provide a general overview of the data retrieved. To draw a better view of the findings to answer the proposed research questions, a deeper interpretation of this information is required. Next chapter describes this broader analysis and synthesis with qualitative analysis based on the data extracted (Table 3.5).

3.7 Threats to Validity

At Planning: A well-defined protocol and process can lead to consistency studies selection, and minimize research bias when performed by different researchers. The protocol evolved until the final trial and then revised by two external researchers aside the four participants of the review. A recurrent issue in systematic literature reviews regards inconsistent terminology, and the keywords used being restrictive. To tackle this, first, we searched for other reviews and observed the terms used to compose our search string. Also, the protocol and structures have similarities with others reviews already revised and published. It works to reduce researcher bias, but with the inclusion and exclusion criteria imposing selection limits, it is a possibility to miss some studies.

At Execution: Despite several trials and string adjustments, the use of only one research database can lead to missing studies as well. As a way to mitigate that, snowballing was performed. All phases of this review were peer-revised, and that was evident during studies selection where four distinct readers were involved. Any doubt was discussed among the readers, to reduce selection bias, following the criteria presented in Table 3.3.

At Result Analysis: A concern on every systematic review regards extraction and interpretation of data. The extraction form has evolved during the execution, it was revised by the authors and also by external researchers. Bias can be introduced by the researcher when analyzing each paper; this was tackled by having a second researcher to revisit all analysis.

3.8 Chapter Considerations

The process of the review undertaken is described in this chapter. It characterizes state-of-the-art of interoperability in context-aware systems. This process allowed the acquirement of experience in executing systematic literature reviews. The execution of trials, results discussions and sharing thoughts with experienced researchers was, however, tiring, a very positive experience.

We considered the need for a deeper analysis of all the information extracted. Therefore a qualitative analysis methodology was used and it is described in the next chapter.

4 Qualitative Analysis: Applying Grounded Theory

This chapter reports principles and usage of Grounded Theory to analyze the data retrieved from the quasi-systematic literature review aiming to inter-link concepts and relate them into higher abstractions synthesizing the results.

4.1 Introduction

Originated in the 1960s in the field of health studies, Grounded Theory (GT) was introduced by Barney Glaser and Anselm Strauss. In their book **The Discovery of Grounded Theory: Strategies for Qualitative Research** (Glaser and Strauss, 1967) they contrast theory generated by deduction and *a priori* assumptions with the theory discovery **arising from** and **grounded in** research data, through constant comparison. One can argue that every research is somehow "grounded" in data, but using this data from systematic research, together with a set of rigorous research procedures to produce a "grounded theory" is a different approach. Although many classify it as a qualitative method, GT is an inductive methodology that can be used with qualitative or quantitative data (GLASER; STRAUSS, 1967). Since the beginning, different thoughts concerning GT have arisen, including disagreement between the authors. Several works in the Software Engineering area using Grounded Theory was developed (CARVER, 2007; CONTE *et al*, 2009; CRUZES, D., DYBÅ, T., 2011) assisting in the choice of this method in this research. The approach developed by Strauss has been most widely used in qualitative research in the Software Engineering area (ADOLPH *et al*, 2008). Therefore we chose to use Strauss and Corbin’s approach in this work, following the procedures presented in their book (STRAUSS; CORBIN, 1998).

In recent years, much progress was made in Empirical Software Engineering research, including the use of techniques and methods employed in other fields, such as social sciences, applied to research in this area. The use of GT is one example of it, as it can be useful in building a body of knowledge about topics of interest (CARVER, 2007). Other works (SEAMAN, 1999; BADREDDIN, 2013) also present the adequacy of studies, like GT applied to Empirical Software Engineering. In our case, GT suits well the characterization studies with broader research questions, such as the ones presented in the *q*SLR, as it aids in the interpretation and clarification of the results found.

The seventeen papers retrieved from the qSLR presented in the previous chapter provided a significant amount of data to be analyzed, like the attributes related to interoperability, for example. The GT methodology comes as a mechanism to understand this data and how they relate to each other considering the application domain and systems features, as context-awareness. The principles and procedures of GT were used to assist us to develop and analyze the concepts related to interoperability in context-aware systems, as presented below:

- **Planning**: like any research, planning has the primary role as it structures the future steps and decisions to take. Initially, it identifies the area of interest and the process to be followed inside the GT paths. From that, it is necessary to formulate the research questions.
- **Data Collection**: with the area of interest and the research questions, in this step is necessary for the researcher to keep an open mind as the data used can be collected in several manners. Initially, placed in social and health studies, GT resorted to interviews where the researchers speak directly with the subjects on a particular research topic, often based on questions in a varying degree of formality. However, any method can be used, like focus groups, observations, artifacts (like images, objects) or texts, and everything can be significant data. Regardless of the method used for data collection, the goal is to select the most relevant constructs which will maximize the discovery of many phenomenon facets as possible (STRAUSS; CORBIN, 1998; DEY, 1999).
- **Coding**: after having enough data, the coding step starts, and it is a crucial step in GT. Here the researcher should work their sensibility to identify significant data and use the **constant comparison analysis method:** through iteration, going back and forth in the codes generated from the data collected constant observing and comparing to find adequacy, conformity, and coherence among the codes. Based on (STRAUSS; CORBIN, 1998) coding process is executed in three steps:
    - **Open coding** is “the process of breaking down, examining, comparing, conceptualizing and categorizing data” (STRAUSS; CORBIN, 1998). The goal is to identify concepts, their proprieties or dimensions, comparing similarities and differences. The task can be accomplished in lower (line by line) or higher (sentences or paragraphs) level. The code can be a single word or a whole sentence; the challenge is to make it self-explanatory and to present a meaningful idea. A group of codes is called a concept, being an abstract representation of something.

    - **Axial coding** is the step where the concepts are organized. The idea is to cluster the concepts in specific "axes", therefore, axial coding (STRAUSS; CORBIN, 1998). Categories rise from concepts and are more abstract. To observe the relation among categories, it is important to see and understand the concepts in new ways, considering condition, context, and consequence that can influence what is being analyzed.
    - **Selective coding** is a step to identify a core category that expresses the central idea of the research. It is concerned to relate other categories in the axial coding and validates the relations, which should be supported in the original data having subcategories linked to the core category.
- **Reporting**: this step has to be kept in mind throughout the entire process. Writing memos, comments, and decision points during the coding phase can enhance the report and further theory development. Being able to narrate the process of abstraction and describing the rationale behind the codes, concepts and categories is the last challenge to a sound analysis.

These are the basics procedures to perform GT according to (STRAUSS; CORBIN, 1998) and were followed in this work with the necessary adjustments. The undertaken steps, the experience applying the method and results generated are reported in this chapter.

4.2 Applying Grounded Theory

For all this research, the main question (**How is interoperability addressed in context-aware software systems?**) drives the investigation towards the understanding of interoperability concept. This analysis was performed to enhance and strengthen the findings of the *q*SLR regarding interoperability concepts, attributes and evaluation methods used in context-aware systems. The data gathered from the literature become the population used in the analysis.

The objective of this analysis is not going to the deepest level of generating theory but as the authors state: knowledge and understanding take many forms (STRAUSS; CORBIN, 1998). In their perspective, a theory is a set of well-developed concepts which together constitutes an integrated framework to explain a phenomenon. Similarly, as the qSLR, the author of this work was responsible to organize and conduct each step of the review. Two researchers that contributed in the qSLR collaborated with the qualitative analysis as well, acting as quality elements, to mitigate bias and reduce the validity threats.

Developing a theory would encompass a complex and long process. However, the procedures proposed in this approach were chosen as the appropriate methodological vehicle for the study as they are flexible enough to describe, elaborate and classify the concepts generating a conceptual framework. The reasons to choose GT are as follows:

- Due to the lack of an integrated approach in the technical literature regarding interoperability in context-aware systems, this analysis emergent from data based on primary studies offers a high potential to fill this gap.
- GT provides reference support for the procedures in original works from the authors and also several other types of research based on this methodology in software engineering and other domains.
- It is adequate to work with a large amount of information, such as the data extracted from a literature review, and to interpret data. Considering that some concepts have different meanings, this methodology is suitable to establish the similarities and differences among them.
- The expectations when planning the review differed from the reality, once the data was extracted for analysis, calling for a step back and no pre-conceived reasoning in mind, as ideally required for GT.
- Already established and credible methodology in several disciplines (e.g. social studies, psychology) and can achieve significant recognition in software engineering research as well.

The QDA Miner Lite tool[7] was used to support the process. It is a free version of a qualitative analysis software for coding, annotating, retrieving and analyzing data in the form of documents and images. One of the key features is the possibility to link each code to the excerpts it is grounded, easing the task of retrieving the codes and observing the relations. Despite the features of this tool, it does not support the axial coding process, being this step conducted manually.

[7] http://provalisresearch.com/products/qualitative-data-analysis-software/

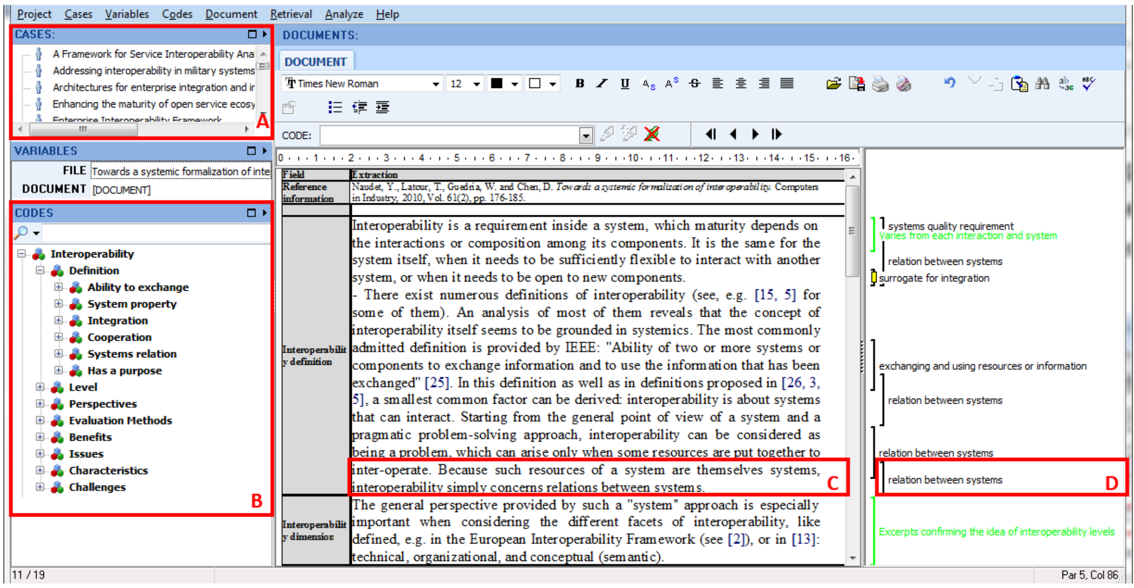

Figure 4.1 - QDA Miner Lite tool

Figure 4.1 presents the tool and four important parts. The box indicated by letter **A** concentrates the **Cases** (i.e., a form containing information extracted from each article) that we treat as the population for the analysis. The box **B**, named **Codes**, contains the codes grouped by categories or subcategories. Box **C** exemplifies an extracted excerpt of one of the studies, and box **D** presents the code assigned to the respective excerpt.

The following subsections presents the application of GT procedures in the light of our research problem, interoperability in context-aware systems, with process and results presented in the format provided by the tool.

4.2.1 Open Coding

In this step, supported by the QDA Miner, the data extraction form from each paper of the *q*SLR was imported into the tool. Following the fields of the form, the coding process started by marking all the excerpts related to an interoperability definition, and then for every data from the extraction form. The excerpts could be a word, a phrase or a full paragraph when relevant for the concept under observation at the moment. To uncover and develop the concepts one has to keep the mind open and also report thoughts and ideas related to it.

Continuing with the coding activity, constant comparative analysis is a regular procedure to execute it. Whenever coming across another excerpt that seemed to talk about the same concept or shared a common attribute, these were grouped together into the same code. This activity requires some abstraction since the code should be written in a form that represents both excerpts. Table 4.1 presents one example of the codes. These are some excerpts from original texts from the articles that compose the code, which in turn is related to interoperability definition.

Table 4.1 - Open coding example

Code	Excerpts
Working together despite differences	*"It refers to make different data models and query languages working together"* (CHEN; DOUMEINGTS; VERNADAT, 2008).
	"Ability of two or more software components to cooperate despite differences in language, interface, and execution platform" (FANG; HU; HAN, 2004).
	"To be able to work together with both different cloud services and providers, and other applications or platforms that are not cloud dependent" (REZAEI *et al*., 2014).

Considering the natural differences between studies, the used (or lack of) taxonomies, diversity of vocabulary and the English as the language for the analysis, we used Merriam-Webster[8], Cambridge[9] and Oxford[10] dictionaries, backing the interpretation process. When analyzing the data, we created our dictionary of frequent terms to consolidate the particular view built during the process.

All the matching from text to code was accompanied by one researcher and then integrally revised by another. Both of them were already conscious of the work since they participated in the *q*SLR as well. The procedure followed was to review each extraction and the respective code in the order they appeared, contributing to the constant comparison. Every match was then classified and justified by the reviewers as:

- **Agreement:** when the code is adequate to represent the excerpt and is consistent in all the segments. Sometimes, based on the comments, better terms to describe the codes were included and other texts parts added in the code.
- **Partial Agreement:** means that the code refers to a part of the text, but to address it completely more text should be included.
- **Disagreement:** in the case of disagreement other code for the text should be suggested.

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

[9] http://dictionary.cambridge.org/

[10] http://www.oxforddictionaries.com/

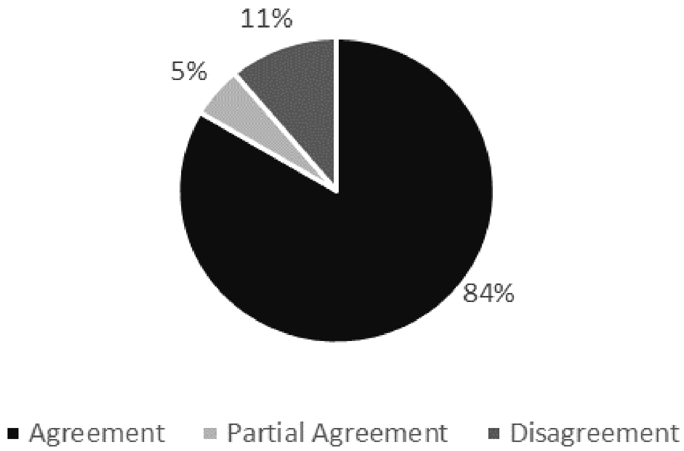

Figure 4.2 - Coding Review

Figure 4.2 shows the review rate among the researchers. In cases of **partial agreement** and **disagreement** meetings and discussions were held among the reviewers to achieve full agreement in the open coding process. This step was conducted with their contribution to decrease potential misinterpretation and bias.

During the coding review, all codes were clustered into categories, and both codes and categories evolved during the open coding process since the analysis is an iterative process. The categories present the concepts of the codes at a higher abstraction level and should naturally rise from the codes. Later grouped into subcategories with is no limit on size, once the concepts are compatible to be clustered together. The use of categories reduces the number of work data the researcher has to deal with, and it is essential for axial coding. Figure 4.3 Categories evolutionFigure 4.3 shows the categories evolution in the QDA Miner Lite tool structure, in which part **A** shows the initial version of the review, and part **B** represents the final organization, where new categories were derived from the original ones. Part **C** exemplifies definition category, with the subcategories organization.

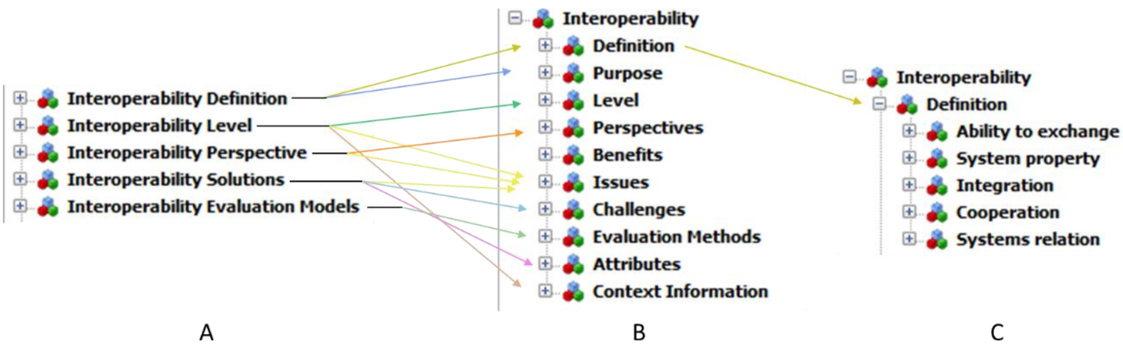

Figure 4.3 Categories evolution

It is important to emphasize that in this step of open coding from the 17 papers retrieved in the literature review: 451 excerpts were coded representing 113 codes grouped into ten categories. Appendix A – Section A presents the code description.

4.2.2 Axial Coding

This section describes the performed process of relating categories to their subcategories. With this step, the fractions of data from the open coding can be reassembled and organized into the categories and subcategories with their descriptions, properties or dimensions.

In Appendix A – Section B a **description** of each category is provided. This description embraces all the codes that belong to the category, to keep the consistency of the segments. It is also presented the **groundedness** of each subcategory, meaning how many excerpts are related to it, and the **density** of each subcategory, meaning how many cases support it. Even the codes with low density (such as only 1 case as evidence) were chosen to be used in order to have more observation. With this information we can observe that there are some subcategories are more frequent than others, leading to stronger concepts due more data to base them. We illustrate the subcategory with one excerpt **example** and the **discussion** regarding the interpretation. The subcategories together with their respective codes are **part of** the whole concept of the category under consideration. The categories, larger circle with strong lines, and subcategories that compose them, smaller circles with thin lines, are presented in this section.

About the interoperability definition, the work of Ford *et al.* (2007) identified 34 distinct definitions for interoperability. It shows that there are still differences in understanding and lack of consensus on the concept, despite the extension of the research associated with the interoperability topic. This study also identified this absence of consensus, since we could observe in the open coding stage that 96 different segments were associated with interoperability definitions. To better observe the definitions found, these codes were grouped into subcategories (axial coding) representing five key concepts that shape the definition of interoperability. These concepts are: the **ability to exchange**, **system´s property**, **integration**, **cooperation** and **system´s relation** (see Figure 4.4). More detail of the subcategories can be seen in Table C.1.1. Finally, the coding process applied allowed us to propose the following interoperability definition, derived from the analyzed data: **Ability of things to interact for a specific purpose, once their differences have been overcome.** This definition also relates to other categories. **To interact,** we understand as relating to the interoperability levels a system can engage interaction. **For a specific purpose** is related to the purpose category, presented next. **Once their differences have been overcome** represents the solutions and decisions to be made to achieve interoperability, described as attributes. So, the definition is not only the representation of its subcategories but also includes other concepts that contribute to the understanding and definition of interoperability.

Figure 4.4 - Category: Definition

Every software system is developed for a particular purpose and often it is necessary for systems to interact to fulfill this purpose. These interacting systems can share the same goal or have independent ones. In this scenario, interoperability enables systems to interact. The goal or objective for the interoperability to happen is motivated by the systems own purpose. This statement may sound obvious, however from the data we observed the purpose could affect how interoperability is addressed.

The data analyzed in open code stage was organized (axial coding) into six subcategories related to purpose, presented in Figure 4.5. **To achieve a purpose** and **to improve people´s quality of life** are subcategories based on excerpts that contribute to supporting the existence of a concept of purpose in the interoperability topic, despite not specifying conditions.

Now, considering the other subcategories: **to develop business, to improve positioning with systems implementation, to improve services provision to citizens and business** and **partnerships more cost efficient** are different between them and more troublesome to achieve. The efforts in development, concerns in decision-making and results will be very different based on the purpose behind them. Table C.1.2 presents more detail of the subcategories.

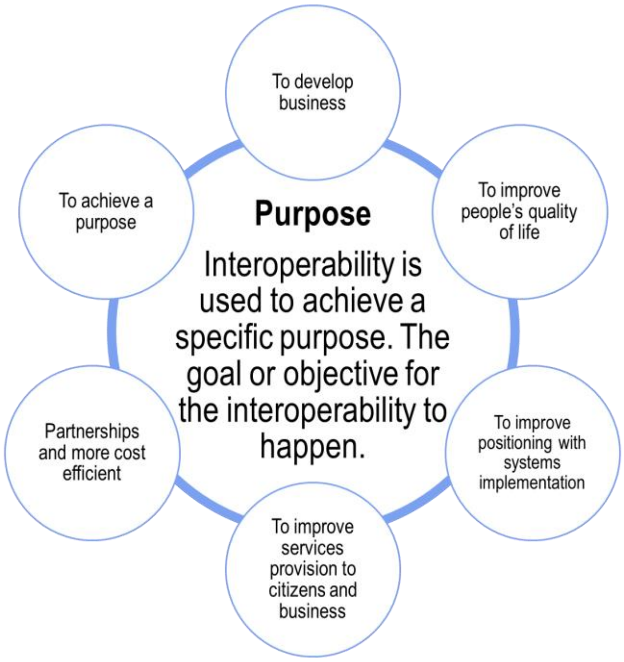

Figure 4.5 - Category: Purpose

From the 17 papers used as the population in this analysis, 14 of them presented the concept we present as **interoperability level**. During open coding we observed that different terms were used (facets, maturity, aspects, degree, category, dimensions, and so on) to present the same idea we interpret as level. We understand this concept as **the level of support provided for interoperability in a given interaction.** To create and maintain interoperable systems considering different system structures and purposes can be more than a connectivity task. The term level was chosen because we understand that it is the most suitable for this category description, as it stands for a position in a scale or rank, in our case, the level of support.

The data analyzed provided enough information to organize them into subcategories and observe four different interoperability levels: **technical**, **syntactic**, **semantic** and **organizational** (Figure 4.6). The **about levels** subcategory comprehend excerpts that only cite or confirm this concept, although not defining or report any other matter. Table C.1.3 presents more detail of the subcategories.

Technical: it is concerned with the connectivity, communication, and operation regarding the interacting entities. To enable interaction, it is important to consider the architecture style, issues regarding authentication and authorization, the use of technical standards, protocols for communication and transport and an interface between the things - when applicable. Our proposition is that this level enables the interaction between things.

Syntactic: this level is concerned with the communication, data exchange, and syntax consistency. To enable this kind of interaction, it is important to consider the use of standards and protocols and provide the suitable infrastructure needed and issues regarding data definition - when applicable. Our proposition, as we understand, is that this level enables the interaction of things through data.

Semantic: this level is concerned with the interpretation and reciprocal understanding between interacting entities. To enable the interaction at this level, it is important to consider, when applicable: the use of semantic web and *a priori* solutions to be decided at design time; the use of standards; models and ontologies; and the sharing of compatible concepts. Our proposition, as we understand, this level enables mutual understanding of the meaning of the data between things.

Organizational: this level is concerned with business rules, policies and constraints, process alignment and the actions necessary to make entities collaborate. To enable interactions at this level, it is important the assets and structures, use compensation or negotiation, use standards, and protocols, share common concepts and process - when applicable. Our proposition, as we understand, this level enables the mutual understanding of data meaning between things, under determined conditions.

With the definition of the levels concept, we can distinguish in what levels systems interoperate. Take for instance the cases where the interaction is an information exchange; communication concerns should support this interaction, and the complexity can increase whether considering additional layers such as a physical link or transport. These **technical** concerns are related to system connectivity, communication and operation. Moreover, they are very different from **organizational** concerned to security and sensitive data.

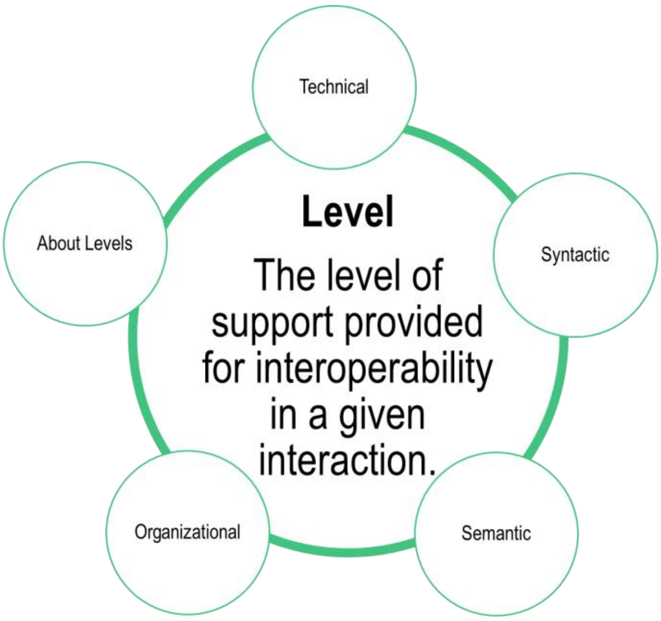

Figure 4.6 - Category: Level

Based on the analyzed data, from what was observed in categories like purpose and definition, there is a clear difference in the conceptual understanding of interoperability. This difference is what we try to embrace in the perspective category. Depending on the perspective, interoperability can change the purpose or needs across the levels. Therefore, this category refers to the interpretation and application of the interoperability concept in different fields, domains and areas regarding the following perspectives: **organization**, **system,** and **service** (Figure 4.7). The **about perspectives** subcategory represents the emphasis given the perspectives associated with interoperability, although not defining or report any other matter. Table C.1.4 presents more detail of the subcategories.

Internally, several discussion sessions were held to organize the data in the presented form. We understand **organization** to cover business, military and health perspectives, based on the data retrieved. However, this does not prevent other perspectives to be included under this concept. The idea here is to consider the goals, the culture, and values of the organization when different computing technologies have to interact to contribute to them.

We also understand that a service can be seen as a system, but we decided to organize into separate perspectives based on the codes and data they are grounded. **System** perspectives refer to physical, technical infrastructure and stakeholders concerns while **Service** is more related to computer-based applications providing a service and the concerns regarding this provision.

Figure 4.7 - Category: Perspectives

The analyzed studies reported some advantages observed in interoperable systems. To better understand them, these were grouped into four subcategories during axial coding stage: **improves understanding**, **increases flexibility**, **hides complexity** and **creates cohesion** (Figure 4.8). The Table C.1.5 presents more detail of these subcategories.

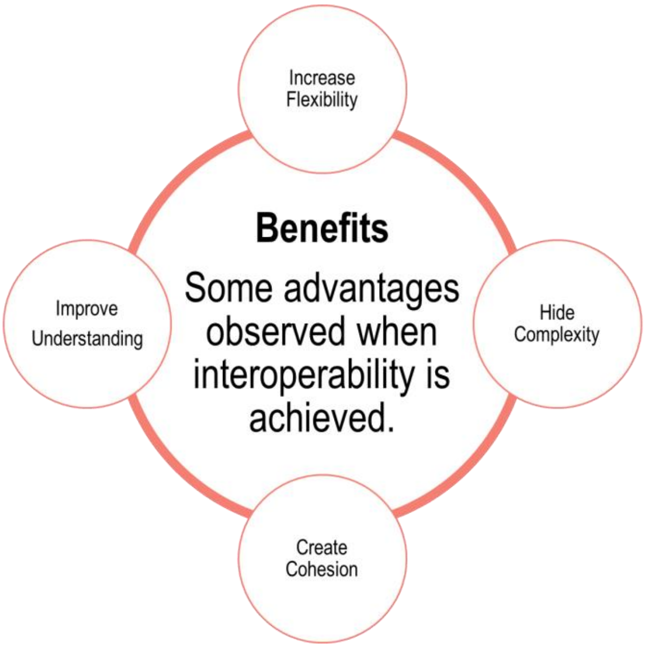

Figure 4.8 - Category: Benefits

When dealing with interoperability some problems and concerns can arise. In the performed analysis under our dataset, we identified 50 different segments that were associated with interoperability issues. The issues were grouped into three subcategories as presented in Figure 4.9. These subcategories represent where the issues can be observed when: **related to levels**, **related to planning** and **related to perspectives**. Table C.1.6 shows details of each of these subcategories.

An example is different currencies, representing domain conflicts that occur when different reference systems are used to measure value. Conflicts like this are examples of **issues related to semantic level**. To develop a system and make it interoperable raise **issues related to planning**, especially regarding costs when interoperability requirement was initially out of scope. System acquisition is a complexity driver due organization´s purchase policies and the need to achieve interoperability among legacy, and new systems are examples of **issues related to perspectives**.

Figure 4.9 - Category: Issues

The studies analyzed present some challenges to be tackled so that interoperability evolves as a field across the several concepts presented so far. We organized them into **research** or **practice challenges** (Figure 4.10). Table C.1.7 presents more details regarding the subcategories.

Research challenges care for the academic development, to improve and focus especially on those fields where compatibility is still low, like the areas lacking or with conflicting standard developments or lacking uniform implementation of standards. **Practice challenges** are related to current interoperability in usage, for example, legacy and new systems interoperability – a recurrent problem in the field.

Figure 4.10 - Category: Challenges

Is it possible to identify the root causes of interoperability problems? Is it possible to quantify interoperability? How do we measure the tradeoff between systems interoperability and other attributes like security or availability? These are some of the questions we are interested in the research regarding evaluation. Some works from our studies set were also interested in questions like these and focused on interoperability measurement (REZAEI *et al*., 2014).

In this context, evaluation methods provide mechanisms to an organization, program or project to assess or aid the decision-making on interoperability. These methods can present the degree of achievement or adequacy of interoperability in a given context. Table C.1.8 presents more details of these methods.

To organize the information related to these methods in subcategories and improve our understanding, we had to check the original references to such methods. Figure 4.11 presents the evaluation methods found: **Athena**, **i-Score**, **IAM** (Interoperability assessment methodology)**, LISI** (Levels of Information Systems Interoperability)**, OIMM** (Organizational Interoperability Maturity Model), **QIM** (Quantification of Interoperability Methodology). The **other methods** subcategory represents 14 methods are cited one time and usually not detailed.

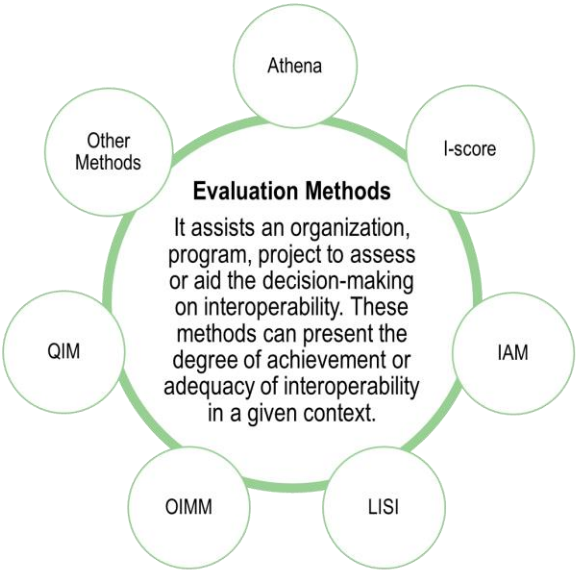

Figure 4.11 - Category: Evaluation Methods

Figure 4.12 presents the subcategories related to attributes: **interoperability mechanisms**, **general attributes**, and their respective codes from the whole concept idea. Table C.1.9 shows more detail of the subcategories.

General Attributes are typical systems' attributes related to interoperability: adaptive behavior, availability, compatibility, conformance with organization requirements, conformance with system’s requirements, dynamic connection and standardization. For example:

- **Adaptive behavior –** the ability to dynamically adapt to the environment the systems is being used within its limitations.
- **Compatibility -** compatibility of a system deals with what ease the system can operate with shared applications. This attribute also covers the system's compatibility on different-different platforms.

Interoperability Mechanisms are the decisions to be taken and mechanisms recurrently used to address these attributes like:

- **Adaptive behavior -** decisions made at runtime, use transformation to enable interaction and others.
- **Compatibility -** use suitable architectures and protocols, define data (data type, data structure, data representation, data format, language) and others.

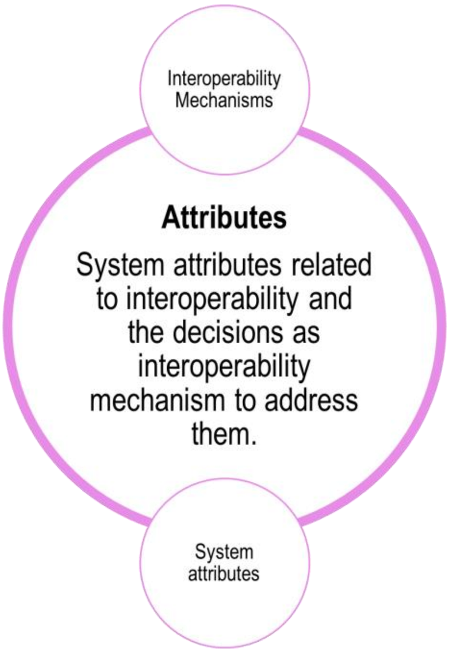

Figure 4.12 - Category: Attributes

This research considers context as a new factor, and our expectation is that context variance influences systems interoperability, adding complexity and overlooked difficulties. Based on the definition provided by Dey (2001) **context** is any information that can be used to characterize the situation of an entity. **Context information** is any data that can be used to characterize the status of a person, place, or object that is considered relevant.

Based on the data analysis, context is presented in the studies as a characterizing factor since in the analysis performed different segments were associated with context (coded). It is not considered an influencing factor once it is not clear the value (e.g. positive or negative) of such influence. These codes were grouped into subcategories (axial coding) representing five context variables identified: **business information**, **extrinsic environment information**, **interaction time**, **intrinsic environment information** and **user knowledge** (Figure 4.13). Table C.1.10 presents more detail of the subcategories.

Regarding the expectations, we observed that each variable is retrieved (or perceived) in a particular manner. Also, each one contributes to the system in a different way.

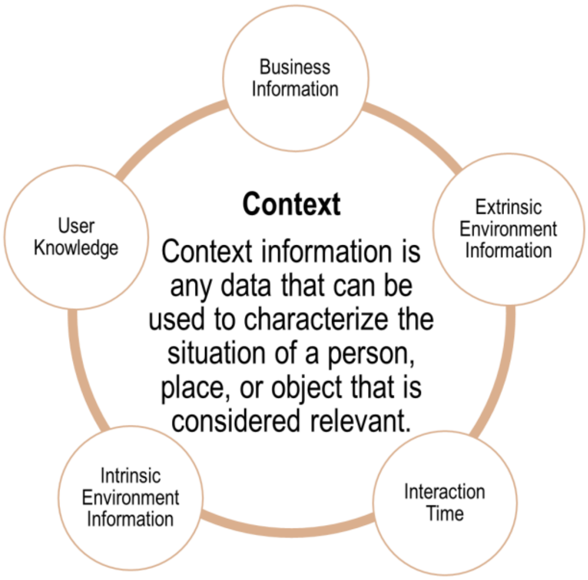

Figure 4.13 - Category: Context

4.2.3 Selective Coding

During open coding (Figure 4.14B) we analyzed the retrieved data (Figure 4.14A) identifying what was relevant, and how the data varied between them. This exercise led to codes that were later organized into categories and subcategories. Then, axial coding (Figure 4.14C) was the step to relate the codes with their categories with the effort to describe how the concepts are grounded in data, with interpretation and abstraction. This process was already described in previous sections.

Finally, in selective coding all the concepts associated with interoperability are integrated and the analysis refined (Figure 4.14D). These two central activities (integrating and refining) support the identification or proposal of a core category. The core category is essential in the phenomena under study and for all identified categories as well; it is frequent in the data analyzed, and it is the main concern that relates the other categories. All the categories (Figure 4.14D) generated relate and support the core category: **interoperability**, the central idea of this work.

The concepts behind each category are abstractions, where the data evolved to codes and then move from the description to a higher level of abstraction. The abstraction contributes to the generalization (giving a general rather than a specific view) and applicability of the findings.

In this work, the set of categories organized during axial coding offers an explanation and form a theoretical framework where the central idea: interoperability, as presented in Figure 4.15.

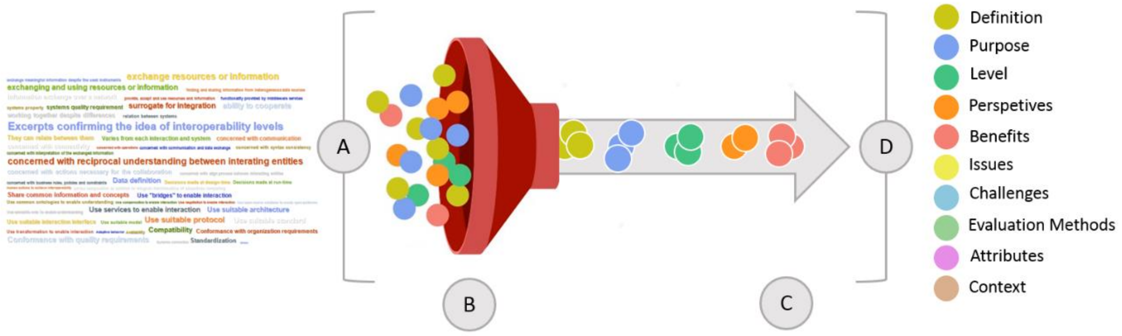

Figure 4.14 - Coding Process

Appendix A presents details of each category, with their descriptions and its relations with the codes that compose them. The relation of the categories to each other and how they relate to the main category are discussed in next section.

4.3 Results

After organizing the core category and its related concepts the data was arranged to observe the excerpts related to each concept. The relations found were identified by analyzing recurrent patterns and implicit meaning between codes, also considering groundedness and density. Once the concepts are related they were organized into a theoretical framework (STRAUSS; CORBIN, 1998), meaning the representation of concepts, together with their definitions and relations among them.

Figure 4.15 presents the conceptual framework. The concepts are divided into **structural** and **behavioral** owing to their significance for interoperability. The structural concepts are the ones that compose interoperability, the organization of the elements considered necessary to establish it. The behavioral concepts concerns to the observable interoperability, differing from the internal structural part. Once the interoperability is established (from structural concepts), it can be measured, improved and observed (from the behavioral concepts).

The relations in the theoretical framework (Figure 4.15) are described as **rn relationship (gn, dn)**, where **rn** means the relationship number, **relationship** is the connection role, **gn** is the groundedness of the relationship, and **dn** is the density of that relationship, as previously defined.

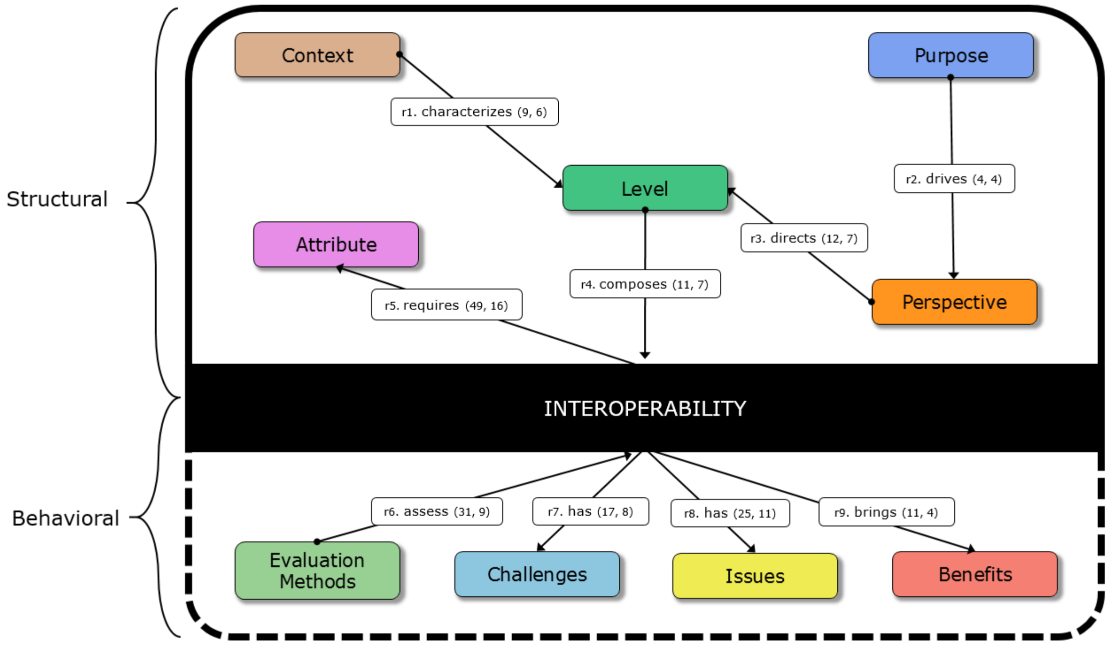

Figure 4.15 - Theoretical Framework

Each relationship is discussed below, and an excerpt of the data is provided to exemplify them. The excerpt also illustrated and represented as an instance of the framework. However, some of the chosen excerpts speak of relationships beyond the relationship discussed at the time. In cases like this, they are marked by a dotted line.

- **R1. Context *characterizes* Level:** Context information describes distinct features in Levels. We understand that interoperability can be observed through different levels **(technical, syntactic, semantic and organizational)**, depending on the level of support provided **(attributes and interoperability mechanisms)**. From the data analyzed, context information **(business information, extrinsic environment information, interaction time, and intrinsic environment information and user knowledge)** does not prevent the interaction. The system required structure to be interoperable will continue to exist regardless of context information. What is changed is the adequacy **(the quality of being good enough for a particular purpose)** of a given interaction.

Excerpt example:

"As important, an interoperable system ***needs to process information in ways that are meaningful to the other systems*** *with which it interoperates. To the extent that*

the meaning and form of information is ***dynamic (i.e., can change over time),*** *these systems need* ***to be able to modify dynamically their information*** *processing approaches and, possibly, their representations."* (MADNI; SIEVERS, 2013)

To our interpretation, semantic level enables mutual understanding of the meaning between interoperable systems. Over time, the meaning and form of information can change, which can affect the interaction itself. If mutual understanding is required, but the information cannot be understood, the interaction is meaningless. As a consequence of this relationship, represented by the dotted line, to adequately address the interoperability some attributes may be required. In this example, to address this concern, an adaptive behavior is necessary so the system can respond to any change during interaction time.

Representation:

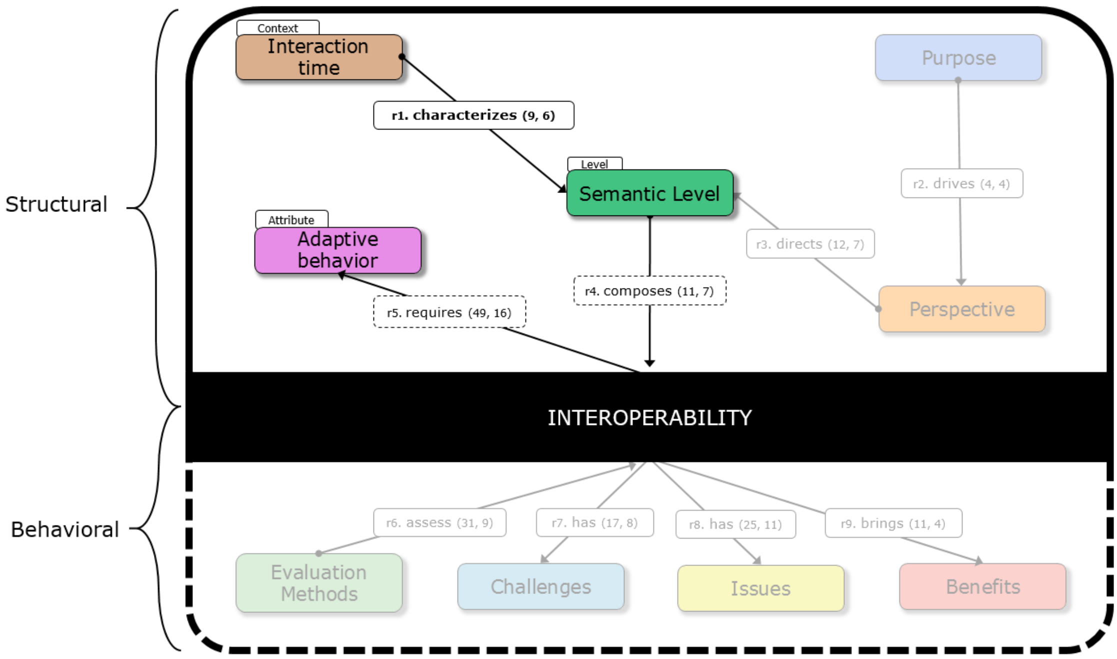

Figure 4.16 - Example: Context characterizes Level

- **R2. Purpose *drives* Perspective:** From the proposed interoperability definition interoperability comes as a way to enable a purpose. This purpose motivates the requirements and decisions to be made. Moreover, as presented before, the interoperability is perceived in different manners according to the perspectives and is the need for interoperability that drives each perspective **(organizational, systems and services)**.

Excerpt Example:

*“It refers to work in a harmonized way **at the level of organization and company** in spite of for example, the different modes of decision-making, methods of work, legislations, culture of the company and commercial approaches etc. **so that business can be developed between companies**.”* (CHEN, 2006)

The organizational perspective addresses the particularities of enterprises. Therefore, if the technology is required to support their interactions, interoperability is a way to overcome their differences and should be treated according to their purpose. The purpose, to develop business, drives the organizational perspective and surpasses the differences like cultural aspects, legislations, organizational structures, and others. As a consequence of this relationship, represented by the dotted line, to surpass the differences, the perspective directs the levels to compose interoperability, discussed in the next relationship.

Representation:

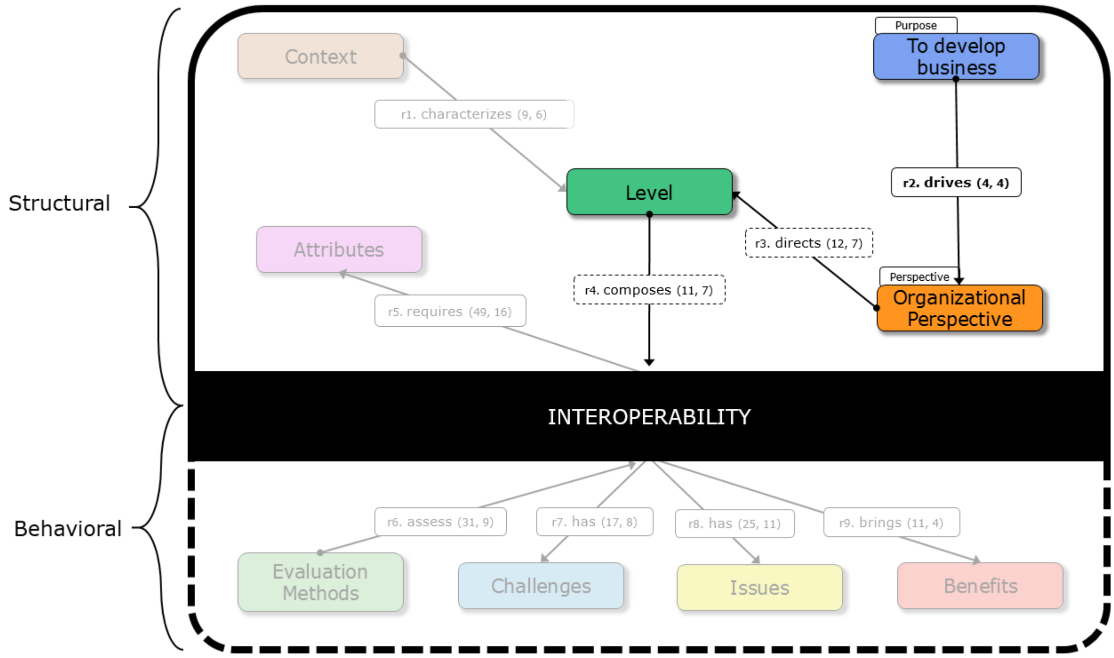

Figure 4.17 - Example: Purpose drives Perspective

- **R3. Perspective *directs* Level:** According to the perspective, interoperability needs are perceived in four different levels. The particularities of each perspective (**organizational**, **systems** and **services**) direct how interoperability should occur and in which level (**technical**, **syntactic**, **semantic** or **organizational**). Therefore, the differences and details of each perspective should be taken into account, according to the purpose, and reflected in the levels to properly achieve interoperability.

Excerpt Example:

*"At the **service level**, both **technical (compatibility between service signatures)** and **semantic interoperability (semantics and behavior of services)** between service end-points must be established. Service discovery mechanisms are used for this purpose and the decision-making procedures are bilateral. Service level interoperability means capability of interoperation between electronic services with well-defined, self-descriptive interfaces."* (RUOKOLAINEN; KUTVONEN, 2006)

What the authors call "service level", we interpreted as service perspective, because it requires particular decisions and has specific concerns. The service perspective drives the technical and semantic level that compose the interoperability requirements needed in the excerpt example. As a consequence of this relationship, represented by the dotted line, some mechanisms and attributes are required to compose the interoperability as necessary in the excerpt. Worrying about compatibility between service signatures is not a concern to the concept of interoperability as a whole, but it is a particularity when dealing with services that can be tackled with technical decisions. Similarly, the semantic level deal with the matter of semantics and behavior of services, a particular need for the service perspective as well.

Representation:

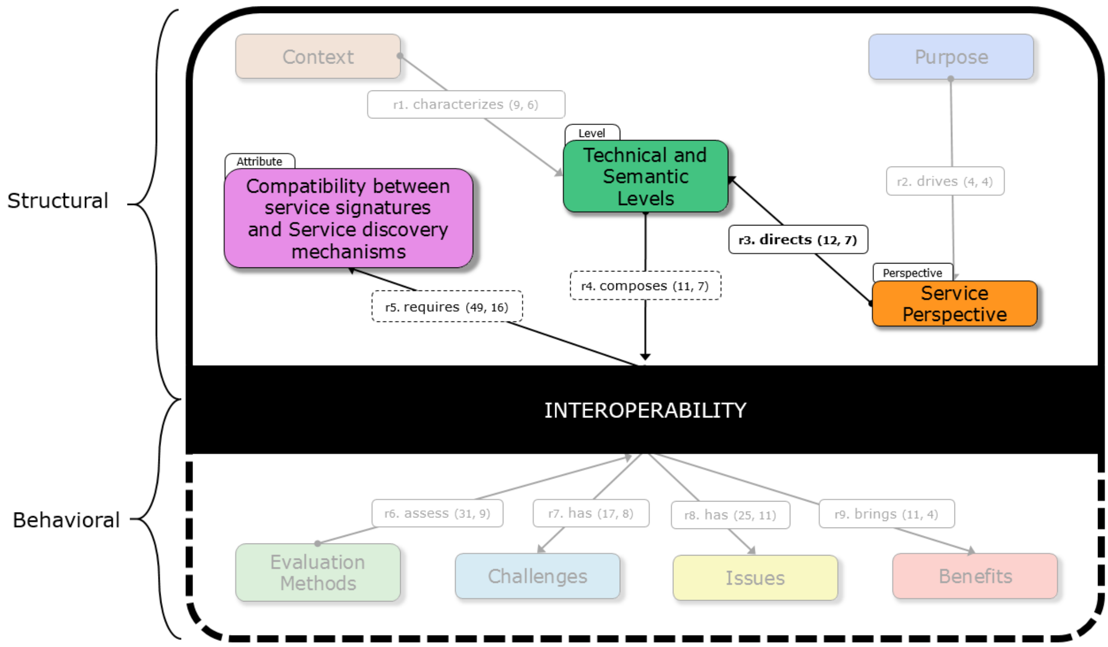

Figure 4.18 - Example: Perspective drives Level

- **R4. Level *composes* Interoperability:** The degree of interoperability is defined case by case because it depends on the level of support provided by each interacting system. The levels **(technical, syntactic, semantic and organizational)** compose the interoperability in a given relation. It is not every interoperation that requires every level; they vary according to the perspective. The levels are driven by the perspective that guides how interoperability should take form, which makes the interoperability adequate for that particular case.

Excerpt Example:

*"Interoperability is achieved on **multiple levels**: **interenterprise** coordination, **business** process integration, **semantic** application integration, **syntactical** application integration and **physical** integration."* (CHEN; DOUMEINGTS; VERNADAT, 2008)

Abstraction is necessary in a coding activity and often require bringing codes together based on their relating concepts. That is the case with inter-enterprise coordination and business process integration; these concepts are related with organizational level. This level is concerned with actions necessary to make entities collaborate on a higher level. These concerns are very different from the ones reported regarding application integration, which was reported as semantic, as well as syntactic. Last, but not least: physical integration representing the technical level, concerned to establish connection and operation regarding the interacting entities. Despite being different, they represent and distinguish in what levels systems interoperate, composing the interoperability concept.

Representation:

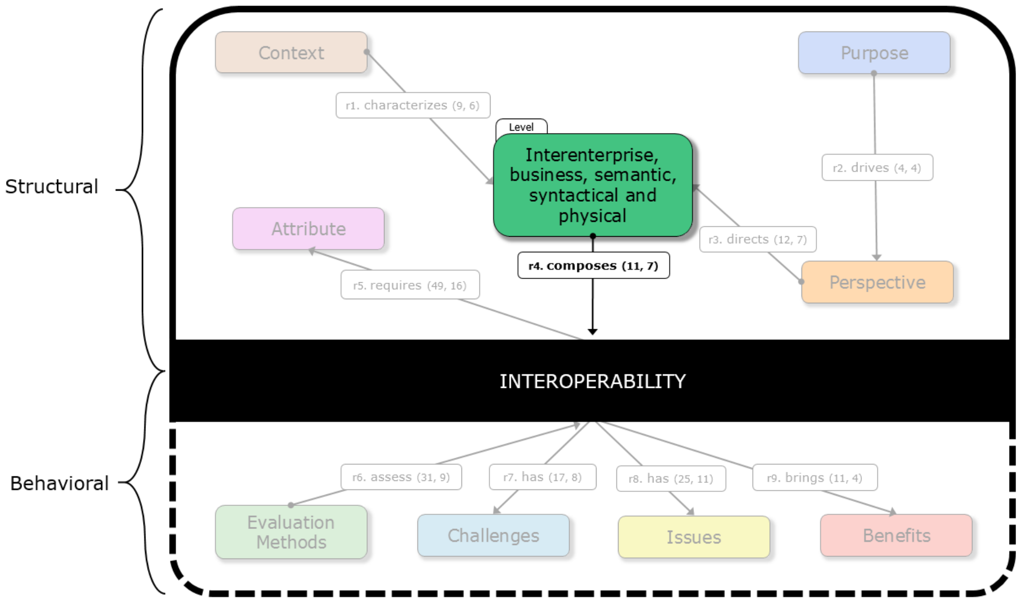

Figure 4.19 - Example: Level composes Interoperability

- **R5. Interoperability *requires* Attributes:** The perspective drives the levels that compose interoperability in a given interaction. Each level, in their turn, adjusts the requirements to enable interoperable systems. The interoperability required attributes depend on the levels that compose the interoperability. Coded in this concept are attributes and interoperability mechanisms. **Attributes** are typical systems' characteristics while **interoperability mechanisms** are the decisions to be taken, and solutions recurrently used to address interoperability. The attributes and mechanisms are how we can translate the needs of each level, presented in an abstract form, in possible solutions and more practical decisions.

Excerpt Example:

"Even so, there are ***internal concerns that need to be addressed by any system****, if some day it is called on to interoperate with others. These include adherence to standards, choice of information processing procedures and algorithms, the validity criteria surrounding the representation and processing of information, interfaces to other systems and to their users, and the nonfunctional, quality attributes such as reliability, availability, security, privacy, and information assurance."* (MADNI; SIEVERS, 2013)

From the excerpt above, we can see:

- **Attributes:** reliability, availability, security, privacy, and information assurance.
- **Interoperability Mechanisms:** adherence to standards, choice of information processing procedures and algorithms, the validity criteria surrounding the representation and processing of information, interfaces to other systems and their users

More than saying that an interoperable system should have a specific attribute, this concept aims to state that any system or components for which the interoperability is needed should have attributes and interoperability mechanisms in common, because, without these requirements above, interoperability cannot be achieved.

Representation:

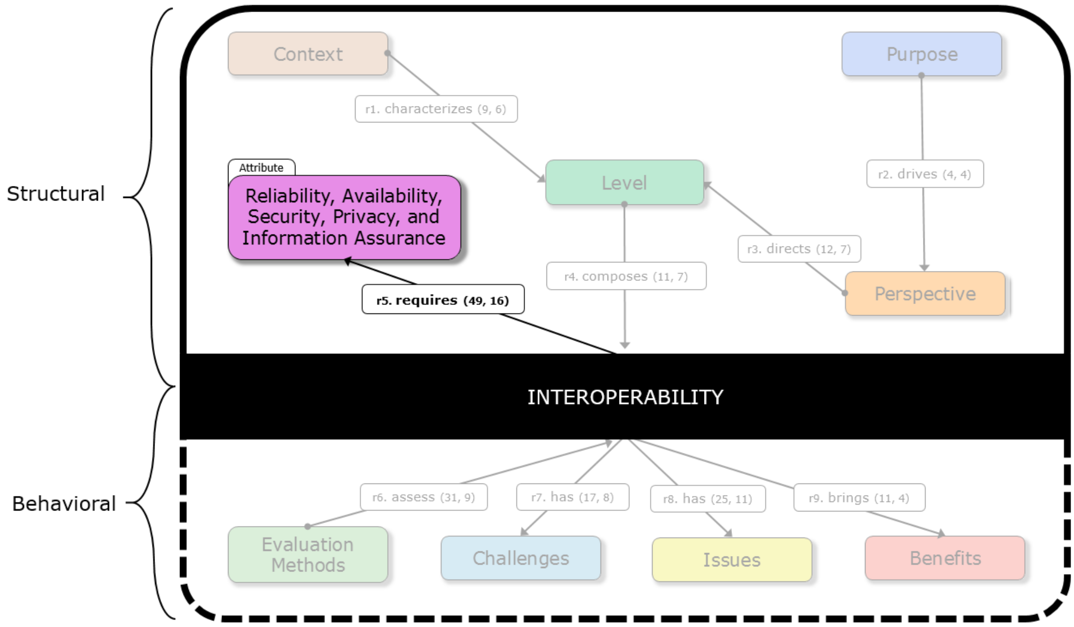

Figure 4.20 - Example: Interoperability requires Attributes

- **R6. Evaluation Methods *asses* Interoperability:** The concept, and therefore this relationship, arises from one of the research questions of the systematic review undertaken: **How to evaluate interoperability in context-aware software systems?** The findings present 20 different evaluation methods in 33 excerpts coded. The methods were analyzed with the same perspective, trying to retrieve any relevant information like metrics and attributes. However, each method deals with different domains, approaches, and goals being hard to unify or compare. Another situation was to see how little these methods consider context as a variable.

Excerpt Example:

*"The **interoperability assessment methodology** model introduced nine components which were **requirements, node connectivity, data elements, protocols, information flow, information utilization, interpretation, latency**, and **standards**."* (REZAEI *et al*., 2014)

The interoperability assessment methodology is "a method that resolves system interoperability deficiencies through the measurement and quantification of a set of interoperability system components. The objective assessment of interoperability thus becomes the sum of the assessments of the individual characteristics" (LEITE, 1998). Table C.1.8 presents more details. As a consequence of this relationship, represented by the dotted line, the structural part can be revised to resolve the deficiencies found.

Representation:

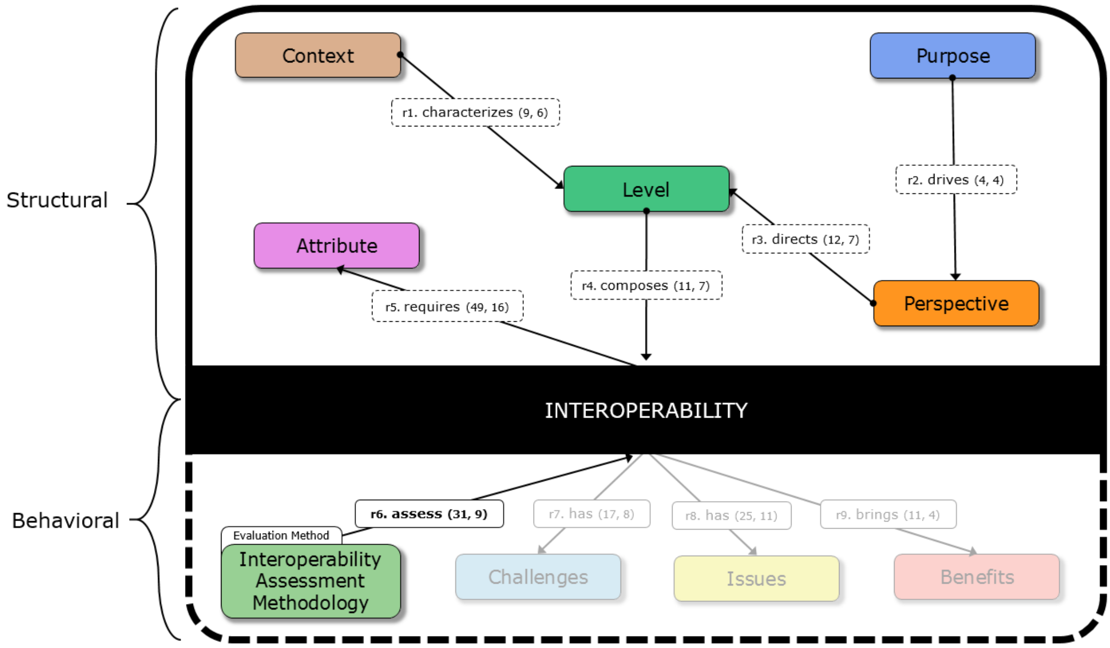

Figure 4.21 - Example: Evaluation Methods asses Interoperability

- **R7. Interoperability *has* Challenges:** Like any other research topic, interoperability has its challenges as well. These well-known challenges, such as interoperability of new systems with legacy ones, or future work and open issues, are highlighted in the analyzed studies. From them, we separate between research or practice challenges.

Excerpt Example:

*"One **future challenge** is to develop Service-Oriented Architectures adopting a federated approach, i.e. **allowing interoperability of services 'on the fly'** through dynamic accommodation and adaptation."* (CHEN; DOUMEINGTS; VERNADAT, 2008)

In the excerpt example, we can see that interoperability "on the fly" or spontaneous interoperability (SPÍNOLA, 2010) is a challenge. As a consequence of this relationship, represented by the dotted line, to address this challenge, it is necessary more research and requires that the systems should pose an adaptive behavior to support dynamic interoperation at runtime (SULLIVAN; LEWIS, 2003).

Representation:

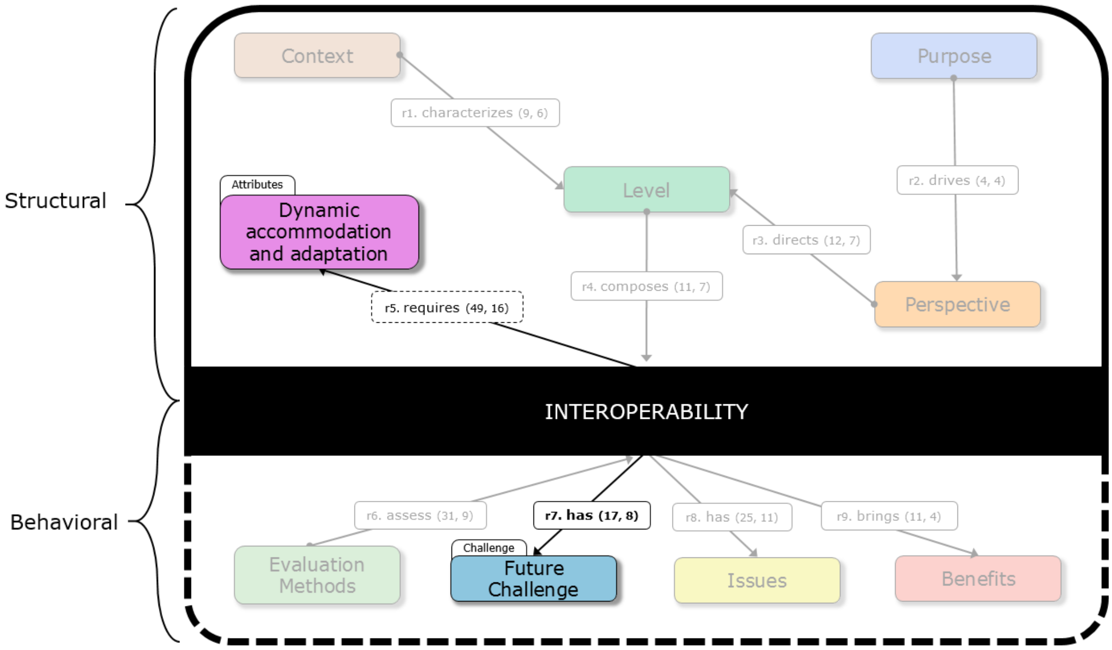

Figure 4.22 - Example: Interoperability has Challenges

- **R8. Interoperability *has* Issues:** Management and developing decisions lead to consequences, and interoperability is no different. We decided to code the concepts as issues because it could encompass problems, concerns, and difficulties when addressing interoperability. Issues are different from challenges because they are more related to the decisions to achieve interoperability in a given interaction than to the interoperability concept evolution.

Excerpt Example:

*"It is also the case that **semantic interoperability** is a substantial, potentially costly undertaking (as a design goal) because of increased **validation and verification complexity**."* (MADNI; SIEVERS, 2013)

The semantic level composes interoperability in the excerpt example. To achieve semantic interoperability some mechanisms, as sharing common information and concepts, are required. Such mechanisms can lead to issues regarding verification and validation (V&V). Semantic specific V&V activities may not be in the original scope of the project or inadequately planned. As a consequence of this relationship, represented by the dotted line, it brings an interoperability issue related to the semantic level, the required attributes and mechanisms, since it is difficult to check for consistency and assure the interpretation of the information by each interacting system (MADNI; SIEVERS, 2013; REZAEI *et al.*, 2014).

Representation:

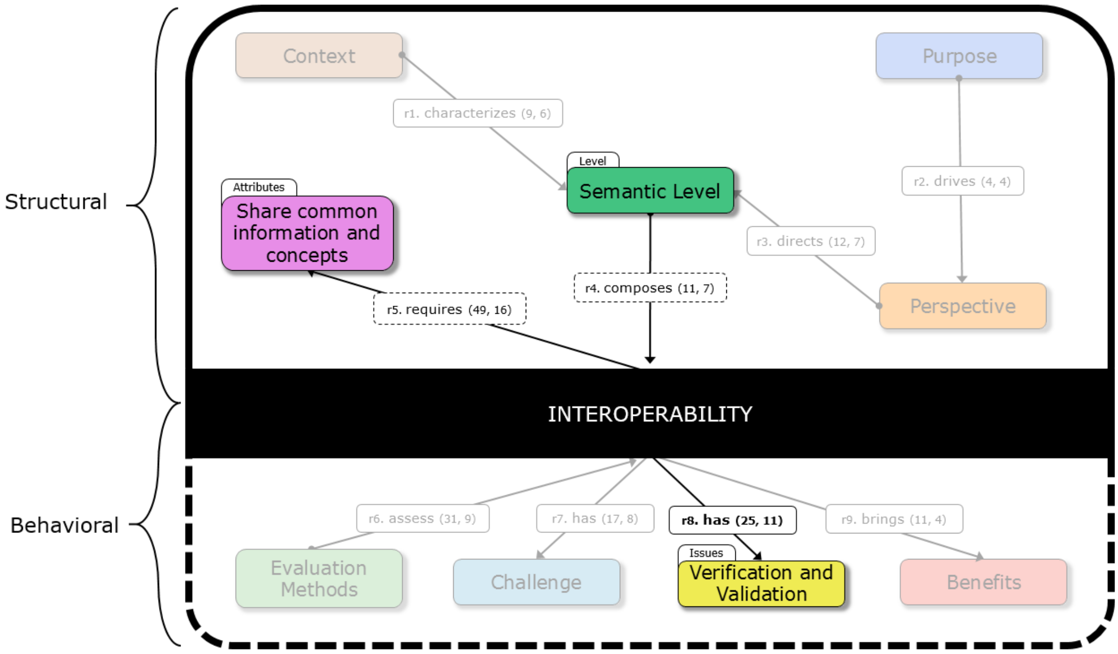

Figure 4.23 - Example: Interoperability has Issues

- **R9. Interoperability *brings* Benefits:** From the analyzed data, we recurrently came across positive consequences, presented in the concept of benefits, which can happen as a result of interoperable systems. Interoperability can be an internal requirement, necessary for project success for systems development. It can also be a request for the interaction itself, to achieve a purpose. Either way, benefits are positive side effects when interoperability is achieved, not the purpose or motivation to interoperate.

Excerpt Example:

*“**Interoperability offers a number of advantages including**: (a) increased flexibility, by allowing mixing and matching of systems; (b) creation of new capabilities, by composing new functions from existing ones; and (c) increased cost-effectiveness, by allowing reuse of existing systems and capabilities.”* (MADNI; SIEVERS, 2013)

Three different benefits are presented in the example: increase flexibility, create new capabilities and increase cost-effectiveness. These benefits can be related to systems reuse, but many others were also observed in the analyzed data. Some of the benefits and the interoperability itself contributed to the system’s capability to make a certain functionality from basic components, as required by the user and desired in ubiquitous computing.

Representation:

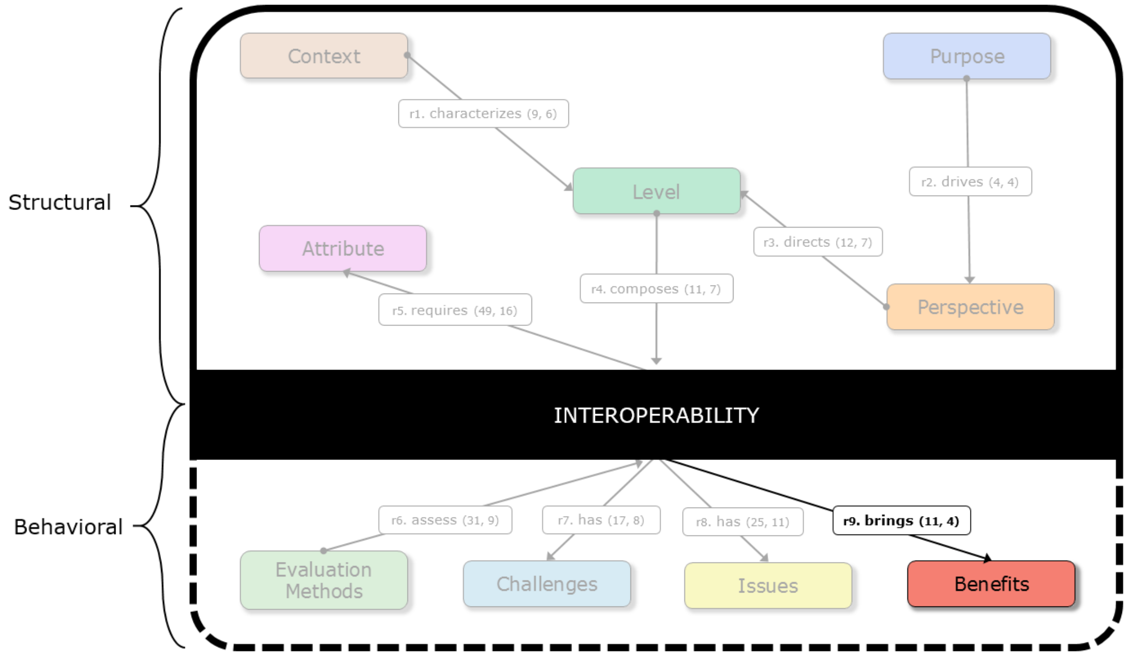

Figure 4.24 - Example: Interoperability brings Benefits

4.4 Threats to Validity

At Planning and Data Collection: the analysis was motivated by the qualitative data retrieved from the literature and the planning occurred in internal discussion among the researchers involved. The strategy used for data collection is documented in the *q*SLR protocol and presented in Chapter 3 as well. No sampling problem was identified because all the included papers were selected and analyzed. The data extracted from the selected papers was revised prior the analysis by other two researchers to reduce bias. Since the analysis was conducted in the studies from the *q*SLR, the threats reported in the secondary study can also be applied here.

At Coding: the primary activity in the analysis is the coding process, with the major validity threat for any qualitative studies that is the researcher bias. To reduce this bias, a second researcher accompanied the coding process and a third revised all the codes during open coding. Partial results were presented to other researchers to gather independent feedback. To mitigate interpretation bias, the meaning of a concept was clarified by the use of a dictionary to avoid any misinterpretation or information lost in translation. Also, QDAMiner Lite, a qualitative analysis software, heavily supported this process. Each code, comments, and thoughts from the data is recorded and organized.

At Reporting: a concern is on the conclusions being reasonable and based on data. The results presented were drawn by an analysis of the data, systematically retrieved from a qSLR and in collaboration with other researchers. We relied on the methods from literature as guidance for the analysis, and the results represent a synthesized view of the concepts related to interoperability in context-aware systems, as found in the technical publications.

4.5 Chapter Considerations

This chapter reported the analysis of the data extracted from the *q*SLR performed, using the principles of GT method (STRAUSS; CORBIN, 1998). Through the analysis, the major concepts related to interoperability emerged, from the studies selected in the *q*SLR that represents the state of the art in the technical literature. The concepts and their relation are presented in conceptual maps and theoretical framework, with examples to support their proposal as well the reasoning and interpretation behind each concept.

The coding process is challenging and was important to have the support of other researchers during the whole analysis. Some difficulties are related to the abstraction and how to raise the concept level without losing what is relevant from this research goal. Conceptualizing, observing the similarities and differences among the excerpts, keeping the consistency among the relating concepts and working in data interpretation are crucial activities to any qualitative analysis.

However, the effort carried out was necessary for the analysis and the valuable contributions generated, for instance, the definitions proposed and the consequences of the observed relationships in the development of context-aware or any other system concerned with interoperability. Moreover, theoretical sampling was not applied.

In qualitative analysis, like any other research, it is necessary to consider the limitations of the study undertaken, and the threats to the validity of analysis performed were presented. In addition to considering the threats posed, it was also presented some concerns reported by Strauss and Corbin (1998) regarding the adequacy of the research process and the grounding of the findings.

Adequacy of the research process: the original sample was selected based on the data extracted of the *q*SLR undertaken. The major categories are presented in the axial coding subsection and emerged considering its groundedness corresponding to the data and suitability for the substantive area of interoperability. The relationships are presented considering density and discussed in the results subsection. These conceptual relations lead to the core category, Interoperability, the central theme of this research.

The grounding of the findings: each concept can be linked to the codes, and then to the excerpts, it is grounded. Some of the concepts are from common usage to the concepts of Level. Others are derived from this analysis, like the concept of Purpose. When a concept relates to another, the relationship is presented in a conceptual map and supported by examples from the data analyzed. These examples also illustrate the variations and conditions of which the concepts were examined and developed.

We understand that despite the adequacy of the research process and the grounding of the findings different researchers may have a different understanding of the data, leading to another interpretation of the concepts and the relationships. These are some of the limitations of these results. However, even with its limitations the results reported in a theoretical framework has its benefits. Since it is based on concepts, it can be flexible enough to evolve, be refined or modified according to the evolution of the interoperability and the technology itself. With the concepts derived from the analyses we were able to answer the research questions, presented in the next chapter.

5 Interoperability Characterization

The findings of the quasi-Systematic Literature Review turned into concepts after the analysis performed. These concepts are discussed in this chapter, characterizing interoperability and answering the research questions that drive this work.

5.1 Introduction

During the early stages of this work, we had expectations that we could clearly observe the influence of the context variation in interoperability between systems. However, during qSLR we understood that the concepts could have different meanings in technical literature, hindering the observation, as initially desired. The excerpts below illustrate how some concepts are understood differently:

> *"The context dimension refers to the conformance between the context representations used by service providers and the context representations requested by service clients."* (ATHANASOPOULOS; TSALGATIDOU; PANTAZOGLOU, 2008)

> *"Context is another level when talking about service interoperability, especially in ubiquitous computing environments. Context information refers to any information acquired by a sensor (e.g. position, time, temperature, speed, etc.). Obviously, a service may deliver different results depending on different context information. For example, the result of a service may satisfy user's need better when it is invoked under 0 °C."* (FANG; HU; HAN, 2004)

The authors in the examples have a different understanding of what context is. Nevertheless, in both cases, context is relevant to characterize an entity (services). Other examples are discussed in Chapter 4, that support our decision to give a step back and organize the knowledge produced so far in a theoretical framework, with the GT support.

Many of the concepts identified as relevant, however different, were kept after the analysis. Thus, it was possible to enrich the organization of the concepts as they had helped to answer the research questions. With the large amount of qualitative data generated at the end of the *q*SLR, the analysis guided by GT principles, it was necessary to make the findings more synthesized and homogeneous.

Even though the questions remained the same since the beginning of this research, the expectations, and their responses were becoming closer to reality as the recovered knowledge was being fragmented (data extraction and open coding) and reorganized (axial and selective coding). The answers presented here are the results of discussions that took place by observed concepts, presented in Chapter 4.

5.2 Answers to the Research Questions

5.2.1 Main research question: How is interoperability addressed in context-aware software systems?

With this question we intend to observe how interoperability is treated in systems whose behavior can be changed by the context information. The main question was intentionally designed to be wide: our boundary is context-aware software systems, with the intention to retrieve as many relevant articles as possible to meet our goal of characterization. Our aim was to focus on interoperability in a broader sense and to characterize, based on technical literature, what the general researches conceptions are when addressing interoperability in this type of system.

A recurrent perception in the *q*SLR findings and during the data analysis is that interoperability can be interchanged with the following concepts: connectivity, integration or exchange (Figure 4.4). The decision on first observing the studies' definitions was to understand the mindset and perspectives behind the selected works and solutions. If a study sets that interoperability is integration, the effort and results of the work will vary from another one in which interoperability is set as communication. Moreover, the definitions can provide the foundations for any further development. Therefore, understanding this concept using general software system perspective is the first step towards its comprehension and observation in other contemporary scenarios, such as context-aware software systems.

From the excerpts analyzed and coded we propose the following interoperability definition:

The ability of things to interact for a specific purpose, once their differences have been overcome.

This definition is generic enough to cover the definitions we found while also reporting the complexity of the interoperability concept. The use of term **things** was decided because we are dealing with systems and its components as well. However, we took advantage of the topic, also considering smart things. In this way, the definition is more comprehensive. The **interaction** stands for any relation where a system, component, service or thing can engage with another. That way it covers exchange and trade of any kind. Together with **for a specific purpose** the definition embraces the idea of cooperation and collaboration, frequently related to interoperability while uniting with one of the concepts uncovered during the data analysis. To affirm in its definition the

differences each one of the interacting things can have is a step towards comprehending interoperability. Once the differences are known, it is easier to **overcome** them to properly achieve interoperability.

In Chapter 2, in the interoperability concept introduction, two different definitions were presented. One of them was given by the American Department of Defense (FORD *et al*., 2007):

> *The ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together.*

Matching our proposed definition with this one we can see that system, units and forces are the **things** that should interact. The **purpose** is to operate effectively together, and the **interaction** is to provide, accept and use services.

Another definition previously presented is one by IEEE (1990):

> *The ability of two or more systems or components to exchange information and to use the information that has been exchanged*

Things can embrace systems and components as presented in this definition, where the **interaction** is exchange and use information.

In both cases interoperability is concerned with identifying, composing and enabling these entities, often designed and implemented separately, to work together for a purpose. Moreover, the differences arise precisely during the interaction. To overcome these differences in a way that the interaction can happen, and interoperability achieved, some strategies are required as we have observed in the interoperability mechanisms and attributes presented.

After observing the general interoperability understanding, proposing and justifying our definition, one of the challenges in investigating context-aware systems is exactly the term: **context**. An effort was made during the coding process to differentiate this term in each study. Rezaei *et al.* (2014) and some referenced works (LAMPATHAKI *et al*., 2009; KOUSSOURIS; LAMPATHAKI, 2011) consider interoperability to the use of different representations, different purposes, different contexts, and different approaches. This vision refers to the various contexts of development and usage, not regarding contextual variation, which can affect the system behavior. In some other works (CHEN; DOUMEINGTS; VERNADAT, 2008; KUTVONEN, 2013) to be context-aware means perceiving the business context or user’s situation and preferences. In these cases, interoperability deals with systems working together, considering (and removing) the interaction barriers, but does not consider context as defined by Dey

(2001), the basis of our research. Gjoreski *et al.* (2015) present a context-aware system, with activity recognition where the system behavior acts according to the context. Interoperability is evaluated when setting up the scenario and observed in this initial phase when the systems are placed about starting the recognition testing.

Other papers (ATHANASOPOULOS; TSALGATIDOU; PANTAZOGLOU, 2008; MEMON *et al*., 2014) consider context as an interoperability dimension and refers to it as any information acquired by a sensor where the results can differ depending on different context information. In these cases, to achieve interoperability in context-aware systems, the solution lies in the use of ontologies, especially for semantic issues (BECKETT; MCBRIDE, 2004; MCGUINNESS; HARMELEN, 2004; RYAN, 2005). Chen (2006), based on ISO 14258 (1999), considers the federated approach, where interoperability is established on the fly with no prior contract or agreement. This spontaneous interoperability considers context switching as not having a pre-determined infrastructure and having to accommodate dynamically and adapt to the changing environments. Interoperability on the fly is what we believe to be a solution for the current issues of heterogeneity and dynamicity part of the context-aware systems and ubiquitous computing scenario. Nonetheless, from the data set we analyzed, this still is a research challenge in the field with few early efforts towards suitable solutions (BLAIR; BENCOMO; FRANCE, 2009; JUÁREZ; RODRÍGUEZ-MONDÉJAR; GARCÍA-CASTRO, 2014). (SULLIVAN; LEWIS, 2003) also states that the key to interoperability in context-aware software systems cannot rely on prior agreements, but somehow to be able to adapt at runtime to be interoperable. In this work, once again, ontologies come up as a possible problem solver. The complexity of this research area can be seen in other challenges to achieve true interoperability due to context. Heterogeneity of networks; different connectivity requirements in several processor forms (PCs, cell phones, tablets); poor application portability and increasing software platform dependency and structural challenges. As the solutions arise, also comes to a growing number of organizations needing to be involved in achieving the seamless interoperability implied by the ubiquitous computing vision.

Based on the definition provided by Dey (2001), we consider context information to be any data that can be used to characterize the situation of a person, place, or object that is considered relevant. From the selected articles and after the data analysis we identified five different context information:

- **Business Information:** concerns with the context information that guides the interoperability levels in the business (organizational) perspective bringing significance to the interactions.

- **Extrinsic Environment Information:** definition of extrinsic: a) not forming part of or belonging to a thing; b) originating from or on the outside; c) originating outside a part and acting upon the part as a whole. It refers to variables like position, time, weather, speed or any information that can be perceived by the system or acquired by a sensor.
- **Interaction Time:** it is related to the idea that the interaction time is a context variable that can influence the interaction among entities. It affects for instance: a) connectivity (e.g. the link should be available at any moment); b) dynamicity (e.g. information is dynamic, over time can change meaning and form).
- **Intrinsic Environment Information:** definition of intrinsic: a) belonging to the essential nature or constitution of a thing; b) originating and included wholly within a part. The environment out of the "operator" control on which the entities are interacting can influence the interoperability.
- **User knowledge:** the user's background can affect interoperability as knowledge compromises the user's comprehension and perspectives regarding a system, (related to semantic).

The information described above represent what the selected articles provide as context information. Therefore, we believe that interoperability concerns, when considering context-aware software systems, should be related to the devices or solutions used to acquire, store, control and use the information mentioned above (LEE; CHANG, 2010). For such cases, there is no "one size fits all" solution. Each context information as well the systems in which they are used has their specific requirements that should be respected and adequately addressed.

With the context information found, as presented above, we could face the expectation of the beginning of this research. Context variance affecting systems interoperability is observed when related to interoperability levels. The context information the system deals with should be known before systems interaction. It means that **levels** that compose the interoperability between two systems that will require attributes and interoperability mechanisms specific to deal with the context information. Given that, even if the interaction between the systems is long enough so that the context variation can change systems behavior, the infrastructure required for interoperability remains as it is part of the system itself. For the observed context information, the system demanded structure to be interoperable will continue to exist regardless of it. What can be changed is the adequacy of a given interaction.

So, how interoperability is addressed in context-aware software systems? From our findings, the answer to this question is still in an incipient stage. To explain it in one sentence it is clear that after many years, Weiser´s (1991) vision is still a vision regarding make things interact automatically based on their context and despite their differences. This vision was conceived at a moment where the Internet was premature, different from what we currently have, and there is no doubt about the evolution in this direction so far. Nevertheless, as far as our review shows, interoperability in context-aware systems is usually addressed in the initial development phase, when composing software systems, it is based on previous agreements and shared knowledge while the context is distinct information with limited influence in the software systems behavior when related to interoperability.

5.2.2 Secondary question: Which are the interoperability concepts and attributes used in context-aware software systems?

Initially, this question aimed to provide a list of interoperability attributes regarding context-aware software systems, from what we could distinguish them from other system properties. However, from the data retrieved, the gathered attributes are software systems characteristics relating to interoperability. In other words, if a software system has these attributes it will be easier to interoperate with other systems. We can say that interoperability cannot be (at high abstraction level) directly observed but perceived through its related attributes and results. Once again, all results presented here are from the analyzed material. Shown in Table 5.1 are the results of the coding process of the extracted data related to attributes, with their description in the light of context-aware systems.

Table 5.1 - Attributes observed in context-aware systems

Attributes observed in context-aware systems	
Attribute	**Description**
Adaptive behavior	*It refers to the ability to adapt dynamically to the environment the system is being used within its limitations. The adaptation is chosen from a set of possible alternatives known by the system. One example of how the adaptive behavior is relevant for context-aware software systems is: when identifying the bandwidth reduction to the point of harming the audio and video transmission, the streaming service continues but audio and video quality are reduced. This attribute is a particular concept, being relevant especially for context-aware software systems.*
Availability	*It is directly related to connectivity since all the interacting systems should be available during the interaction time. Availability refers to the state of a system being available, able to be used. In a dynamic environment, availability may vary as a function of the time, being the interaction time a context variable observed in our analysis. One example of how the adaptive behavior is relevant for context-aware software systems is: GPS satellites and the mobile phone should be available at all times to collect the correct and up-to-date position data. This attribute is a general concept, being relevant not only to context-aware software systems but any interoperable system.*
Compatibility	*Compatibility of a system deals with what ease the system can operate with shared applications. It also covers the system's compatibility on different-different platforms. The platforms can be either hardware, software or both. One example of how the compatibility is relevant for context-aware software systems is: extrinsic environment information, like temperature, should be in a compatible scale between the sensor and the system. This attribute is a general concept, being relevant not only to context-aware software systems but any interoperable system.*
Conformance with organization requirements	*Each organization has its issues that should be aligned with the other ones to interoperate. Related to decision-making and also non-technological aspects. One example of how the adaptive behavior is relevant for context-aware software systems is:*

	including context into access control and support resolutions based on the current situation, dynamically adjusting user role and permissions according to organization rules. This attribute is a general concept, being relevant not only to context-aware software systems but any interoperable system.
Conformance with system's requirements	*It refers to an explicit prescription of the general requirements that should be present in systems that desire to interoperate. If quality constraints are not satisfied, the systems are not suitable to be interoperable. One example of how the adaptive behavior is relevant for context-aware software systems is hardware components from sensors, mobile client, and servers should be supported regardless the variety of networking settings and programming languages. This attribute is a general concept, being relevant not only to context-aware software systems but any interoperable system.*
Dynamic connection	*It refers to enabling interaction between entities; it is necessary to identify the relating ones. Decisions like these relate to connecting with the identified partners according to their permissions. Authenticate and authorize interactions in automatic connection based on previous interactions (history) or re-establish lost connection. One example of how the dynamic connection is relevant for context-aware software systems is: one application adjusts the phone status based on the user's current location, the time of the day, and the network in use. Once the user is in the work environment, the connection is automatically established and change the device preferences like converting the email account setting to the office account. This attribute is a particular concept, being relevant especially for context-aware software systems.*
Standardization	*It refers to bringing conformity between the systems that have to interoperate. Some decisions cannot necessarily be an actual standard, but should have an agreement by all parties wishing to interoperate, to ensure compatibility and integration with different systems. Despite the efforts in the field to spontaneous interoperability, where no a priori knowledge is shared, until now standardized solutions are necessary to achieve interoperability.*

One example of how standardization is relevant for context-aware software systems is the use of practices and standards for spatial and geographic information in applications, like a standardized system of coordinates. This attribute is a general concept, being relevant not only to context-aware software systems but any interoperable system.

Along with the attributes, we also recovered what we call **interoperability mechanisms**, which are decisions to be taken and solutions recurrently used throughout the systems to achieve interoperability. Before the coding process, we recovered 99 interoperability mechanisms from the selected studies. Its origin initially classified These mechanisms (if proposed by the authors in the paper or if it presents references to support their existence) and **evaluation** (if it is evaluated in the paper or if it does not perform any assessment). Due to inadequate description of some mechanisms and high recurrence of others, the mechanisms were also coded. This initial analysis, after the coding process, gave place to 18 higher level interoperability mechanisms.

With the presentation of these mechanisms, we do not seek to introduce a definitive list of solutions. We believe that technology evolves, and the most appropriate solutions vary over time and from system to system. In fact, these mechanisms aim to be a guide for decision making, once the need for interoperability arises. This need can be planned, considering interoperability as a requirement during systems development, or it can be included in existing systems, designed without interoperability considerations. Either way, the mechanism listed in Table 5.2 can offer support.

Table 5.2 - Interoperability Mechanisms observed in context-aware systems

interoperability Mechanisms observed in context-aware systems			
Id	**Mechanism**	**Description**	**Related attributes**
1	**Data definition**	*To interact with the data, it is important to make decisions related to the data type, data structure, data representation, data format, language – and this also is related to the standards to be used and the consistency between the relating entities. One example of how data definition is relevant for context-aware software systems features to assure data type representations like the values retrieved by sensors in numerical form having the same accuracy and format.*	**Compatibility, Standardization**
2	**Decisions made at design-time**	*Design-time interoperability is the matter of analyzing the effort needed for possible future systems to interoperate, regardless of their current relationship to each other. One example of how decisions made at design-time are relevant for context-aware software systems is the use of ontologies in ubiquitous computing for integration at design-time.*	**Conformance with organization requirements, Conformance with systems requirements**
3	**Decisions made at runtime**	*Runtime interoperability is concerned with the interoperability of systems that are supposed to be working together, after systems completion the need for interoperability comes. One example of how decisions made at runtime are relevant for context-aware software systems is: decisions related to modifying key features of an existing system to make it interoperable will require effort and raise costs.*	**Conformance with organization requirements, Conformance with systems requirements**
4	**Human actions regarding interoperability**	*It addresses unexpected business issues requiring human reasoning. One example of*	**Conformance with**

		how human actions are relevant for context-aware software systems is related to the choice of services and devices to be used in the desired task.	**organization requirements, Conformance with systems requirements**
5	**Share common information and concepts**	*Information specifications provide the base for semantic exchange such as standardized terms, terminology and controlled vocabularies; data definitions; units of expressions; computational methods and assumptions; precise definition of limits and restrictions and so on. Often this is more than simple semantic, and the concepts have to be aligned with organizational conditions. One example of how common information and concepts are relevant for context-aware software systems is a shared understanding of patient conditions with data shared across different health care environments.*	**Compatibility, Standardization**
6	**Use "bridges" to enable interaction**	*A posteriori solution that acts as an intermediate between inter-related systems enabling interoperability. One example of how the use of "bridges" is relevant for context-aware software systems is the use of middleware as an intermediate layer between systems and devices.*	**Adaptive behavior, Availability, Dynamic connection**
7	**Use common ontologies to enable understanding**	*The use of ontologies is pointed out to be a way to provide name, the definition of the properties and the relationships between the entities, to bring compatibility to achieve a mutual understanding. One example of how the use of ontologies is relevant for context-aware software systems is that ontologies can be a solution for modeling contextual*	**Compatibility, Standardization**

		information because it can be evolved and extended.	
8	**Use compensation to enable interaction**	*A posteriori solution is to use some compensation as an alternative to something that went wrong. One example of how compensation is relevant for context-aware software systems is: because their adaptive behavior, other alternative services can be used if the initially desired one is not available.*	**Adaptive behavior, Availability**
9	**Use negotiation to enable interaction**	*Negotiation mechanisms are used when different actors influencing different systems have different views, objectives or business processes that must be taken into account when trying to build an interoperable system. Negotiating solutions are more related to organizational decisions to find mutual agreements to interact.*	**Availability, Conformance with organization requirements**
10	**Use open source solutions**	*Necessary for a primary goal to create open platforms. One example of how open source solutions are relevant for context-aware software systems is to popularize the use of open solutions going against proprietary solutions.*	**Conformance with organization requirements, Standardization**
11	**Use semantic web to enable understanding**	*It provides a common framework allowing data to be shared and reused across applications, enterprise, and community boundaries aiming reciprocal agreement between things. One example of how semantic web is relevant for context-aware software systems is: knowledge compromises the comprehension and interpretation of the information in the systems, being considered a context variable. Semantic web come as a possible solution to bring this comprehension to*	**Compatibility, Standardization**

		systems, to an unambiguous and understandable form.	
12	**Use services to enable interaction**	*Some authors defend the idea of using services as a general solution to ubiquitous computing, Internet of Things and others. One example of how services are relevant for context-aware software systems is: all the devices and applications present in a ubiquitous environment will only interact through services.*	**Availability, Dynamic connection**
13	**Use suitable architecture**	*Architectural decisions have consequences affecting the whole system infrastructure. One example of how architecture is relevant for context-aware software systems is if it is decided to use services, it will also require decisions related to the technology to be used (SOAP, Rest, other?). It is necessary to decide on the architecture to address connectivity and communication issues.*	**Availability, Dynamic connection**
14	**Use suitable interaction interface**	*To achieve conformance between parts to enable the interaction, the designers should decide on operations, types, the order of parameters and others technical issues. One example of how the interface is relevant for context-aware software systems is a method of a library not compatible with the calling program.*	**Standardization**
15	**Use suitable models**	*Like standards and protocols, the use of compatibles models is necessary to achieve interoperability. One example of how models are relevant for context-aware software systems is: a reference model is available to any other system that makes use of a given service.*	**Compatibility, Standardization**
16	**Use suitable protocols**	*A protocol stipulates the rules for communication between entities, and a*	**Compatibility, Standardization**

		standard is a specification agreed upon and implemented and adopted by industry/vendors. One example of how protocols are relevant for context-aware software systems goes from simple internet protocols to security data protocols.	
17	**Use suitable standard**	*A standard is a document providing requirements, specifications, guidelines or characteristics that can be consistently used to ensure that materials, products, processes and services are fit for their purpose (ISO definition). One example of how standards are relevant for context-aware software systems is Health Level Seven International (HL7)[11] standard for healthcare interoperability.*	**Compatibility, Standardization**
18	**Use transformation to enable interaction**	*It requires transformation or modification of one of the systems to achieve interoperability. One example of how transformation is relevant for context-aware software systems is that adaptations can go from data representation modifications to alter the communication pattern.*	**Adaptive behavior, Availability**

It is important to make considerations regarding the decision of using such mechanisms because it encompasses the necessity to provide the required structure for each one.

When taking context-awareness into account, the following characteristics were found: monitoring changes of state to trigger actions within the environment, dynamically modifying their information processing approaches and representations, runtime monitoring to maintain information up to date and monitoring of service behavior on the contract. All of these consider context variance and modifications at runtime to perform according to the changes. These characteristics exemplify a particular attribute of context-aware software systems: **adaptive behavior**. This behavior refers to the system ability to adapt itself dynamically to the environment where it is used within its limitations;

---

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

that calls for monitoring the systems environment and adjusting it to a different situation or condition. Monitoring tasks in such software systems demand systems **availability**, meaning that the systems should be in the correct operational state during the whole interaction. The system´s availability attributes lead to other concerns such as connectivity. These are examples of how the interoperability mechanisms relate to the required attributes and are present in one system.

The attributes found deals with two aspects of context-aware systems presented in the introduction. After that, interoperability comes as a way to solve **heterogeneity** and **dynamicity** issues. The attributes of **compatibility** and **standardization** tackle heterogeneity issues while **adaptive behavior** and **dynamic connection** deal with the dynamicity. The tricky part is how to achieve these attributes, and there is where the mechanisms are essential.

Most of the listed attributes and interoperability mechanisms are not restrict to context-aware software systems. They can be naturally observed in any software system on which interoperability is important. These attributes allow seeing the importance of technical decisions during systems development. To deliver interoperable solutions, the concerns about interoperability must start at the conception phase. Resolutions at this step cover decisions on the infrastructure and the use of standards as well. Having compatible processes and data are more complex tasks, but having these internal concerns already established makes it easy to solve interoperation issues with other systems. It is a difficult task to assure interoperability, but having the listed attributes and using such mechanisms is a step towards an interoperable system.

5.2.3 Tertiary question: How to evaluate interoperability in context-aware software systems?

As seen in the first question, interoperability research in context-aware software systems and consequently its results are still in an initial stage. Regarding the second research question, seven attributes related to interoperability were presented, along with 18 interoperability mechanisms to address them. Now, in the tertiary research question that represents our interest in this work, we report the findings related to the evaluation of interoperability in such systems.

These following options were included into "**Other evaluation methods**" during the coding process as presented in Appendix A. The ones listed below were not thoroughly discussed in the works we analyzed. However, we cite them here to expand the list of evaluation options and enhance our answer to the research question.

- **Business Interoperability Quotient Measurement Model** (ZUTSHI; GRILO; JARDIM-GONCALVES, 2012)
- **INTEROP Framework** (CHEN; DOUMEINGTS; VERNADAT, 2008)
- **IDEAS Framework** (CHEN; DOUMEINGTS; VERNADAT, 2008)
- **E-health Interoperability Framework** (CHEN; DOUMEINGTS; VERNADAT, 2008)
- **European Interoperability Framework** (CHEN; DOUMEINGTS; VERNADAT, 2008)
- **Levels of Conceptual Interoperability Model** (MADNI; SIEVERS, 2013)
- **Military Communications and Information Systems Interoperability** (REZAEI *et al*., 2014)
- **Spectrum of Interoperability Model** (REZAEI *et al*., 2014)
- **Stoplight Model** (REZAEI *et al*., 2014)
- **Systems-of-Systems Interoperability Model** (ULLBERG; LAGERSTRÖM; JOHNSON, 2008; WYATT; GRIENDLING; MAVRIS, 2012)

Other methods for interoperability evaluation were found in the selected papers. The methods listed below were described in more depth, so here presented:

- **Enterprise Interoperability Maturity Model:** each area of concern would be defined by a set of objectives and goals relevant to interoperability and collaboration issues. Depending on the presence or lack of the maturity indicators, interoperability, and collaboration maturity level would be defined for each area of concern (CHEN; DOUMEINGTS; VERNADAT, 2008; REZAEI *et al*., 2014).
- **i-Score Model:** it is an architecture-based method of measuring interoperability of complex networks of non-homogeneous systems. It uses a multiplicity matrix for the pairs, where each element is evaluated as -1 (human translation necessary to be interoperable), 0 (machine translation required to be interoperable), 1 (no human or machine actions necessary to be interoperable) (ULLBERG; LAGERSTRÖM; JOHNSON, 2008; REZAEI *et al*., 2014).
- **Interoperability Assessment Methodology:** introduces nine components, which are: requirements, node connectivity, data elements, protocols, information flow, information utilization, interpretation, latency, and standards. The nine components each included either a "yes/no" response or a mathematical equation (REZAEI *et al*., 2014).
- **Levels of Information Systems Interoperability:** with five interoperability levels (0–4) are Isolated, Connected, Functional, Domain, and Enterprise, in which each

interoperability level exists in a specific environment. The attributes of the LISI Reference Model contain Procedures, Applications, Infrastructure, and Data (CHEN; DOUMEINGTS; VERNADAT, 2008; ULLBERG; LAGERSTRÖM; JOHNSON, 2008; WYATT; GRIENDLING; MAVRIS, 2012; REZAEI *et al*., 2014).

- **Organizational Interoperability Maturity Model:** this model extends the LISI model into the more abstract layers of command and control support. Describing the ability to interoperate, five levels of organizational maturity are defined in this model (independent, *ad-hoc*, collaborative, combined and unified). Preparedness, Understanding, Command style, and Ethos (REZAEI *et al*., 2014).
- **Quantification of Interoperability Methodology:** they stated "interoperability of systems, units, or forces can be factored into a set of components that can quantify interoperability" and they identified the seven necessary components as languages, standards, environment, procedures, requirements, human factors, and media (WYATT; GRIENDLING; MAVRIS, 2012; REZAEI *et al*., 2014).

Despite finding 16 different evaluation methods, they do not address the context information. However, two initiatives attempt to tackle this issue:

- Fang *et al.* (2004) present an interesting proposal on dealing with evaluation, as they consider that a quantifiable measure of interoperability will be more attractive and concrete than simple judgment, i.e. yes or no. In that matter, it assesses the signature level (syntax consistency of service description), protocol level (order in which services are invoked) and semantic level (common understanding of the service’s function) by fuzzy interoperability quantization. Additionally, the quality and context levels are considered. The limitations of this work are the focus on the service composition and do not provide any experimental study to observe the proposal application. Despite considering the context, this level is not assessed because it is a work in progress.
- Gjoreski *et al.* (2015) present a laboratory experiment, which evaluated eight software systems in the recognition of seven activities representing an elderly person’s lifestyle and provides the context for smart control of home automation. Interoperability represented 15% of the overall score, assessed through a questionnaire form. They considered the use of an API for integrating the systems, available documentation, sample application, open source code and use of any well-known application-level protocol.

Despite the efforts made in this research area with the settings of this study, we could not retrieve an evaluation method designed for context-aware systems or that takes context information into account while assessing interoperability. From the data, we analyzed it was possible to outline some alternatives for interoperability evaluation in context-aware software systems.

One option is to use the theoretical framework as a guide for assessment. While answering questions regarding the **purpose**, **perspective**, **levels** and the other concepts related to a given interaction, one can check for the consistency between what is ideally desirable and what is in actual use.

Another way to evaluate can be extending some of the existing evaluation models to consider the context information and the infrastructure required to acquire, store, control and use the information. This alternative compares existing models regarding the seven attributes and 18 interoperability mechanisms presented in the secondary question.

Even though none of the selected works answered this research question, considering the interoperability concepts organized in this research and the proposed alternatives pointed out herein we believe we can provide a suitable solution for the evaluation issue which configures an opportunity for future work. Therefore, this research question requires further investigation.

5.3 Chapter Considerations

In this chapter we considered the importance of interoperability, and according to our characterization goal after conducting a secondary study and analyzing the findings with GT we answer the following research questions: a) How is interoperability addressed in context-aware software systems? b) Which are the interoperability concepts and attributes used in context-aware software systems? c) How to evaluate interoperability in context-aware software systems?

In the first question, we introduce our definition of interoperability, grounded in the data analyzed and how it covers other known interoperability definitions. Was observed the comprehension of what is context and how it is used in this systems domain, by presenting the context information found. In the second question, it is presented seven attributes and 18 interoperability mechanisms, some that are general concepts, which can relate to every software system, and particular concepts, which are domain specific ones. Only a few papers contributed to the last question, showing a research gap that needs to be considered aiming the progress of the area and interoperable software systems. The area is recent and has much development to achieve. We hope to contribute to further investigations into the topic.

Due to a large amount of data generated after the analysis is a challenge to fit the knowledge generated into the answers to the research questions, at the risk of not including some concepts or not be very detailed. Another point to note is that due to the nature of the questions, the answers may not be unique or direct. The answers provided were based on the review performed with the retrieved data later analyzed and interpreted. Therefore, the responses show the understanding of the author of this work. However, limitations are known as the threats to validity presented for the activities (review and analysis) performed.

6 Conclusion

The conclusion of this work is presented in this chapter. We outline the main contributions of this research and some limitations. Finally, some possible paths for future work are described to continue the research here presented.

6.1 Introduction

This work presented the research on interoperability in context-aware software systems, detailing the activities performed to achieve the discussed concepts. Considering that software systems should be able to interact with different actors, accomplish their purposes, to complete their tasks and act according to the context, regardless of their differences in development or organization, interoperability comes as a challenge.

A secondary study, classified as a quasi-Systematic Literature Review, was conducted to observe how interoperability is addressed in such systems. Nevertheless, the review was also interested in finding interoperability concepts, attributes and evaluation methods. Based on the findings, principles of Grounded Theory were applied to analyze, organize and relate the concepts. The questions that drive this research were later answered with the information from the qualitative analysis.

This chapter presents the contributions of this dissertation, also showing its limitations. In the Future Work Section possibilities of development are described.

6.2 Contributions

The main contributions of this work are as follows:

- It contributes to one of the interoperability challenges presented in Chapter 2 as it extends the interoperability concept. The new definition proposed is grounded in data and is not limited to technologies. It also organizes others relating concepts enriching the comprehension of interoperability as well contributing to the extension of the interoperability concept.
- It provides the *quasi*-Systematic Literature Review protocol, available online as a technical report, for other researchers interested in expanding or replicating a review addressed to interoperability in context-aware software systems.

- It organizes the state-of-the-art of this research area, presented as the initial results of the *quasi*-Systematic Literature Review making the information summarized.
- The organization of a body of knowledge regarding interoperability in context-aware software systems presented as a theoretical framework. This contribution enhances the understanding of the relationship of the concepts related to interoperability and the context information observed in the data. Having the research findings organized in such manner can stimulate research and lead to the extension of knowledge. The body of knowledge and theoretical framework will contribute to the CAcTUS Project, being this work part of a wider research with the purpose to search for solutions to the context-aware systems domain.
- It answers the research questions providing information on how interoperability is addressed in context-aware software systems, which interoperability concepts and attributes are used and an overview of how to evaluate interoperability in such systems.

6.3 Limitations

The boundaries of this work are as follows:

- The findings of the *quasi*-Systematic Literature Review provide the basis for this research, and as limitations of this work are the validity threats presented in Chapter 3. We highlight some of these treats such as a) the possibility to miss some studies because of the database selection, the string used and the inclusion criteria; b) researcher bias regarding data extraction. We point out these threats as limitations of this work because it may impair the analysis conducted since the findings, and the theoretical framework, relies on the data from the literature review.
- The use of Grounded Theory and the procedures executed have limitations on their own, also presented as validity threats in Chapter 4. We highlight some of these treats such as a) the ones related with to interpretation bias and information abstraction during the coding process that directly affects the results of this work; b) concerned if the conclusions are reasonable and based on data. The applicability of the results, at the moment, is limited to the data analyzed, so it is necessary to confirm its validity with additional data.
- The set of concepts, attributes, and interoperability mechanisms are results of the qualitative analysis and were revised by another researcher and represent the interpretation of this author of the information retrieved. However, from the

perspective of others researchers, the findings may not be adequate to characterize interoperability or considered incomplete.

- Due to issues related to research time, no experimental studies to assess the validity of the proposed framework were performed, neither a refinement with additional research data.

6.4 Future Work

The possibilities of future work are as follows:

- There are other challenges presented in Chapter 2 such as spontaneous interoperability and evaluation. Spontaneous interoperability was coded as a research challenge (Appendix A – Section A). Addressing the challenges retrieved from technical literature and organized in this work is a possibility of development of this investigation that requires further investigation. Regarding interoperability evaluation, it was the tertiary question of this work that was left open. Despite the effort to present the current evaluation methods retrieved from the technical literature, they do not address the context information as presented in this work. Following the proposed alternatives for evaluation previously presented is another option for future work.
- The extension or re-execution of the *quasi*-Systematic Literature Review is an option to refine the theoretical framework by adding more data to be analyzed. A different approach for the refinement is that other researchers follows the coding process and compare the findings with the ones presented here. Another option is to use theoretical sampling to gather data driven by the concepts found to unfold variations or enhance the concepts.
- A deeper analysis can be conducted specific to the relationships described in Chapter 4. Considering strong concepts and relationships, based on the groundedness and density, exploring how the relationships are built, the key factors that influence them and how are some of the possible paths of research.
- Execute experimental studies to assess the validity of the proposed framework. With an extension of this work, it would be interesting to observe how the concepts and relationships behave with real system data. That can bring more reliability in the proposal and provide input for improvements.
- The concepts, attributes and interoperability mechanisms presented in this work can be organized and made available as a tool or meta-model to support the decision-making in two moments: a) during the planning phase of systems of this nature, acting as a guide for the development; b) also later for integration testing to confirm the adequacy of the interaction between the systems.

6.5 Final Considerations

This last chapter presents a summary of the activities performed and the contributions of this work. The limitations and possibilities of future work are also presented. With all the technological development of the later years, some can say that the future is not near, is here (GREENFIELD, 2010). Nevertheless, as presented in this work, there are still challenges to be overcome. Throughout this research, we could observe that the research area of context-aware software systems is recent presenting several possibilities of research to be conducted. Also, the interoperability field is comprehensive and cannot be treated as a little research. We hope that with the body of knowledge organized in this work and the possibilities of future work pointed out in this chapter we contribute to the progress in both fields of research.

References

ABOWD, G. D. Software engineering issues for ubiquitous computing. **Proceedings of the 1999 International Conference on Software Engineering**, p. 75–84, 1999.

ADOLPH, S.; HALL, W.; KRUCHTEN, P. A methodological leg to stand on: Lessons learned using grounded theory to study software development. **Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds**, **ACM**, p. 13, 2008.

AITENBICHLER, E. *et al*. Engineering intuitive and self-explanatory smart products. **Proceedings of the 2007 ACM symposium on Applied computing**, p. 1632-1637, 2007.

ATHANASOPOULOS, G.; TSALGATIDOU, A.; PANTAZOGLOU, M. Interoperability Among Heterogeneous Services, **IEEE International Conference on Services Computing,** p. 174-181, 2008.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.

BADREDDIN, O. Thematic Review and Analysis of Grounded Theory Application in Software Engineering. **Advances in Software Engineering**, v. 2013, n. OCTOBER, p. 1–9, 2013.

BALDAUF, M.; DUSTDAR, S.; ROSENBERG, F. A survey on context-aware systems. **International Journal of Ad Hoc and Ubiquitous Computing**, v. 2, n. 4, p. 263-277, 2007.

BASILI, V. R. Software modeling and measurement: the Goal/Question/Metric paradigm. **Tech. Rep. CS-TR-2956, Department of Computer Science, University of Maryland, College Park,** MD 20742, 1992.

BAZIRE, M.; BRÉZILLON, P. Understanding context before using it. **Modeling and using context**, Springer Berlin Heidelberg, n. November, p. 29–40, 2005.

BECKETT, D.; MCBRIDE, B. **RDF/XML SyntaxSpecification W3C Recommendation,** 2004. Available at: http://www.w3.org/TR/REC-rdf-syntax/. Accessed in: April 2016.

BIOLCHINI, J. C. et al. Scientific research ontology to support systematic review in software engineering. **Advanced Engineering Informatics**, v. 21, n. 2, p. 133–151, 2007.

BLAIR, G.; BENCOMO, N.; FRANCE, R. B. Models@ run.time. **Computer**, v. 42, n. 10, p. 22–27, 2009.

BURRELL, J.; GAY, G. K. E-graffiti: Evaluating real-world use of a context-aware system. **Interacting with Computers**, v. 14, n. 4, p. 301–312, 2002.

C4ISR ARCHITECTURE WORKING GROUP. Levels of information systems interoperability (LISI). **U.S. Department of Defense Report.**, v. 30, n. March, 1998.

CARVER, J. The Use of Grounded Theory in Empirical Software Engineering. **Empirical Software Engineering Issues. Critical Assessment and Future Directions.** Springer Berlin Heidelberg, p. 42, 2007.

CHAPMAN, C.; LAHAV, M.; BURGESS, S. Digital pen: Four rounds of ethnographic and field research. **HICSS'09. 42nd Hawaii International Conference on System Sciences**, p. 1–10, 2009.

CHEN, D. Enterprise Interoperability Framework. **Interoperability for Enterprise Software and Applications**, I-ESA05, 77–88. 2005.

CHEN, D.; DOUMEINGTS, G.; VERNADAT, F. Architectures for enterprise integration and interoperability: Past, present and future. **Computers in Industry**, v. 59, n. 7, p. 647–659, 2008.

CHEN, G.; KOTZ, D. A Survey of Context-Aware Mobile Computing Research. **Technical Report TR2000-381, Dartmouth Computer Science**, p. 1–16, 2000.

CHEVERST, K. et al. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM**, p. 17–24, 2000.

CRUZES, D., DYBÅ, T. Research synthesis in software engineering: A tertiary study. **Information and Software Technology**, v. 53, n. 5, p. 440-455, 2011.

CONSOLVO, S.; ARNSTEIN, L.; FRANZA, B. User study techniques in the design and evaluation of a ubicomp environment. **Proceedings of the 4th International Conference on Ubiquitous Computing**, p. 73–90, 2002.

CONTE, T., CABRAL, R., TRAVASSOS, G. Aplicando Grounded Theory na Análise Qualitativa de um Estudo de Observação em Engenharia de Software–Um Relato de Experiência. In: **V Workshop - Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)**. p. 26-37, 2009.

COSTA, C. A. DA; YAMIN, A. C.; GEYER, C. F. R. Toward a General Software Infrastructure for Ubiquitous Computing. **IEEE Pervasive Computing,** n. 1, p. 64-73, 2008.

DAMAS, M. et al. Ambient assisted living devices interoperability based on OSGi and the X73 standard. **Telemedicine journal and e-health : the official journal of the American Telemedicine Association**, v. 19, n. 1, p. 54–60, jan. 2013.

DESPONT-GROS, C. et al. The Digital Pen and Paper Technology: Implementation and Use in an Existing Clinical Information System. **Studies in health technology and informatics**, v. 116, n. FEBRUARY 2005, p. 328–333, 2005.

DEY, A.; ABOWD, G.; SALBER, D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. **Human-Computer Interaction**, v. 16, n. 2, p. 97–166, 2001.

DEY, A. K. Understanding and using context. **Personal and Ubiquitous Computing**, v. 5, n. 1, p. 4–7, 2001.

DEY, A. K.; ABOWD, G. D. Towards a Better Understanding of Context and Context-Awareness. **Computing Systems**, v. 40, n. 3, p. 304–307, 1999.

DEY, I. Grounding Grounded Theory: Guidelines for Qualitative Inquiry. **San Diego, CA, Academic Press**, 1999.

EDWARDS, W. K.; GRINTER, R. E. At Home with Ubiquitous Computing: Seven Challenges. **Proceedings of the 3rd International Conference on Ubiquitous Computing**, p. 256–272, 2001.

FANG, J.; HU, S.; HAN, Y. A service interoperability assessment model for service composition. **IEEE International Conference on Services Computing**, p. 153-158, 2004.

FAVELA, J. et al. Integrating Context-Aware Public Displays Into a Mobile Hospital Information System. **IEEE Transactions on Information Technology in Biomedicine**, v. 8, n. 3, p. 279–286, 2004.

FLORES, A. et al. Towards context-aware testing for semantic interoperability on PvC environments. **IEEE International Conference on Systems, Man and Cybernetics**, v. 2, p. 1136-1141, 2004

FORD, T. et al. A Survey on Interoperability Measurement. **International Command and Control Research and Technology Symposium**, n. 937, 2007.

GJORESKI, H. et al. Competitive Live Evaluations of Activity-Recognition Systems. **IEEE Pervasive Computing**, v. 14, n. 1, p. 70–77, 2015.

GLASER, B. G.; STRAUSS, A. L. The Discovery of Grounded Theory: Strategies for Qualitative Research. **Transaction Publishers**, 2009.

GOULARTE, R.; PIMENTEL, M. DA G. C.; MOREIRA, E. DOS S. Context-aware support in structured documents for interactive-TV. **Multimedia Systems**, v. 11, n. 4, p. 367–382, 2006.

GRACE, P.; FLORES, C.; BLAIR, G. Emergent middleware: rethinking interoperability for complex pervasive systems. **Distributed Computing**, p. 30:1–30:2, 2009.

GREENFIELD, A. Everyware: The dawning age of ubiquitous computing. **New Riders**, 2010.

GUILLEMIN, P. *et al*. Internet of things strategic research roadmap. **The Cluster of European Research Projects, Tech. Rep**, 2009.

HONG, J.; SUH, E.; KIM, S.-J. Context-aware systems: A literature review and classification. **Expert Systems with Applications**, v. 36, n. 4, p. 8509–8522, 2009.

IEEE. IEEE Standard Glossary of Software Engineering Terminology, **IEEE Computer Society**, v. 610121990, n. 121990, p. 3, 1990.

ISO 14258. Industrial Automation Systems—Concepts and Rules for Enterprise Models. **ISO TC 184 SC5**, **International Organization for Standardization**, Geneva, Switzerland, 1999.

ISO/IEC WD. **Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) –** Measurement of system and software product quality, 2011.

JEON, N. J. et al. A Taxonomy of Ubiquitous Computing Applications. **Wireless Personal Communications**, v. 43, n. 4, p. 1229–1239, 2007.

JIMENEZ, H.; MAVRIS, D. N. Characterization of Technology Integration Based on Technology Readiness Levels. **Journal of Aircraft**, v. 51, n. 1, p. 291–302, jan. 2014.

JUÁREZ, J.; RODRÍGUEZ-MONDÉJAR, J. A.; GARCÍA-CASTRO, R. An ontology-driven communication architecture for spontaneous interoperability in Home Automation systems. **19th IEEE International Conference on Emerging Technologies and Factory Automation**, 2014.

KIDD, C.; ORR, R.; ABOWD, G. The aware home: A living laboratory for ubiquitous computing research. **Cooperative buildings. Integrating information, organizations, and architecture. Springer Berlin Heidelberg**, p. 191–198, 1999.

KIENTZ, J. A. et al. Where's My Stuff? Design and Evaluation of a Mobile System for Locating Lost Items for the Visually Impaired. **Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility**, p. 103, 2006.

KIENTZ, J. A.; JONES, B. The Georgia Tech Aware Home. **CHI'08 extended abstracts on Human factors in computing systems**, p. 3675–3680, 2008.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. TR/SE-0401, p. 28, 2004.

KOUSSOURIS, S.; LAMPATHAKI, F. Digging into the real-life enterprise interoperability areas definition and overview of the main research areas. **Proceedings of Collaborative Enterprises – Platforms, Processes, and Practices Advancing the Enterprise 2.0**, p. 19–22, 2011.

KUTVONEN, L. Enhancing the maturity of open service ecosystems and inter-enterprise collaborations. **Enterprise Interoperability. Springer Berlin Heidelberg**, p. 6-21, 2013.

KUTVONEN, L.; RUOKOLAINEN, T.; METSO, J. Interoperability Middleware for Federated Business Services in Web-Pilarcos. **International Journal of Enterprise Information Systems**, v. 3, n. 1, p. 1–21, 2007.

KUZIEMSKY, C. E.; WEBER-JAHNKE, J. H. An eBusiness-based Framework for eHealth Interoperability. **Journal of Emerging Technologies in Web Intelligence**, v. 1, n. 2, p. 129-136, 2009.

LAMPATHAKI, F. et al. Business to business interoperability: A current review of XML data integration standards. **Computer Standards & Interfaces**, v. 31, n. 6, p. 1045–1055, 2009.

LEE, S.; CHANG, J. Survey and Trend Analysis of Context-Aware Systems. **Information-An International Interdisciplinary Journal**, v. 14, n. 2, p. 527–548, 2010.

LEEM, C. S. et al. A Business Model (BM) development methodology in ubiquitous computing environments. **Computational Science and Its Applications**. Springer Berlin Heidelberg, p. 86-95, 2005.

LEITE, M. J. Interoperability assessment. **Proceedings of the 66th Military Operations Research Society Symposium. Monterey, CA**, 1998.

LOUREIRO, A. A. F. et al. Computação Ubíqua Ciente de Contexto: Desafios e Tendências. **27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, p. 99–149, 2009.

MADNI, A. M.; SIEVERS, M. System of Systems Integration : Key Considerations and Challenges. **Systems Engineering**, v. 17, n. 3, p. 330-347, 2014.

MATALONGA, S.; RODRIGUES, F.; TRAVASSOS, G. Challenges in Testing Context Aware Software Systems. **9th Workshop on Systematic and Automated Software Testing**, p. 51 2015.

MATTERN, F. The Vision and Technical Foundations of Ubiquitous Computing. **Upgrade, CEPIS**, v. 2, n. 5, p. 2–6, 2001.

MCBRYAN, T.; MCGEE-LENNON, M. R.; GRAY, P. An Integrated Approach to Supporting Interaction Evolution in Home Care Systems, **Proceedings of the 1st International Conference on Pervasive Technologies Related to Assistive Environments**. ACM. p. 45, 2008.

MCGUINNESS, D.; HARMELEN, F. OWL WebOntology Language Overview. **W3C Recommendation.** Available at: http://www.w3.org/TR/owl-features/. Accessed in: April 2016.

MEMON, M. et al. Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. **Sensors, Basel, Switzerland**, v. 14, n. 3, p. 4312–41, jan. 2014.

MOTTA, R. et al. qSLR Protocol - Investigating Interoperability in Context-Aware Systems. Available at: http://lens.cos.ufrj.br/cactus/?q=node/15. Accessed in: April, 2016.

NAKADA, T.; KANAI, H.; KUNIFUJI, S. A support system for finding lost objects using spotlight. **Proceedings of the 7th international conference on Human computer interaction with mobile devices & services**, p. 321, 2005.

NAUDET, Y. et al. Towards a systemic formalization of interoperability. **Computers in Industry**, v. 61, n. 2, p. 176–185, fev. 2010.

NIEMELÄ, E.; LATVAKOSKI, J. Survey of requirements and solutions for ubiquitous software. **Proceedings of the 3rd International Conference on Mobile and ubiquitous multimedia MUM 04**, n. October, p. 71–78, 2004.

O'NEILL, E. et al. A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments. **First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities**, p. 60–69, 2005.

PAI, M. et al. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. **Natl.Med J India**, v. 17, n. 2, p. 86–95, 2004.

PASCOE, J.; RYAN, N.; MORSE, D. Issues in developing context-aware computing. **Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing**, v. 1707, p. 208–221, 1999.

QUEIRÓS, A. et al. Usability, accessibility, and ambient-assisted living: a systematic literature review. **Universal Access in the Information Society**, v. 14, n. 1, p. 57–66, 5 out. 2013.

REYNOLDS, M.; MEMBER, S. Telemonitoring Systems Interoperability Challenge: An Updated Review of the Applicability of ISO/IEEE 11073 Standards for Interoperability in Telemonitoring. **29th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society**, p. 6161–6165, 2007.

REZAEI, R. et al. Interoperability evaluation models: A systematic review. **Computers in Industry**, v. 65, n. 1, p. 1–23, 2014.

ROCHA, L. S. et al. Ubiquitous software engineering: Achievements, challenges and beyond. **Proceedings - 25th Brazilian Symposium on Software Engineering,** p. 132–137, 2011.

RODRIGUES, F.; MATALONGA, S.; TRAVASSOS, G. H. Systematic literature review protocol: Investigating context-aware software testing strategies. Available at: www.cos.ufrj.br/~ght/cactus_pr012014.pdf. Accessed in: April 2016.

RODRÍGUEZ-DOMÍNGUEZ, C. et al. A communication model to integrate the Request-Response and the Publish-Subscribe paradigms into ubiquitous systems. **Sensors, Basel, Switzerland**, v. 12, n. 6, p. 7648–68, Jan. 2012.

RODUNER, C. et al. Operating appliances with mobile phones: strengths and limits of a universal interaction device. **Pervasive Computing. Springer Berlin Heidelberg**, p. 198–215, 2007.

ROSE, K.; ELDRIDGE, S.; CHAPIN, L. The Internet of Things (IoT): An Overview–Understanding the Issues and Challenges of a More Connected World. **Internet Society**, Tech. Rep., October 2015.

RUIZ, A. et al. Addressing interoperability in multi-organisational Web-based systems. **Workshop on Object Interoperability**, 2000.

RUOKOLAINEN, T.; KUTVONEN, L. Interoperability in Service-Based Communities. **Business Process Management Workshops. Springer Berlin Heidelberg**, v. 68, p. 317–328, 2006.

RYAN, N. Smart environments for cultural heritage. **24th International**

Research Symposium, International Research Center for Japanese Studies, p. 17-33, 2005.

SAKAMURA, K. The Tron Project. **IEEE Micro**, v. 7, n. 2, p. 8–14, 1987.

SANTOS, I. S.; ROCHA, L. S.; ANDRADE, R. M. C. Systematic Literature Review Protocol: Investigating Test Case Design For Context-Aware Applications. Available at: https://sites.google.com/site/ismaylesantos/test-context-aware-app-sr. Accessed in: April, 2016.

SCHILIT, B. N.; THEIMER, M. M. Disseminating Active Map Information to Mobile Hosts. **IEEE Network**, v. 8, n. 5, p. 22–32, 1994.

SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. **IEEE Transactions on Software Engineering**, v. 25, n. 4, p. 557–572, 1999.

SPÍNOLA, R.; MASSOLLAR, J.; TRAVASSOS, G. H. Checklist to Characterize Ubiquitous Software Projects. **XXI Simpósio Brasileiro de Engenharia de Software**, n. 1991, p. 39–55, 2007.

SPÍNOLA, R. O. Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software. **Universidade Federal do Rio de Janeiro**, 2010.

SPÍNOLA, R. O.; PINTO, F. C. R.; TRAVASSOS, G. H. Supporting requirements definition and quality assurance in ubiquitous software project. **Leveraging Applications of Formal Methods, Verification, and Validation: Communications in Computer and Information Science**. Springer Berlin Heidelberg, v. 17, p. 587-603, 2009.

STRANG, T.; LINNHOFF-POPIEN, C. Service interoperability on context level in ubiquitous computing environments. **Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet**. 2003.

STRAUSS, A.; CORBIN, J. Basics of qualitative research: Techniques and procedures for developing grounded theory. **Sage publications**, 1998.

SULLIVAN, D. O.; LEWIS, D. Semantically Driven Service Interoperability for Pervasive Computing. **Proceedings of the 3rd international workshop on Data engineering for wireless and mobile access**. ACM, p. 17–24, 2003.

TANG, L. et al. Supporting rapid design and evaluation of pervasive applications: Challenges and solutions. **Personal and Ubiquitous Computing**, v. 15, n. 3, p. 253-

269, 2011.

TRAVASSOS, G. H. et al. An Environment to Support Large Scale Experimentation in Software Engineering. **13th IEEE International Conference on Engineering of Complex Computer Systems**, p. 193–202, 2008.

ULLBERG, J.; LAGERSTRÖM, R.; JOHNSON, P. A Framework for Service Interoperability Analysis using Enterprise Architecture Models. **2008 IEEE International Conference on Services Computing**, p. 99–107, jul. 2008.

VEGA-BARBAS, M. et al. Smart Spaces and Smart Objects Interoperability Architecture (S3OiA). **2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing**, p. 725–730, jul. 2012.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Designing context-sensitive systems: An integrated approach. **Expert Systems with Applications**, v. 38, n. 2, p. 1119–1138, 2011.

VILJANEN, L.; RUOHOMAA, S.; KUTVONEN, L. The {TuBE} approach to trust management. **Proceedings of the 3rd iTrust internal workshop**, 2004.

WANT, R.; PERING, T. System challenges for ubiquitous & pervasive computing. **Proceedings of the 27th International Conference on Software engineering**, p. 9, 2005.

WEISER, M. The Computer for the 21st Century. **Scientific American**, 1991.

WEISER, M. Ubiquitous computing. **IEEE Computer**, 1993.

WYATT, E. J.; GRIENDLING, K.; MAVRIS, D. N. Addressing Interoperability in Military Systems-of- Systems Architectures. **Systems Conference, IEEE International**, p. 1-8. 2012.

ZDRAVKOVIĆ, M.; NORAN, O.; TRAJANOVIĆ, M. Interoperability as a property of ubiquitous healthcare systems. **IFAC Proceedings Volumes**, v. 19, p. 7849–7854, 2014.

ZUTSHI, A.; GRILO, A.; JARDIM-GONCALVES, R. The Business Interoperability Quotient Measurement Model. **Computers in Industry**, v. 63, n. 5, p. 389–404, jun. 2012.

APPENDIX A – Systematic Literature Review Protocol: Investigating Interoperability in Context-Aware Systems

A.1 Research Protocol

The need for this review was commissioned inside the CAcTUS project, and an initial plan decided with basic procedures before conducting it. In this perspective, this review concerns to observe interoperability in context-aware systems as a preliminary step to further project activities.

This protocol was based on Biolchini et al. (2005). To organize and structure the search string, it explores the **PICO** strategy (Pai et al., 2004). This approach organizes the search words into four groups: **P**opulation of interest, **I**ntervention or exposure being evaluated, **C**omparison intervention (if applicable) and **O**utcome. Because of the objective of the study (mainly characterization), it will not be possible to apply any comparison. Therefore, we can classify this study as *quasi*-systematic literature review (Travassos et al., 2008).

A.1.1 Question Focus

This study objective is to observe how interoperability is addressed in context-aware systems and identify the different aspects and attributes of interoperability in context-aware systems. Eventually, measures to evaluate them.

A.1.2 Question Quality and Amplitude

- **Problem:** With widespread technology and interconnection of different systems and devices, interoperability has become critical. In context-aware systems, it is essential that various types of devices and services interact automatically without user intervention. Therefore, interoperability is an important feature in this kind of system.
- **Question:**
    - **Main Question:**
      How is interoperability addressed in context-aware systems?
    - **Secondary Questions:**
      Which are the interoperability attributes used in context-aware systems?
      How to evaluate interoperability in context-aware systems?
- **Population:** Context-aware Systems
- **Intervention Control:** Interoperability

- **Comparison:** None
- **Outcome:** Definition, Attributes, and Evaluation Methods
- **Experimental Design:** None statistical method is going to be applied.
- **Control Articles:** There were no control articles for this protocol.

A.1.3 Source Selection

- **Sources Selection Criteria Definition:** Works presented as articles, available on the web.
- **Studies Language:** English.
- **Source Identification**
  - **Source Search Method:** Manual search through web search engine.
  - **Search Engine:** Scopus, at http://www.scopus.com/, chosen because it covers the majority of software engineering publications and due to previous experience of the authors in conducting systematic literature reviews. The search considers title, abstract and keywords for an article to be selected.
  - **JabRef [12] reference tool:** The results of the search in Scopus are exported in a BibTeX format with all the available information. The reviewers use JabRef tool to work with the generated document for the studies selection described below.

A.1.4 Studies Selection

- **Studies Inclusion and Exclusion Criteria Definition:**

  Inclusion Criteria:

  - (To talk about interoperability OR interconnection OR interoperation OR interaction OR integration OR exchange) AND
  - (The discussed system including context-awareness characteristics; OR To be applied to Context-Aware Systems).

  Exclusion Criteria:

  - Not being available; AND
  - Published earlier than 1991[13]; AND
  - Studies in duplicity; AND
  - Register of proceedings.
- **Study type definition:** Articles that presents an example, application, proof of concept or study related to interoperability in context-aware systems.

[12] http://jabref.sourceforge.net/
[13] Considering the seminal work of Weiser, M. (1991).

- **Procedures for Studies Selection:** Using JabRef tool, read title and abstract of each retrieved study and evaluate it according to inclusion and exclusion criteria.
- **Acceptance Criteria:** Four distinct readers evaluated each study. The discussion scenery involves the readers expressing the rationale behind the choice and reach a consensus to include or exclude the study under discussion. The studies acceptance criteria happened as described in Table 3.3.

A.1.5 Data Extraction

For each selected paper the following information shall be extracted:

Table A.1. Information about the extraction fields

Field		Description
Reference information	Title	The title of the paper.
	Authors	List of authors, including email addresses and affiliation.
	Year of Publication	The year the paper was published.
	Source of Publication	Name of the Journal, Conference or place where the paper was published.
Abstract		The complete abstract of the paper.
Interoperability definition		The definition used in the paper.
Interoperability dimension		The range variation of interoperability (e.g. Semantic, Syntatic…) - As presented in the article - *No taxonomy*
Applied Interoperability		Domain where interoperability is considered in the paper (e.g. Business Interoperability, Service Interoperability…) - As presented in the article - *No taxonomy*
Interoperability attribute and theoretical basis		Name or list of interoperability attributes presented in the paper and how were derived.
Evaluated interoperability attribute		Name or list of interoperability attributes evaluated in the presented technique / article
Interoperability measures or method		Type and value of the measurement system used to evaluate the presented interoperability attributes, or the model or method used to evaluate interoperability
Pre-existent approach		The previous approach if the article presents an adaptation or evolution
Conditions or Restrictions		Limitations for the applicability
Software Systems category		Software Category according to Pressman 2010. See Annex I - Types of Software System Category.
Development Target		Main objective or target - As presented in the article - *No taxonomy*
Type of experimental study		Types of empirical studies according to Wynekoop and Conger with definitions taken from ESE Wiki. See Annex II – Types of Experimental Study Taxonomy.
Experimental Study Data		Include any information regarding design, variables and threats if available. Also, which interoperability evaluation method and attributes were used.

A.1.6 Paper Quality Evaluation Criteria

The following criteria will be used to evaluate the quality of the selected papers. It aims to highlight those papers that could be more related to the investigation theme and, consequently, strengthen the confidence in the final result.

- **Criteria related to interoperability concepts:**
    1. Is there any interoperability definition? (1 pt)
    2. Is there any description of the interoperability dimension?(1 pt)
    3. Is there any description about the interoperability application? (1 pt)
    4. Is there any definition regarding interoperability attributes (i.e. security, reliability, and so on)? (0.5 pt) Is there any description about how the interoperability attributes have been derived? (0.5 pt)
- **Criteria related to interoperability evaluation:**
    9. Is the interoperability evaluation described? (1 pt)
    10. Does the interoperability evaluation include the proposed attributes? (1 pt)
    11. Is there an empirical evaluation of the interoperability approach? (1 pt)
- **Criteria related to the background theory or applicability:**
    19. Does the paper describe any adaptation/evolution of pre-existent interoperability approach? (1pt)
    20. Is there any description of restrictions and conditions about the applicability of the interoperability approach? (1 pt)
    21. Is it possible to identify for which types of system can the interoperability approach be used? (1 pt)
- **Criteria related to the interoperability approach generalization:**
    11. Is there any description of the interoperability approach application in additional settings? (0.5 for each setting)

A.2 Execution and Results

A.2.1 Execution

This protocol has evolved iteratively based on the research progress, and we try to report here the evolution steps. It is also important to remind, for replicability purposes, that the used strings are according to the Scopus search engine - another engine might have different structures and rules.

Trial 0: Ad-hoc -Executing the initial search string on the Scopus.

Process: Despite being *ad-hoc*, some structure was used based on a previously conducted literature review in the project (Rodrigues et al., 2014). Using the search string presented below, the objective was to find papers related to interoperability measures in the context-aware application, to form a golden standard. The participants read title and abstracts from the papers, in a subjective and particular way applying a personal inclusion/exclusion criteria. Each disagreement was followed by a discussion – putting all the readers in the same level of understanding.

Search String applied:

TITLE-ABS-KEY(("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR pervasive OR ubiquitous OR usability OR "event based" OR "self adaptive" OR "self adapt") AND (interoperability) AND (metric OR measure OR "quality attributes" OR "quality characteristics"))

Date: December 2014.

Number of Articles Found: 89

Number of Articles for Full Reading: 22 (2 were not available)

Results: Two readers divided the returned papers for a full reading. It was possible to observe the difference between what we want to find (concepts and aspects that describe interoperability in context-aware systems) from the returned results (interoperability used as a *buzz* word). In this trial, it was not possible to retrieve a paper that could be used as a control. Due that, it has been decided to extend the search, to see how interoperability is addressed in context-aware systems and how to evaluate it, and not only concepts and aspects. This first trial contributed to better alignment of the readers' perspective, but the articles were not included in the final list.

Trial 1: String tuning

Process: From the papers in Trial 0, some keywords and synonyms (based on the original string) were included in a list. Also, it was carried out a search for others systematic or structured reviews – not limited to any specific field – regarding interoperability, with the objective to find used terms that might apply[14]. These terms were compared with those in the list and included if suitable. After, each term was submitted to a search engine separately to confirm its relevance considering the

[14] Process: Conducted in March, 2015, used the Google Scholar as search engine and considered results until page 5. Searched for: systematic review, literature review, systematic literature review, structured literature review, structured review and systematic mapping, for the terms: interoperability, integration, interoperation – separately. Were only considered papers were the search string was presented and when relevant for our perspective. The list of considered papers is presented in Appendix A.

expected results. For this, the **population** was fixed at all times. When the term considered was part of the **intervention**, the **outcome** remained fixed as well - and vice-versa. If relevant, the term was kept in the search string. Otherwise, the term was discarded.

Search String:

TITLE-ABS-KEY(("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR pervasive OR ubiquitous OR usability OR "event based" OR "self adaptive" OR "self adapt") AND (interconnection OR interoperation OR interaction OR interoperability OR interoperable OR integrate OR integration OR connection OR exchange OR interchange) AND (metrics OR evaluations OR assessments OR characteristics OR measurements OR models OR guidelines OR approaches OR methods OR methodology OR process))

Date: 23th March 2015.

Number of Articles Found: 31,922

Results: The results of this string coverage a wide range and different areas, but low precision regarding our research. So was decided that before going to selection, the string should be revised, as shown in Trial 2. For this reason, we do not show here the included or read articles.

Trial 2: String review

Process: The string of Trial 1 was revised by researchers in the software engineering field, with experience in systematic reviews. The protocol, string, and results were presented for the team. It was an open discussion, and several suggestions were made. After the considerations and adjustments, the resulting string is provided below.

Search String:

TITLE-ABS-KEY(("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR pervasive OR ubiquitous OR "event based" OR "self adaptive" OR "self adapt" OR "ambient intelligence" OR "assisted living" OR "agents systems" OR "multiagent systems") AND (interoperability OR interconnection OR interoperation OR interaction OR integration OR exchange) AND ("evaluation metric" OR "evaluation method" OR "evaluation model" OR "evaluation process" OR "evaluation methodology" OR "evaluation criteria" OR "evaluation approach" OR "evaluation strategy" OR "measurement method" OR "measurement model" OR "measurement process" OR "assessment method" OR "assessment model" OR "assessment strategy" OR "quality attributes" OR "quality properties" OR "quality characteristics" OR "quality features"))

Date: 25th April 2015.

Number of Articles Found: 257

Number of Articles Included in the final list: 12

Results: This trial was carried according to the process described in Section 3. From the 257 articles, 3 were duplicates and 34 registers of proceedings, hence discarded. Four readers evaluated the remaining 220 following the criteria presented in section 2.4.1. After this review, 20 candidates articles were kept for a full reading and data extraction according to section 2.6 performed by one of the readers, and then the articles were equally divided and the extractions revised among the other readers for consensus. The remaining 15 articles were to be included in this review, but one of them[15] was not recovered. After contact with one of the authors an already included paper from the same authors was received, for that reason, only one was included – and it is presented as article 7 in the list of included papers in section 3.2.

Trial 3: Including “software systems” and “systems of systems” terms

Process: During Trial 2 execution some articles were found considering “assisted living” systems in “systems of systems” and other works relating interoperability evaluation models with “software systems”. Therefore these terms were included aiming to recover papers that could contribute to the research considering this perspective. We do not consider these new terms as synonyms of the initial target population (context aware systems), but to extend the range of the research. The terms were submitted separately as **population**, with **intervention** and **outcome** fixed as the string in Trial 2, and re-submitted the string above to retrieve publications from April to July.

Search String:

TITLE-ABS-KEY(("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR pervasive OR ubiquitous OR "event based" OR "self adaptive" OR "self adapt" OR "ambient intelligence" OR "assisted living" OR "agents systems" OR "multiagent systems" OR "systems of systems" OR “software systems”) AND (interoperability OR interconnection OR interoperation OR interaction OR integration OR exchange) AND ("evaluation metric" OR "evaluation method" OR "evaluation model" OR "evaluation process" OR "evaluation methodology" OR "evaluation criteria" OR "evaluation approach" OR "evaluation strategy" OR "measurement method" OR "measurement model" OR "measurement process" OR "assessment method" OR "assessment model" OR "assessment strategy" OR "quality attributes" OR "quality properties" OR "quality characteristics" OR "quality features"))

[15] Reference: Jimenez, H. and Mavris, D. *Assessment of technology integration using technology readiness levels* 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, **2013 –** contact in June, 2nd 2015.

Date: 14th July 2015.

Number of Articles Found: 408

Number of Articles Included in the final list: 16

Results: The string with only "software systems" in the population presented 116 articles in the results, and 21 using "systems of systems". The re-execution of the string used in Trial 2 recovered 14 articles. That results in 151 articles, being 8 register of proceedings and 3 studies in duplicity. The revision process was followed as presented in section 3, and after data extraction and discussion 5 articles were include, however one of them[16] could not be recovered for full reading even after contact with one of the authors.

Revisiting results: Performed after Trial 3

Process: The terms presented in Trial 3 brought a broader perspective to the review, expanding the mindset, hence the need to review past results to mitigate possible articles losses. The changes in the perspective were motivated by the results obtained previously, and made with intent to retrieve more work that could contribute to the review. Every trial that generated a BibTex file was kept and possible to revisit with JabRef.

Date: 10th September 2015.

Considered articles: 3

Number of Articles Included in the final list: 0

Results: The aim was to improve concepts regarding interoperability, extending the concepts of systems and context. We review title and abstract considering 3 possible articles that could add in the outcome of this review. After full reading and data extraction, performed by one of the readers, the result was revised by other reader and decided not to include them in the final set.

Snowballing: After trials and review a list of all the included articles was made.

Process: From the included papers we performed snowballing in two forms: backward and forward. **Backward:** one of the readers was responsible for checking the references presented in the articles and if the title seemed appropriate for the review it was considered. **Forward:** Using Scopus database, one of the readers searched if each article was cited by any other. If so, by reading the title and it appears suitable for the review, this new paper was considered.

Date: September 2015.

---

[16] Reference: Fiebrandt, M. and Wilson, J. *System of systems measurement framework in evaluation of joint mission effectiveness* U.S. Air Force T and E Days, **2008 –** contact in July, 27th 2015.

Considered articles: 15

Number of Articles Included in the final list: nine (six backward, three forward)

Results: The articles found in the snowballing were in a list, with the title and abstract. This list was reviewed by another reader, to include or exclude the article of the final set. It was performed to provide adequate coverage to the review. The final list was made after an agreement among the participants and every paper read, and the data extracted. All articles included are listed in section 3.3.

A.2.2 Paper Quality Evaluation Results

To weigh the importance of individual studies and better detail the data extracted, the selected papers were also evaluated regarding quality. It is important to state that quality in this evaluation is related to adequacy, or how much a paper attend the objectives of this research. By answering the questions presented in section 2.6 is possible to observe each study with this perspective.

Despite the relevance of every question, question 1 to 6 are specific to interoperability, this review topic, therefore more desirable to be answered. Considering a paper that could answer each question and applied in at least one setting, the total in the quality evaluation would be 11 points.

These questions were answered by extracting data presented in section A.1.5 Data Extraction. The forms were completed for one of the authors and the fields reviewed by the others. After applying necessary adjustments, each paper passed through this quality evaluation, and the results were joined in a table form, presented below. The table is ordered from lowest to highest points.

As a cut-off point was used the first quartile, calculated in 3 points. It was chosen to keep only papers with a score above the cut-point specified. All articles are marked in red in the following table did not achieve the score, therefore not considered for the final analysis (see Table A.2 and Figure A.1).

Table A.2. Paper quality evaluation regarding the questions criteria

Paper Index	Paper	Question 1	Question 2	Question 3	Question 4	Question 5	Question 6	Question 7	Question 8	Question 9	Question 10	Question 11	TOTAL
1	McBryan2008	0	0	0	0	0	0	0	0	0	1	0	1
2	Rodrıguez-Domınguez2012	0	0	0	0,5	0	0	0	0	0	1	0	1,5
3	Queriós2015	0	0	0	0	0	0	0	0	1	1	0	2
4	Damas	0	0	0	0,5	0	0	0	0	1	1	0	2,5
5	Reynolds	1	0	0	0,5	0	0	0	0	0	1	0	2,5
6	Flores2004	0	1	0	1	0	0	0	0	0	1	0	3
7	Jimenez2014	1	0	0	0	0	0	0	1	0	1	0	3
8	Kuziemsky	0	0	1	1	0	0	0	0	0	1	0	3
9	Memon2014	0	1	1	0,5	0	0	0	0	0	1	0	3,5
10	Chen2006	0	1	1	1	0	0	0	0	0	1	0	4
11	Wyatt2012	1	0	1	0	1	0	0	0	0	1	0	4
12	Vega-Bargas	0	0	0	0,5	1	1	0	1	0	1	0	4,5
13	Kutvonen2007	1	1	1	1	0	0	0	0	0	1	0	5
14	Naudet	1	1	0	1	0	0	0	1	0	1	0	5
15	Kutvonen2013	1	1	1	0,5	0	0	0	1	0	1	0	5,5
16	Ruokolainen	1	1	1	1	0	0	0	1	0	1	0	6
17	Sullivan	1	1	1	1	0	0	0	1	0	1	0	6
18	Tsalgatidou	0	1	1	1	0	0	0	1	1	1	0	6
19	Álvarez-garcía	0	0	1	0,5	1	1	1	1	0	1	0	6,5
20	Zutshi	1	0	1	0,5	1	0	1	0	1	1	0	6,5
21	Chen2008	1	1	1	1	1	1	0	0	0	1	0	7
22	Rezaei2014	1	1	1	1	1	1	0	0	0	1	0	7
23	Ullberg	1	0	1	0,5	1	1	0	1	1	1	0	7,5
24	Fang2004	1	1	1	1	1	1	0	1	1	1	0	9
25	Madni2014	1	1	1	1	1	1	0	1	1	1	0	9

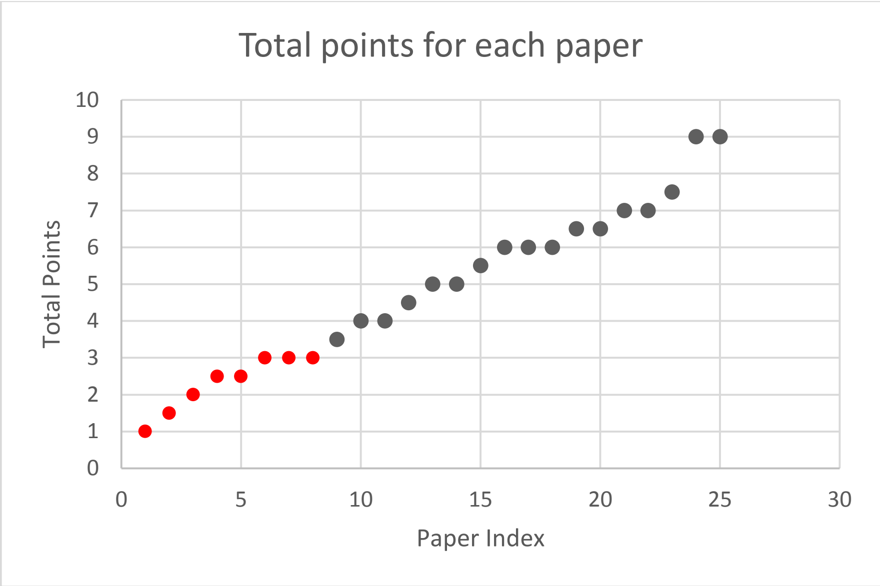

Figure A.1. Total points for each paper

APPENDIX B – Coding Detailing

This appendix presents the detailing of open and axial steps of the coding process of the Qualitative Analysis presented in Chapter4.

B.1 Categories, Subcategories and Codes Description

The information is presented in a hierarchical manner (tree structure) with the category, followed by their subcategories and then the codes that compose them. It intends to give the rationale and justify the belonging of code to a given subcategory, and in turn, the subcategory in the category in question.

B.1.1 Definition

Definition: ***D*****escription of what interoperability means. Our proposed definition is: The ability of things to interact for a specific purpose, once their differences have been overcome. In the categories below we try to identify the key concepts of the definition: the interaction, purpose, and differences.**

- *Ability to exchange*: The exchange gives the interaction between systems here. Exchange means a trade; something is given, and something is received in every transaction. The concept of exchange is one of the most used to define interoperability. The trade can be data, information and other resources of different types. Here, interoperability means the system's ability to exchange something.
    - **Exchange meaningful information despite the instruments** - The interaction here is the exchange of meaningful information, on semantics. Instruments used to enable the interaction are the differences to overcome, representing language, interface or operation environments, information systems, infrastructure types, geographic regions, and cultures.
    - **Exchange resources or information** - The interaction means exchange. Despite being similar (and grouped in the same category), it differs from other exchange information codes, because here the meaning is related only to the trade itself – it does not address the use or purpose and specifies neither instruments nor differences.
    - **Exchanging and using resources or information** - Interoperability means the exchange and use of information. The interaction is the exchange, once more. The distinction here is that only exchanging information is not enough. The information should be used. This use can be seen as the interaction purpose. We differ resources (materials and

other assets that can be drawn to effectively function or to help to achieve something) from information (Data as processed, stored, or transmitted by a computer) as objects of exchange.

- **Finding and sharing information from heterogeneous data sources** - The interaction is to share information. However, to share information, it should be found. Data sources in different systems and different machines. Despite that, the sharing of information is possible. The object of exchange is only information.
- **Information exchange over a network** - The interaction is simply exchanging, and it is concerned with neither the purpose nor the meaning. The object of exchange is only information. This definition clarifies the instrument through the exchange is performed: network (in the used instrument) relies on the possible differences to be overcome.
- **Provide, accept and use resources and information** – "To provide and accept" is the interaction with this code, but it has the idea of choice. The resource is provided and can be accepted or not. If it is accepted, the resource should be used. Only with these three parts this definition is complete. Once again, different objects interact to exchange, the use of exchanged objects is the purpose of the interaction.

- *System property*: Interoperability as a systems property. It can be seen as functionality or requirement.
  - **Functionality provided by middleware services –** It can be seen as functionality when using middleware as bridging solution interoperability can be considered as end purpose. For me, I can see middleware services as an instrument. In some cases, middleware is a solution for the interoperability problem, not the definition *per se*. Therefore, it is a limited definition, related to the author's perspective or domain, being hard to generalize.
  - **Systems property** - Defined as a systems' characteristic. In my opinion, since it is some ability, it makes sense.
  - **Systems quality requirement -** A condition or capability that must be met or possessed by a system. Like ISO, it classifies interoperability as a non-functional requirement.
- *Integration*: Interoperability can be seen as systems integration. The usual misconception of related terms, but in our perspective interoperability is not only integration, although the systems parts need to be somehow related.

    - **Surrogate for integration -** Integration meaning to compose existing systems contributing to a larger one. To consider: *a priori*, integration was considered as a synonym for interoperability to retrieve more information and observe similarities and differences between these concepts.
- *Cooperation*: Interoperability indicates cooperation among systems, meaning to act or work together for a particular purpose. Cooperation also covers: collaboration, operate together, work together, work in harmonized way, function together, function jointly, operate in synergy, composed together.
    - **Ability to cooperate** - Cooperate meaning to act or work together for a particular purpose. For me, they are related concepts but, interoperation differs from cooperation because of purpose. In interoperability, the objective for the interaction can be different among the systems.
    - **Working together despite the differences -** Similar to cooperate, but here the possible differences are stated: regarding decision-making, methods of work, legislations, culture or commercial approaches.
- *Systems relation*: Interoperability meaning systems put into relation. If the system does not interact with others, there is no need for interoperability.
    - **Relation between systems** - The relation represents the way in which two or more people or things are connected; Interaction meaning the mutual influence between two things. To establish a relation, a form to interact, it is the first step towards interoperability. This relation can evolve and change, be more complex, depending on the purpose.

B.1.2 Purpose

***Purpose*: Interoperability is used to achieve a specific purpose. The goal or objective for the interoperability to happen.**

- *To develop business* – The purpose here is to develop collaboration among companies to do business. Therefore, interoperability is key because it is necessary to overcome the organizations' differences.
- *To improve people's quality of life* – It is a purpose and benefit not only of interoperability but also of systems in general. Systems used in daily life and their evolution considering ubiquity, IoT and smart devices need to be interoperable to achieve their particular purpose as well.
- *To improve positioning with systems implementation* – The purpose is related to the improvement in the implementation. With interoperable systems, it is possible to relate front and back systems, web services, payment.

- *To improve services provision to citizens and business* – The purpose here is to improve the services provision, and the perspective is with their government. Example: application interoperability enables people to communicate and make transactions with the government from a single point of entry.
- *To achieve a purpose* – General code of purpose, it changes according to the domain or the relating entities.

B.1.3 Level

Level**: The level of support provided for interoperability in a given interaction. The interoperability occurs between things; we call it here systems. System A has a particular architecture and capabilities, same for system B. They have to interact. Like their particularities they can have different purposes, this can affect their relation in various forms, what we consider here as levels.**

- *About Levels*: Excerpts contributing to the idea of levels being coded mostly as Memos several paragraphs in one code, not individually coded. Some of these excerpts confirm the idea of levels, some reporting that the levels can relate to each other, and other confirms that the level varies from each interaction.
    - **Excerpts confirming the idea of interoperability levels** - Some of these excerpts do not define or present any other information, only list possible interoperability levels, evidencing the levels concept. Level: a position on a scale of amount, rank, quantity, extent, or quality. Level represents here facets, degree, aspects, form and dimension.
    - **They can relate between them** - Some excerpts present the idea of relation among levels E.g.: You can only have semantic level once the syntactic level is established.
    - **Varies from each interaction and system** - Each system has its functionalities. When interacting with other ones, this will establish the interoperability.
- *Technical*: It is concerned with the connectivity, communication, and operation regarding the interacting entities. To enable interaction it is important to consider the architecture style, issues regarding authentication and authorization, the use of technical standards, protocols for communication and transport and an interface between the things - when applicable. Our proposition is that this level enables the interaction between things. Below, the concepts supporting the proposed definition.
    - **Concerned with communication** - To establish the channel where the interaction will be held. Like in the cases where the interaction is an information exchange, communication concerns support this interaction.

The complexity can increase whether considering additional layers such as transport, communication, etc.

- **Concerned with connectivity** - It concerns the connectivity between the parts, and it is the first level to be achieved to allow basic interoperability and to evolve to another level if necessary. It concerns to support and establish a connection to make the interaction happen.
- **Concerned with operations -** The mechanisms used to provide interoperability related to other interaction types aside exchanging information, such as to share functionality and provide services.

- *Syntactic*: This level is concerned with the communication, data exchange, and syntax consistency. To enable this kind of interaction is important to consider the use of standards and protocols, and provide the suitable infrastructure needed and issues regarding data definition - when applicable. Our proposition, as we understand is that this level enables the interaction of things through data. Below, the concepts supporting the proposed definition.
    - **Concerned with communication and data exchange** - When the interaction happens with data, communication and data exchange are major concerns.
    - **Concerned with syntax consistency -** More than just send and receive data, a concern involving both sides is related to the interaction object consistency, in this case, data. Consistency: the quality of always behaving or performing in a similar way, or of always happening in a similar way.
- *Semantic*: This level is concerned with the interpretation and reciprocal understanding between interacting entities. To enable the interaction at this level, it is important to consider the use of semantic web and a priori solutions to be decided at design time, the use of standards, models, and ontologies and shares compatible concepts - when applicable. Our proposition, as we understand, this level enables mutual understanding of the meaning of the data between things. Below, the concepts supporting the proposed definition.
    - **Concerned with the interpretation of the exchanged information** – To achieve a mutual understanding of the data it is necessary to share common concepts. The interpretation is essential for the semantic because it is related to the data meaning explanation.
    - **Concerned with the reciprocal understanding between interacting entities -** For the interactions to be meaningful and understandable, the

interoperable systems need to share common/compatible semantics. Therefore, the interoperating systems should have mechanisms for exchanging, representing, modifying, and updating semantic interaction object descriptions. To the extent that the meaning and form of information are dynamic (i.e., can change over time), these systems need to be able to dynamically modify their information processing approaches and, possibly, their representations.

- *Organizational*: This level is concerned with business rules, policies and constraints, process alignment and the actions necessary to make entities collaborate. To enable interactions at this level, it is important that the assets and structures compatibility, use compensation or negotiation, use standards, and protocols, share common concepts and process - when applicable. Our proposition, as we understand, this level enables the mutual understanding of data meaning between things, under determined conditions. Below, the concepts supporting the proposed definition.
    - **Concerned with actions necessary for the collaboration** - Pragmatic aspect of organizational interoperability, it is usually related to collaboration and communication. It differs from other concerns because it clears states about collaboration, despite not specifying the actions.
    - **Concerned with aligning processes between the interacting entities** - More related to solve differences between the business processes of governance, finance, legislation to conduct business in a seamless way (that is the interaction).
    - **Concerned with business rules, policies and constraints –** The specific conditions for each organization can be represented by its business aspects, influence structures, and behavior. It can be internal (intra-organization) or external, but the concern is that it should be compatible to help the organization achieve its purposes. Ex. Legal issues can affect business interactions.

B.1.4 Perspective

***Perspective*: Different perspectives related to interoperability interpretation in diverse areas. The way one looks at interoperability can change the definition and the needs for the levels. The perspective in which interoperability is seen affects interoperability in different levels.**

- *About Perspectives*: Excerpts contributing to the idea of perspectives, or different manners to observe, achieve and treat interoperability.

    - **Excerpts confirming the idea of interoperability perspectives** - Some of these excerpts do not define or present any other information than those confirming the concept of perspectives. They present different domains or questions related to interoperability interpretation in diverse areas.
- *Organization perspective*: This perspective is related to public organizations (as governments and military forces), business and enterprises. It considers policies and capabilities of each organization, the interaction among people and applications. The technology is used to support business. Interoperability in this perspective should concern human actions that require reasoning, differentiate decisions in design, runtime, and address process issues to each interacting organization.
    - **Concerned with applications that have to interoperate** - Applications and systems are specific implementations parts of business domains. In the organizational perspective, it is important to consider what the interacting entities are.
    - **Organization policies and capabilities** - This perspective is based on a business knowledge that must be explicitly represented to achieve the interoperation.
    - **Use technology to interoperate with others to conduct business -** The instruments used to enable the interoperation between organizations. For example, people-to-people interaction. In this work, the concern is when their relations are supported by IT, like the use of technical solutions to communication in distributed teams.
- *Systems perspective*: With no special effort from users, systems can seamless interact and be available for automatic runtime composition - these concepts are related to interoperability in the systems perspective, and for that, semantic and syntactic levels should be addressed.
    - **Concerns with availability for runtime composition in a self-administrative manner** - This is important for ubiquity, IoT, and should be evolved**.** Related to challenges and need further research.
    - **Concerns with instant and seamless interaction among systems** - Present instant and seamless characteristics of interoperability, and are related to hiding complexity from the user and not demanding any effort.
    - **Concerns with systems interaction without special effort from stakeholders** - The idea is automatic interaction, spontaneous

interoperability where there should be no effort for the user. Consider interoperability characteristics.

    - **Relates to syntactic and semantic level** - System interoperability needs to address both syntactic interoperability and semantic interoperability.

- *Service perspective*: Using service architecture as a solution to integrate functionalities, relates to technical and semantic levels. Issues that should be addressed relates to orchestration, verification of service provision and specific challenges of new devices.
    - **Concerns with availability for runtime composition in a self-administrative manner** - Relates to interoperability characteristic of automatic and spontaneous.
    - **Concerns with the dynamically registering, aggregating and consuming composite services** - Related to the interoperability definition of providing, accepting and using services from heterogeneous sources.
    - **Concerns with the protocol, signature and semantic** - Another example of how the perspectives are related to levels
    - **Concerns with a service oriented architectures** - This affects the decision at the technical level.
    - **Relates to technical and semantic levels** - another example of how the perspectives are related to levels.

B.1.5 Benefits

***Benefits*: Some advantages observed when interoperability is achieved.**

- *Improves Understanding*: Interoperability can bring meaning to the interactions and a better understanding of the related entities.
    - **Allows better understanding and meaningful interaction of systems -** With interoperability, in different levels of support, meaning is related to comprehension – as it enhances simple interactions.
    - **Enables information to be universally used** - Which means to be accessible, reusable and comprehensible. Consider the concept of "universal access".
- *Increases flexibility*: When entities can relate and interact to achieve a purpose their boundaries are extended and their limits more flexible.
    - **Compose new systems based on existent capabilities –** To be flexible enables the composition and brings unanticipated/unprecedented tasks from new combinations of existing functions.

- **Enables reuse** - Interoperability can decrease costs when creating a new system by reusing existing ones in multiple ways for different purposes.
- **Increase systems flexibility** - It refers to designs that can adapt when external changes occur, by allowing mixing and matching of systems.

- *Hides complexity*: Beautiful seams or seamless interaction.
    - **Hides systems complexity from the user** - Where one thing begins and the other ends is not easily noticed.
- *Creates cohesion*: Cohesion: A condition in which the entities are closely united // the degree, which the elements belong together. With the compatibility among entities required to interoperability takes place the entities are more easily related.
    - **Creates cohesion among entities** - When in agreement and working well together to achieve a specific purpose.

B.1.6 Issues

***Issues*: Some reported issues when a) addressing interoperability, or b) interoperability is not addressed.**

- *Related to Levels*: Issues related to the previously presented levels.
    - *Related to* Semantic: Issues related to possible conceptual conflicts, different context and the complexity to deal with semantic level.
        - **Complexity to deal with semantic level** - Generic excerpts that report the difficulty in addressing semantic issues.
        - **Conceptual conflicts** - Naming conflicts, Domain conflicts, Structural conflicts, Metadata conflicts.
        - **Divergent knowledge and context** - This undermines comprehension and interpretation (in the context of the actor for instance).
        - **The dynamic environment and runtime composition** - It is the difference whether the parties know *a priori* they want to interact, but spontaneously is more complex because it is difficult to confirm if there is a mutual understanding or shared interpretation.
    - *Related to Syntactic:* Syntax differences and representations.
        - **Different formats and representations** - When any definition related to data is missing (data type, data structure, data representation, data format, language) can lead to syntactic issues.

    - *Related to Technical*: When having interoperability in mind the complexity can raise, and infrastructures are the major issues.
        - **Different infrastructure** - Basic interoperability problems including platform heterogeneity, operating system heterogeneity, and any environment feature.
        - **Increase technical complexity** - Having interoperability in mind as a system requirement some technical decisions should be addressed, and this can increase complexity.
- *Related to Planning*: Like any other project decision, to develop for interoperability affects costs, and it can rise and become more complex later in the project as the time goes.
    - **Increase costs** - To make interoperable systems, like the need for interoperability comes after the system implementation.
    - **Modifying a system can bring problems not previously planned** - In the case of transformations, the systems will be different from original scope, bringing various problems. V&V activities are required.
    - **When the need for interoperability arises after implementation, complexity rises** - It extends interoperability beyond the original scope of the system.
- *Related to Perspectives*: Issues related to the particularities of addressing interoperability in a given perspective.
    - **Complexity drivers to achieve interoperability** - Examples: Certification and accreditation, Acquisition, Structure, Integration mechanisms and verification/validation.
    - **Different incompatibilities have different impacts -** Example: The difference in the targeted clients has an impact on the platform and application available.
    - **It is a concern involving different perspectives** - To be truly interoperable the perspectives should be aligned**.**
    - **Non-technology aspects of interoperability** - Human factors address the non-technology aspects of interoperability, the effect of established procedures as well as the naturally unpredictable behavior of any operation involving humans.
    - **Privacy and Security issues -** Examples of other requirements that can affect interoperability**.**

B.1.7 Challenges

***<u>Challenges</u>*: Possible paths for the field to evolve.**

- *<u>Research</u>*: Academic development.
    - **Address specific challenges for ubiquitous computing -** Service oriented approach based on web services may not deal with all the complexity related to ubiquitous computing like processing and memory.
    - **Interoperability "on the fly" -** For spontaneous interoperability, one cannot rely on shared *a priori* knowledge on the scale needed for pervasive computing. So it should somehow adapt at deployment time and runtime to integrate their functionality and dynamically interoperate with other application software.
    - **The field needs further development** - Focus especially on those fields where compatibility is still low, i.e., areas lacking or with conflicting standard developments or lacking uniform implementation of standards.
- *<u>Practice:</u>* To be achieved so that truly interoperability exists.
    - **Dynamically locate and integrate application functionality** - Relates to: runtime and on the fly complexity
    - **Legacy and new objects interoperability** - How to deal with legacy systems that need to interact with new ones?
    - **Protocols are not a silver bullet** - Some problems cannot be tackled with protocols only.
    - **Runtime monitoring** - Relates to runtime and on the fly complexity.
    - **Unify protocols and architecture to IoT** - One of the ideas to overcome it aims to create a standardized new IoT common protocol or interoperable architectures abstracting these inconsistencies.

B.1.8 Evaluation Methods

***<u>Evaluation Methods</u>*: It assists an organization, program, a project to assess or aid the decision-making on interoperability. These methods (or models) can present the degree of achievement or adequacy of interoperability in a given context.**

- *<u>Other methods</u>* **-** Each method is coded with name and original work if cited 2 or more times, the others are in "others".
    - **Other evaluation options**
    - Questionnaire 2 different
    - Quantitative methods 1
    - Business Interoperability Quotient Measurement Model - BIQMM 1

- Levels of Conceptual Interoperability Model - LCIM 2 (only cited)
- INTEROP Framework 1
- IDEAS Framework 1
- E-health interoperability framework 1
- European Interoperability Framework 1
- Systems-of-Systems Interoperability Model - SOSI 2 (only cited)
- Military Communications and Information Systems Interoperability 1
- Stoplight Model 1
- Spectrum of Interoperability Model 1

- *Athena*
  - **Enterprise Interoperability Maturity Model –** Each area of concern would be defined by a set of objectives and goals relevant to interoperability and collaboration issues. Depending on the absence, or presence of the maturity indicators, interoperability, and collaboration maturity level would be defined for each area of concern.
- *I-score*
  - **i-Score Model –** It is an architecture -based method of measuring interoperability of complex networks of non-homogeneous systems. It uses a multiplicity matrix for the pairs, where each element can be evaluated as -1 (human translation necessary to be interoperable), 0 (machine translation required to be interoperable), 1 (no human or machine actions necessary to be interoperable).
- *IAM*
  - **Interoperability Assessment Methodology** - Introduces nine components, which are: requirements, node connectivity, data elements, protocols, information flow, information utilization, interpretation, latency, and standards. The nine components each included either a "yes/no" response or a mathematical equation.
- *LISI*
  - **Levels of Information Systems Interoperability** - With five interoperability levels (0–4) Isolated, Connected, Functional, Domain, and Enterprise, in which each interoperability level exists in a specific environment. The attributes of the LISI Reference Model contain Procedures, Applications, Infrastructure, and Data.
- *OIMM*

- **Organizational Interoperability Maturity Model** - This model extends the LISI model into the more abstract layers of command and control support. Describing the ability to interoperate, five levels of organizational maturity are defined in this model (independent, *ad-hoc*, collaborative, combined and unified). Preparedness, Understanding, Command style, and Ethos.

- *QIM*
    - **Quantification of Interoperability Methodology** - They stated "interoperability of systems, units, or forces can be factored into a set of components that can quantify interoperability" and they identified the seven necessary components as languages, standards, environment, procedures, requirements, human factors, and media.

B.1.9 Attributes

Attributes: **Systems attributes related to interoperability and the decisions/mechanism to address them**.

- *Interoperability Mechanisms:* Recurrently used throughout the levels being suitable (adequate, having the qualities that are right, needed, or appropriate) to address interoperability.
    - **Decisions made at design-time** - Each decision involving interaction between different entities is related to interoperability and brings consequences and particularities to be dealt. Design-time service interoperability is the matter of analyzing the effort needed for possible future services to interoperate, regardless of their current relationship to each other. Ex.: Assure compatibility before interaction - A priori "contract" to interoperate.
    - **Decisions made at runtime** - Each decision involving interaction between different entities is related to interoperability and brings consequences and particularities to be dealt. Runtime interoperability is concerned with the interoperability of services that are supposed to be working together. Ex.: After systems completion, the need for interoperability comes.
    - **Human actions regarding interoperability** - It addresses unexpected business issues requiring human reasoning. Human decisions are present throughout the process of interaction, but it becomes clearer with useful purposes like business interoperation.

- **Share common information and concepts** - Information specifications providing base for semantic exchange such as Perspective (standardized terms and their synonyms for many purposes. Ex.: client x user), terminology and controlled vocabularies; data definitions; units of expressions; computational methods and assumptions used to produce results; and clear definition of limits and restrictions. Often this is more than simple semantic, and the concepts have to be aligned with organizational conditions.
- **Use suitable architecture** - Architectural decisions has consequences affecting the whole system infrastructure. Ex: Architecture: Services, how - SOAP or Rest? It is necessary to decide on the architecture to address connectivity and communication.
- **Use suitable protocols** - A protocol stipulates the rules for communication between entities, and a standard is a specification agreed upon and implemented and adopted by industry/vendors. Similar idea here, to support connectivity, communication, transport, and operations for the interaction according to the relating entities. This mechanism encompasses the decision to use a protocol and to provide the required structure to the interaction and consistency between the entities.
- **Data definition –** To interact through the data it is important to make decisions related to the data type, data structure, data representation, data format, language – and this also is related to the standards to be used and the consistency between the relating entities. Ex. If I use XML, what else is required?
- **Use suitable interaction interface** - To achieve conformance between parts to enable the interaction, the designers should decide on operations, types, the order of parameters and others technical issues.
- **Use suitable models –** Like standards and protocols, this is a decision of levels and affects the whole perspective.
- **Use suitable standard** - A standard is a document providing requirements, specifications, guidelines or characteristics that can be consistently used to ensure that materials, products, processes and services are fit for their purpose (ISO definition). The Standards are present in every level, and their use is interoperability characteristic. Here the idea is to support connectivity, communication, and operations to

enable that are required for the interaction. **Ex.:** XML or RDF based solutions to address syntactic level.

- **Use services to enable interaction** – Concerns to: orchestration (describes the automated arrangement, coordination, and management of complex computer systems, middleware, and services), cohesion between description and provision in services, and the required verification activity.
- **Use transformation to enable interaction** - It requires transformation/modification of related systems. Transformation is possible if the change is permitted and is enough knowledge to make it happen. It aims to homogenize systems.
- **Use semantic web to enable understanding –** The use of semantic web can be a solution as it provides a common framework allowing data to be shared and reused across applications, enterprise, and community boundaries aiming reciprocal understanding between things.
- **Use "bridges" to enable interaction** - *A posteriori* solution that acts as an intermediate between inter-related systems causing an interoperability problem. Ex.: middleware.
- **Use common ontologies to enable understanding -** It is pointed out to be used to achieve conformance between context representations. A way to provide name, the definition of the types, properties, and interrelationships of the entities, to bring compatibility to achieve a mutual understanding.
- **Use open source solutions to create open platforms** - Necessary for a major goal to create open platforms.
- **Use compensation to enable interaction –** To provide a mechanism for the interaction to happen under / despite the conditions: when the negotiation process fails at system build time, an alternative is to use some compensation a posteriori: i.e. an actor accepts something in exchange of a disagreement, for the fulfillment of the super-system objective.
- **Use negotiation to enable interaction** - To provide a mechanism for the interaction to happen under/despite the conditions: negotiation mechanisms are used when different actors influencing different systems have divergent views, objectives or business processes that must be

taken into account when trying to build an interoperable system. Negotiation allows finding mutual agreements between actors.

- *General Attributes :* Typical systems' attributes related to interoperability.
    - **Availability -** Directly related to connectivity, because during the interaction time all the interacting systems should be available.
    - **Adaptive behavior -** Ability to dynamically adapt to the environment the systems is being used within its limitations.
    - **Compatibility -** Compatibility of a system deals with what ease the system is can operate with shared applications. This attribute also covers the system's compatibility on different-different platforms. The platforms can be either hardware, software or both.
    - **Conformance with systems' requirements** - To the overall project success regarding not only interoperability. It varies from system to system and affects interoperability.
    - **Conformance with organization requirements** - Each organization has its issues that should be aligned with the other ones to interoperate. Related to decision-making and also non-technological aspects. Ex. Authority (who is authorized to do what?).
    - **Dynamic connection** – To enable the interaction between entities is necessary to identify the relating ones. Decisions like these refer to the identified partners to be connected according to their permissions. Authenticate and authorize interactions in automatic connection based on previous interactions (history) or re-establish lost connection.
    - **Standardization** - It can be applied to any level, with the necessary changes.

B.1.10 Context

***Context*: Context is any information that can be used to characterize the situation of an entity. Context information is any data that can be used to characterize the status of a person, place, or object that is considered relevant.**

- *Business Information –* Concerns with the context information that guides the interoperability levels in the business (organizational) perspective bringing significance to the interactions.
- *Extrinsic Environment Information* **-** Definition of extrinsic: a) not forming part of or belonging to a thing; b) originating from or on the outside; c) originating outside a part and acting upon the part as a whole. This information refers to variables like position,

time, weather, speed or any information that can be perceived by the system or acquired by a sensor.

- *Interaction Time -* It is related to the idea that the interaction time is a context variable that can influence the interaction among entities, affecting for instance: a) connectivity (e.g. the link should be available at any moment); b) dynamicity (e.g. information is dynamic, over time can change meaning and form).
- *Intrinsic Environment Information -* Definition of intrinsic: a) belonging to the essential nature or constitution of a thing; b) originating and included wholly within a part. The environment out of the "operator" control on which the entities are interacting can influence the interoperability.
- *User knowledge -* The user's background can affect interoperability as knowledge compromises the user's comprehension and perspectives regarding a system, (related to semantic).

APPENDIX C – Categories Description

C.1 Subcategories Examples

A **description** of each category is provided. This description embraces all the codes that belong to the category, to keep the consistency of the segments. It is also presented the **groundedness** of each subcategory, meaning how many excerpts are related to it, and the **density** of each subcategory, meaning how many cases support it. We illustrate the subcategory with one excerpt **example** and the discussion regarding their relation and interpretation. The relation of the categories to their subcategories is the type **part of**, used to represent that the subcategories, and their respective codes, form the whole concept under discussion.

C1.1 Definition

Table C.1.1 - Category: Definition

Category: Definition		
Ability to exchange: the exchange gives the interaction between systems here. Exchange means a trade; something is given and something is received in every transaction. The trade can be data, information and other resources of different types.		
Groundedness: 42 segments are coded in this subcategory that represents six codes. **Density:** 11 cases support it.	**Example:** "… such that information gets exchanged between collaborating parties and it is used in a meaningful way despite differences in language, interface or operation environments" (RUOKOLAINEN; KUTVONEN, 2006)	**Discussion:** the interaction information exchange. The purpose is to use this information in a meaningful way. Differences in language, interface or operation environments can not be an obstacle.
System property: interoperability as a system's property. Can be seen as functionality or requirement.		
Groundedness: 10 segments are coded in this subcategory that represents three codes. **Density:** 7 cases support it.	**Example:** "Interoperability was first defined as a property of IT systems" (WYATT; GRIENDLING; MAVRIS, 2012)	**Discussion:** it is a vague and open definition since it does not provide or bring information of what is involved in interoperability. However, it sustains what most of the definitions considers: interoperability as an ability of the system.

Integration: in our perspective interoperability is not only integration, although the systems parts need to be somehow related. Integration means to compose existing systems contributing to a larger one.		
Groundedness: 7 segments are coded in this subcategory that represents one code. **Density:** 6 cases support it.	**Example:** "…interoperability often serves as a surrogate for integration when independently developed, stand-alone systems are combined to provide a new capability" (MADNI; SIEVERS, 2013)	**Discussion:** independently developed, stand-alone systems are very to have differences that do not prevent the interaction, that aims to provide a new system capability.
Cooperation: interoperability indicating cooperation among systems, meaning to act or work together for a particular purpose. Cooperation also covers: collaboration, operate together, work together, work in harmonized way, function together, function jointly, operate in synergy, composed together.		
Groundedness: 28 segments are coded in this subcategory that represents two codes. **Density:** 8 cases support it.	**Example:** "ability of two or more software components to cooperate despite differences in language, interface, and execution platform" (FANG; HU; HAN, 2004)	**Discussion:** interaction is regulated by cooperation, which has a purpose built in its meaning, not being hampered by differences.
Systems relation: interoperability meaning systems put into relation. Relation representing the way in which two or more systems are connected. If the system does not interact with others, there is no need for interoperability. This relation can evolve and change, be more complex, depending on the purpose.		
Groundedness: 9 segments are coded in this subcategory that represents one code. **Density:** 3 cases support it.	**Example:** "Because such resources of a system are themselves systems, interoperability simply concerns relations between systems" (NAUDET et al., 2010)	**Discussion:** in this example, it considers the systems components as systems as well, a reality observed in systems-of-systems for example. Relations between systems can be presented in different manners like a simple communication link, a constraint imposed on a system by another, influence, each one with its differences.

C.1.2 Purpose

Table C.1.2 - Category: Purpose

<table>
<tr><th colspan="3">Category: Purpose</th></tr>
<tr><td colspan="3">Despite the differences, business is developed between companies: the purpose here is to develop collaboration among companies to do business. Therefore, interoperability is key because it is necessary to overcome the organizations' differences.</td></tr>
<tr><td>Groundedness: 2 segments are coded in this subcategory that represents one code. Density: 1 case support it.</td><td>Example: "to work in a harmonized way at the level of organization and company despite, for example, the different modes of decision-making, methods of work, legislations, culture of the company and commercial approaches…" (CHEN, 2006)</td><td>Discussion: differences exist between organizations but to do business and "work in a harmonize way" they should be overcome (interoperability).</td></tr>
<tr><td colspan="3">To improve people's quality of life: is a purpose and benefit not only of interoperability but also of systems in general. Systems used in daily life and their evolution considering ubiquity, IoT and smart devices need to be interoperable to achieve their particular purpose as well.</td></tr>
<tr><td>Groundedness: 1 segment is coded in this subcategory that represents one code. Density: 1 case support it.</td><td>Example: "uses technology to improve elderly people's quality of life by increasing their autonomy in daily activities" (GJORESKI et al., 2015)</td><td>Discussion: in the ambient assisted living scenario the purpose is to improve people's life, and it is done by different technologies (accelerometers sewed into clothing, 3D scanners, smartphones and others) that should interact.</td></tr>
<tr><td colspan="3">To improve positioning with systems implementation: the purpose is related to the improvement in the implementation. With interoperable systems is possible to relate front and back systems, web services, payment.</td></tr>
<tr><td>Groundedness: 1 segment is coded in this subcategory that represents one code. Density: 1 case support it.</td><td>Example: "to assess their current status concerning eGovernment interoperability and the steps needed to improve their positioning about system implementation and services provision to citizens and businesses" (ZUTSHI;</td><td>Discussion: the authors relate interoperability and its assessment to improve systems and services concerning government (organizational perspective).</td></tr>
</table>

	GRILO; JARDIM-GONCALVES, 2012)	
To improve services provision to citizens and business: the purpose here is to improve the services provision, and the perspective is with their government. Example: application interoperability enables people to communicate and make transactions with the government from a single point of entry.		
Groundedness: 1 segment is coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** “to assess their current status concerning eGovernment interoperability and the steps needed to improve their positioning about system implementation and services provision to citizens and businesses” (ZUTSHI; GRILO; JARDIM-GONCALVES, 2012)	**Discussion:** the authors relate interoperability and its assessment to improve systems and services concerning government (organizational perspective).
To achieve a purpose: general code of purpose, it changes according to the domain or the relating entities. Coded when no specific objective is defined.		
Groundedness: 5 segments are coded in this subcategory that represents one code. **Density:** 4 cases support it.	**Example:** "This enables enterprises to, for instance, build partnerships, deliver new products and services, and/or become more cost efficient" (ZUTSHI; GRILO; JARDIM-GONCALVES, 2012)	**Discussion:** the codes in this subcategories report several different needs as motivation and possible benefits for interoperable entities.

C.1.3 Level

Table C.1.3 - Category: Level

Category: Level		
About Levels: excerpts contributing to the idea of levels being coded most as *Memos* several paragraphs in one code, not individually coded. Some of these excerpts confirm the idea of levels, some reporting that the levels can relate to each other, and other confirms that the level varies from each interaction.		
Groundedness: 40 segments are coded in this subcategory that represents three codes. **Density:** 12 cases support it.	**Example:** "The degree of interoperability should be defined when referring to specific cases" (WYATT; GRIENDLING; MAVRIS, 2012)	**Discussion:** in this case, the authors confirm the levels concept (named in the original paper as a *degree*) but they do not detail it with further information.

<table>
<tr><td colspan="3">Technical: it is concerned with the connectivity, communication, and operation regarding the interacting entities. To enable interaction it is important to consider the architecture style, issues regarding authentication and authorization, the use of technical standards, protocols for communication and transport and the interface between the things - when applicable. Our proposition is that this level enables the interaction between things.</td></tr>
<tr><td>Groundedness: 9 segments are coded in this subcategory that represents three codes. Density: 7 cases support it.</td><td>Example: "Technical Interoperability is typically associated with hardware/software components, systems, and platforms that enable machine-to-machine communication" (REZAEI et al., 2014)</td><td>Discussion: the definition presented in this study clearly states the concerns with connectivity and the infrastructure necessary. Several decisions and mechanisms have to be addressed to deal with such concerns.</td></tr>
<tr><td colspan="3">Syntactic: this level is concerned with the communication, data exchange, and syntax consistency. To enable this kind of interaction is important to consider the use of standards and protocols, and provide the suitable infrastructure needed and issues regarding data definition - when applicable. Our proposition, as we understand is that this level enables the interaction of things through data.</td></tr>
<tr><td>Groundedness: 5 segments are coded in this subcategory that represents two codes. Density: 3 cases support it.</td><td>Example: "Syntactic interoperability is the ability of two or more systems to communicate and exchange data." (Madni and Sievers 2013)</td><td>Discussion: once connectivity is enabled (technical) the systems can interact with data. Data formats, syntax consistency and the used language should be decided for the systems "to communicate and exchange data."</td></tr>
<tr><td colspan="3">Semantic: this level is concerned with the interpretation and reciprocal understanding between interacting entities. To enable the interaction at this level is important to consider the use of semantic web and a priori solutions to be decided at design time, the use of standards, models, and ontologies and share compatible concepts - when applicable. Our proposition, as we understand, this level enables mutual understanding of the meaning of the data between things.</td></tr>
<tr><td>Groundedness: 12 segments are coded in this subcategory that represents two codes. Density: 8 cases support it.</td><td>Example: "To achieve semantic interoperability in a heterogeneous environment, the meaning of the information that is interchanged has to be understood across the</td><td>Discussion: an example is where one entity calls charge in a particular web service “Fee” with the currency type "Dollars" and the other entity calls the charge “Cost” and</td></tr>
</table>

	systems" (Sullivan and Lewis 2003)	expects "Euro." The misunderstanding of the term charge due hinders the service to lack of semantic interoperability.
Organizational: this level is concerned with business rules, policies and constraints, process alignment and the actions necessary to make entities collaborate. To enable interactions at this level, it is important the assets and structures compatibility, use compensation or negotiation, use standards, and protocols, share common concepts and process - when applicable. Our proposition, as we understand, this level enables the mutual understanding of data meaning between things, under determined conditions.		
Groundedness: 8 segments are coded in this subcategory that represents three codes. **Density:** 5 cases support it.	**Example:** "Enterprise Interoperability has been defined as a field of activity with the aim to improve the manner in which enterprises, using Information and Communications Technologies (ICT), interoperate with other enterprises, organizations, or with other business units of the same enterprise, to conduct their business." (ZUTSHI; GRILO; JARDIM-GONCALVES, 2012)	**Discussion:** an example is to align and match business and legal rules for automated transactions through services of organizations from different countries. To conduct legitimate negotiations, determinate, and particular conditions should be addressed.

C.1.4 Perspectives

Table C.1.4 - Category: Perspectives

Category: Perspectives		
About Perspectives: excerpts contributing to the idea of perspectives, or different manners to observe, achieve and treat interoperability. They also present different domains or questions related to interoperability interpretation in diverse areas.		
Groundedness: 6 segments are coded in this subcategory that represents one code. **Density:** 3 cases support it.	**Example:** "However, there are multiple interpretations of what exactly interoperability means and in what areas (communications, physical systems, personnel, etc.) it is	**Discussion:** excerpts like the one in example present the authors understanding that there are multiple interoperability interpretations and its

	relevant." (WYATT; GRIENDLING; MAVRIS, 2012)	relevance in different areas. To our knowledge these two concepts are related: the understanding on interoperability can change according to the perspective in the area it is applied to.
Organization perspective: this perspective is related to public organizations (as governments and military forces), business and enterprises. It considers policies and capabilities of each organization, the interaction among people and applications. The technology is used to support business. Interoperability in this perspective should concern with human actions that require reasoning, differentiate decisions in design, runtime, and address process issues to each interacting organization.		
Groundedness: 10 segments are coded in this subcategory that represents three codes. **Density:** 6 cases support it.	**Example:** "Hence, the concept of Business Interoperability goes beyond IT, into organizational aspects of businesses, and includes the level of people-to-people interactions." (ZUTSHI; GRILO; JARDIM-GONCALVES, 2012)	**Discussion:** this excerpt presents the idea of some authors regarding the extension of the interoperability concept. In this work, we recognize the particularities of each organization, but we focus on systems and the interactions supported by them.
Systems perspective: with no special effort from users, systems can seamless interact and be available for automatic runtime composition - these concepts are related to interoperability in the systems perspective.		
Groundedness: 4 segments are coded in this subcategory that represents four codes. **Density:** 3 cases support it.	**Example:** "Any system or components thereof, for which the interoperability is considered, must have requirements in common. Without such requirements, systems developers and acquisition managers have no obligation to deliver interoperable systems." (REZAEI et al., 2014)	**Discussion:** systems are interoperable according to their requirements. Requirements elicitation, project planning, and the whole development process is addressed from the systems perspective.
Service perspective: using service architecture as a solution to integrate functionalities, relates to technical and semantic levels. Issues that should be addressed relates to orchestration, verification of service provision and specific challenges of new devices.		

Groundedness: 9 segments are coded in this subcategory that represents five codes. **Density:** 8 cases support it.	**Example:** "to dynamically register, aggregate and consume composite services of an external source, such as a business partner or an Internet-based service provider, in a seamless manner " (REZAEI et al., 2014)	**Discussion:** recurrently in the studies set we analyzed the service perspective was considered separately from the systems perspective. We decided to code them separately as well considering services particularities like compatibility between service signatures, invocation order, service provision, and others.

C.1.5 Benefits

Table C.1.5 - Category: Benefits

Category: Benefits		
Improves Understanding: interoperability can bring meaning to the interactions and a better understanding of the related entities.		
Groundedness: 4 segments are coded in this subcategory that represents two codes. **Density:** 3 cases support it.	**Example:** "If computational algorithms and information processing are not semantically compatible, the results are likely to be meaningless or, worse yet, misleading." (Madni and Sievers 2013)	**Discussion:** is evident that interoperability is desirable (and sometimes essential) for meaningful interactions among services, systems, and organizations.
Increases flexibility: when entities can relate and interact to achieve a purpose their boundaries are extended, and their limits are more flexible.		
Groundedness: 5 segments are coded in this subcategory that represents three codes. **Density:** 2 cases support it.	**Example:** "Similarly, interoperability can reduce the cost of creating new capabilities by allowing existing systems to be reused in multiple ways for multiple purposes." (Madni and Sievers 2013)	**Discussion:** Compose new systems based in existent ones requires flexibility, that represents the extent to which a system is adaptable or easily applied to other needs.
Hides complexity: beautiful seams, hides systems complexity from user where one thing begins, and the other ends do not easily notice.		

Groundedness: 1 segment is coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "An unheralded advantage of interoperability is that it hides overall system complexity from users by creating the illusion of an integrated system" (Madni and Sievers 2013)	**Discussion:** the use of computer inserted in a seamless manner in the environment, making it effectively invisible to the user is desirable in ubiquitous computing (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007), and interoperability directly contributes to creating this illusion.
Creates cohesion: with the compatibility among entities, required to interoperability takes place, the entities are more easily related.		
Groundedness: 1 segment is coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "Organizational interoperability creates cohesion amongst approaches to governance, finance, legislation and business processes." (CHEN; DOUMEINGTS; VERNADAT, 2008)	**Discussion:** cohesion can be seen as the degree to which entities belong together. In the example, interoperability enables parts to be connected to each other harmonically.

C.1.6 Issues

Table C.1.6 - Category: Issues

Category: Issues		
Related to Levels: issues related to the previously presented levels like possible conceptual conflicts, different context or different infrastructure.		
Groundedness: 36 segments are coded in this subcategory that represents seven codes. **Density:** 9 cases support it.	**Example:** "The tools to realize web services standards may cause interoperability problems because different developers have a different explanation on these standards and standards often have some open points to be freely extended." (FANG; HU; HAN, 2004)	**Discussion:** standards are often seen as one of the main mechanisms to achieve interoperable systems. However, this excerpt in example reports some issues regarding the use of standards that can be related to semantic (*different explanation on these standards*) and technical (to be freely extended) levels.

Related to Planning: like any other project decision, to develop for interoperability affects costs, and it can rise and become more complex later in the project as the time goes.		
Groundedness: 4 segments are coded in this subcategory that represents three codes. **Density:** 2 cases support it.	**Example:** "It is also important to realize that systems are often designed and implemented before a recognized need for their interoperation exists. Such a need, when it arises, extends interoperability beyond the original scope of the system." (Madni and Sievers 2013)	**Discussion:** constant improvements and project changes can lead to a system evolve to something unplanned over time. To achieve interoperability in a system not originally designed for it can bring issues of privacy and security, change in performance and increase the project cost.
Related to Perspectives: issues related to the particularities of addressing interoperability in a given perspective, like privacy and security issues.		
Groundedness: 10 segments are coded in this subcategory that represents five codes. **Density:** 5 cases support it.	**Example:** "Acquisition is a complexity driver due to multiple acquisition programs, multiple systems' life cycles across programs, and the need to achieve interoperability among legacy and new systems." (Madni and Sievers 2013)	**Discussion:** system acquisition is a complexity driver due organization´s purchase policies and the need to achieve interoperability among legacy, and new systems are examples of issues related to perspectives.

C.1.7 Challenges

Table C.1.7 - Category: Challenges

Category: Challenges		
Research: academic development especially on those fields where compatibility is still low, i.e., areas lacking or with conflicting standard developments and particularities of other research areas like ubiquitous computing.		
Groundedness: 10 segments are coded in this subcategory that represents four codes. **Density:** 5 cases support it.	**Example:** "…we cannot, therefore, rely on shared *a priori* knowledge via common interoperability standards to solve the application interoperability problems on the scale needed for pervasive	**Discussion:** the usual approach of an *a priori* agreement between the interacting systems should evolve to be adapted at runtime to spontaneous

	computing." (SULLIVAN; LEWIS, 2003)	interoperate with another system.
Practice: to be achieved in systems development to truly interoperability exists, in different levels and perspectives.		
Groundedness: 8 segments are coded in this subcategory that represents five codes. **Density:** 4 cases support it.	**Example:** "The integration of heterogeneous objects (legacy and future created) from a technological standpoint." (VEGA-BARBAS et al., 2012)	**Discussion:** this is a recurrent problem, where legacy systems usually valuable for organizations, but they may be enhanced by adding new services or composed in different manners. However, changes are often difficult to achieve due to old platform and architecture, among others.

C.1.8 Evaluation Methods

Table C.1.8 - Category: Evaluation Methods

Category: Evaluation Methods		
Other methods: 14 methods are cited only one time and usually not detailed.		
Groundedness: 14 segments are coded in this subcategory that represents one code. **Density:** 9 cases support it.	**Example:** "The evaluation committee evaluated interoperability using the questionnaire and three metrics: use of open source solutions, availability of libraries for development, and integration with standard protocols." (GJORESKI et al., 2015)	**Discussion:** illustration example, no discussion held.
Athena: Enterprise interoperability maturity model – each area of concern would be defined by a set of objectives and goals relevant to interoperability and collaboration issues. Depending on the absence, or presence of the maturity indicators, interoperability, and collaboration maturity level would be defined for each area of concern.		
Groundedness: 2 segments are coded in this subcategory that represents one code. **Density:** 2 cases support it.	**Example:** "The ATHENA interoperability framework is structured into three levels and based on sources of results and usage of the framework"	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that

	(CHEN; DOUMEINGTS; VERNADAT, 2008)	cites them, varying only the amount of information provided, not the definition of the model itself.
i-Score model – is an architecture-based method of measuring interoperability of complex networks of non-homogeneous systems. It uses a multiplicity matrix for the pairs, where each element can be evaluates as -1 (human translation necessary to be interoperable), 0 (machine translation required to be interoperable), 1 (no human or machine actions necessary to be interoperable).		
Groundedness: 2 segments are coded in this subcategory that represents one code. **Density:** 2 cases support it.	**Example:** "What distinguishes the i-Score method is the mechanism it uses for determining an empirical upper limit of interoperability for those systems that support the operational process" (REZAEI et al., 2014)	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that cites them, varying only the amount of information provided, not the definition of the model itself.
IAM: Interoperability assessment methodology - introduces nine components, which are: requirements, node connectivity, data elements, protocols, information flow, information utilization, interpretation, latency, and standards. The nine components each included either a "yes/no" response or a mathematical equation.		
Groundedness: 2 segments are coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "The interoperability assessment methodology model introduced nine components." (REZAEI et al., 2014)	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that cites them, varying only the amount of information provided, not the definition of the model itself.
LISI: Levels Of Information Systems Interoperability - with five interoperability levels (0–4) are Isolated, Connected, Functional, Domain, and Enterprise, in which each interoperability level exists in a specific environment. The attributes of the LISI Reference Model contain Procedures, Applications, Infrastructure, and Data.		
Groundedness: 6 segments are coded in this subcategory that represents one code. **Density:** 4 cases support it.	**Example:** "LISI defines five levels of interoperability across four attributes. The levels range from "isolated level" to "enterprise level" and	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that

	the attributes are Procedures, Applications, Infrastructure, and Data (PAID)." (WYATT; GRIENDLING; MAVRIS, 2012)	cites them, varying only the amount of information provided, not the definition of the model itself.
OIMM: Organizational interoperability maturity model - this model extends the LISI model into the more abstract layers of command and control support. Describing the ability to interoperate, five levels of organizational maturity are defined in this model (independent, *ad-hoc*, collaborative, combined and unified). Preparedness, Understanding, Command style, and Ethos.		
Groundedness: 2 segments are coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "This model extends the LISI model into the more abstract layers of command and control support." (REZAEI et al., 2014)	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that cites them, varying only the amount of information provided, not the definition of the model itself.
QIM: Quantification of interoperability methodology - They stated "interoperability of systems, units, or forces can be factored into a set of components that can quantify interoperability" and they identified the seven necessary components as languages, standards, environment, procedures, requirements, human factors, and media.		
Groundedness: 3 segments are coded in this subcategory that represents one code. **Density:** 2 cases support it.	**Example:** "… introduced a method called The Quantification of Interoperability Methodology which forms the foundation for the levels of information systems interoperability model." (REZAEI et al., 2014)	**Discussion:** we understand that the mentioned models are complete works, leading to an understanding regardless of the work that cites them, varying only the amount of information provided, not the definition of the model itself.

C.1.9 Attributes

Table C.1.9 - Category: Attributes

Category: Attributes
Interoperability Mechanisms: decisions to be taken and mechanisms recurrently used throughout the levels to achieve interoperability.

Groundedness: 86 segments are coded in this subcategory that represents 18 codes. **Density:** 15 cases support it.	**Example:** "Methods and techniques like interface description languages, adaptors, wrappers, middleware and middleware bridges have quite successfully been applied " (RUOKOLAINEN; KUTVONEN, 2006)	**Discussion:** the excerpt in example represents our understanding that the different needs required for each perspective, purpose or level should be addressed in a suitable manner. A middleware acts as an intermediate layer between the systems by standardizing the interface between them. It is a mechanism widely used but does not assist the concepts compatibility required by the semantic level.
General Attributes: typical systems' attributes related to interoperability like availability and compatibility.		
Groundedness: 35 segments are coded in this subcategory that represents seven codes. **Density:** 11 cases support it.	**Example:** "Incompatibility is the fundamental concept used in defining the scope of interoperability domain. It is the obstacle to establishing seamless interoperation. " (CHEN; DOUMEINGTS; VERNADAT, 2008)	**Discussion:** Incompatibility as the opposite of the compatibility attribute, desired in any system aiming to be interoperable.

C.1.10 Context

Table C.1.10 - Category: Context

Category: Context		
Business Information: concerns with the context information that guides the interoperability levels in the business (organizational) perspective bringing significance to the interactions.		
Groundedness: 1 segment is coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "Interoperability is considered as significant if the interactions can take place at least on three different levels: data, services and processes, with a semantics defined in a given business context"	**Discussion:** like perspectives, different domains look distinctly to concepts (for instance, the term *client* for a bank and a server). The business context

	(CHEN; DOUMEINGTS; VERNADAT, 2008)	is crucial for interoperability, particularly at semantic level.
Extrinsic Environment Information: definition of extrinsic: a) not forming part of or belonging to a thing; b) originating from or on the outside; c) originating outside a part and acting upon the part as a whole. It refers to variables like position, time, weather, speed or any kind of information that can be perceived by the system or acquired by a sensor.		
Groundedness: 1 segment is coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "Obviously, a service may deliver different results depending on various context information. For example, the product of a service may satisfy user's need better when it is invoked under 0 °C" (FANG; HU; HAN, 2004)	**Discussion:** acquiring the information through the sensor is the first step, but the data should be reliable, consistent and comprehended. For instance, based on the user's position his device can be automatically connected with others.
Interaction Time: it is related to the idea that the interaction time is a context variable that can influence the interaction among entities, affecting for instance: a) connectivity (e.g. the link should be available at any moment); b) dynamicity (e.g. information is dynamic, can change meaning and form over time).		
Groundedness: 5 segments are coded in this subcategory that represents two codes. **Density:** 2 cases support it.	**Example:** "As the node connectivity is a variable dependent on time, that is both discrete and continuous time intervals, it can be said that it is a disturbing element to measure the interoperability amongst others." (REZAEI et al., 2014)	**Discussion:** Connectivity is imperative for interoperability. All the entities should be in continuous operation through all the interaction. If the connecting channel, like a network, is operated intermittently, the interoperability is impaired.
Intrinsic Environment Information: definition of intrinsic: a) belonging to the essential nature or constitution of a thing; b) originating and included wholly within an part. The environment out the "operator" control on which the entities are interacting can influence the interoperability.		
Groundedness: 5 segments are coded in this subcategory that represents one code. **Density:** 4 cases support it.	**Example:** "For any interoperable system the operator has control of the medium and equipment; the environment represents those items which are outside the	**Discussion:** despite all the mechanisms and infrastructure required by to systems to interact, their internal context, as we understand here as intrinsic environment information, can

	operator's direct control." (REZAEI et al., 2014)	impair the interaction. Variables like memory consumption and processing speed are relevant for a given interaction, and out the operators control.
User knowledge: the user's background can affect interoperability as knowledge compromises the user's comprehension and perspectives regarding a system.		
Groundedness: 1 segments are coded in this subcategory that represents one code. **Density:** 1 case support it.	**Example:** "The same problem can bring quite divergent views because of users' knowledge context." (FANG; HU; HAN, 2004)	**Discussion:** Semantic interoperability problem lies in the different understanding of the shared information. Different users have different comprehension and needs. Based on their knowledge they can judge if information is correct and choose the most suitable services for their needs.