



HAL
open science

A conceptual perspective on interoperability in context-aware software systems

Rebeca Campos Motta, Káthia Marçal de Oliveira, Guilherme Horta Travassos

► **To cite this version:**

Rebeca Campos Motta, Káthia Marçal de Oliveira, Guilherme Horta Travassos. A conceptual perspective on interoperability in context-aware software systems. *Information and Software Technology*, 2019, 114, pp.231-257. 10.1016/j.infsof.2019.07.001 . hal-03574522

HAL Id: hal-03574522

<https://uphf.hal.science/hal-03574522v1>

Submitted on 3 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Conceptual Perspective on Interoperability in Context-Aware Software Systems

Rebeca C. Motta, Káthia M. de Oliveira, and Guilherme H. Travassos

Abstract—Context: Context-Aware Software Systems can interact with different devices to complete their tasks and act according to the context, regardless of their development and organizational differences. Interoperability is a big challenge in the engineering of such systems. **Objective:** To discuss how interoperability has been addressed in Context-Aware Software Systems, strengthening the scientific basis for its understanding and conceptualization. **Method:** A *quasi*-systematic literature review was undertaken to observe interoperability in such Context-Aware Software Systems to support the discussions. Its dataset includes 17 from 408 papers identified in the technical literature. The extracted information was qualitatively analyzed by following the principles of Grounded Theory. **Results:** The analysis allowed to identify ten interoperability concepts, organized into a Theoretical Framework according to structural and behavioral perspectives, which deals with interoperability as the ability of things (an object, a place, an application or anything that can engage interaction with a system) to interact for a particular purpose, once their differences (development platforms, data formats, culture, legal issues) have been overcome. Once the interoperability is established from structural concepts (context, perspective, purpose, the level of provided support and system attributes), it can be measured, improved and observed from the behavioral concepts (evaluation method, challenges, issues, and benefits). **Conclusions:** The Interoperability Theoretical Framework provides relevant information to organize the knowledge related to interoperability, considering context, and can be used to guide the evolution of software systems regarding changes focused on interoperability.

Index Terms—Interoperability, Context-aware software systems, Internet of Things, Contemporary Software Systems, Ubiquitous Computing, *quasi*-Systematic Literature Review, Evidence-Based Software Engineering

1 INTRODUCTION

Internet of things, smart cities, smart homes, ubiquitous computing, and many other contemporary software systems (CSS) have ceased to be just expectation and have invaded our daily activities. These types of software-based systems put interoperability issues as a priority for both the researchers and practitioners. Sensors, computer devices, and different applications should interact, exchange information, and consider distinct elements to guarantee the supporting of our everyday activities natively. Moreover, in this ecosystem, the software applications become context-aware and must behave according to the context in which the users expect the best possible support for their tasks.

Context is any piece of information that may be used to characterize the situation of an entity (logical and physical objects present in the systems' environment) and the relations relevant for the actor-computer interaction [1]. An actor can be a person, place, or object relevant to the interaction with the software system [2]. Context-aware Software Systems (CASS) are a type of CSS. They use information about the actor, its behavior, and their inserted environment to provide adequate services and enhance the user experience. From that, a context-aware software system can perceive, understand, and use contextual information to its self-adjustment to the current situation [3]. Context awareness is one of the characteristics of ubiquitous and pervasive systems, often related to autonomy, self* capabilities, and smartness. However, a ubiquitous or pervasive system has other characteristics such as the omnipresence of services, invisibility, the capture of experiences, the composition of functionalities, and others [4]. In this paper, the primary focus is on context-awareness

systems and their interoperability issues with non-human actors.

CASS usually use distinct sensors and interoperate with distinct applications to enact context-awareness, which configures the interactions between a CASS and its ecosystem. Different types of actors, interactions, and relevant context should be considered while performing the system functionalities. CASS 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. Consequently, interoperability becomes essential since such systems use different types of computing technologies interacting to provide expected behaviors. In turn, 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.

For example, traffic information collected from sensors in a smart city can adjust the traffic lights for an ambulance concerning the adequate route in a proper time. With the combination of different software systems, new mashup services arise, and it is possible to improve existing ones through interoperability. Context information, semantics, and interoperability are often combined [5] since the data collected within each system needs to be represented with their meanings, that way other services from different domains can interpret and use such data.

The interoperability characteristics should cope with heterogeneity issues, aiming at making the software systems compatible and harmonious, and is still even more relevant when considering CASS sensors and more complex objects.

However, the heterogeneity of current computing and software technologies makes interoperability an exciting engineering challenge [6].

Different definitions describe interoperability for conventional software systems, such as:

- a) The degree on which two or more systems, products or components can exchange information and use such information [7];
- b) The ability of two or more object request brokers to cooperate to deliver requests to the proper object [8];
- c) The capability to communicate, execute programs, and transfer data among various functional units in a manner that requires the user to have little or no units' characteristics knowledge [9].

However, do these traditional definitions support the understanding of CASS interoperability? How to deal with interoperability issues in CASS? What should be considered when developing and evaluating CASS regarding interoperability?

To address these questions, we decided to perform an investigation on CASS interoperability, with the idea of dealing with it in a broader sense. In other words, we aimed to analyze conceptual issues associated with CASS interoperability by observing how they interrelate complementarily and not to delve into a specific sort of computing or software technology.

To that end, we start by performing a quasi-Systematic Literature Review (qSLR) [10] to characterize the current state of research regarding interoperability for CASS. Such characterization involves identifying interoperability definitions and other relevant concepts related to it. Then, from the results of the qSLR, we collected a set of pre-defined data that was analyzed with the Grounded Theory (GT) methodology [11]–[13]. As a consequence of this analysis, we obtained an Interoperability Theoretical Framework (ITF) [13] of structural and behavioral concepts concerned with interoperability that should be taken into account while developing and evaluating CASS.

The remainder of this paper is organized as follows. Section 2 briefly introduces our investigation strategy, and Section 3 presents the background of the main topic related to this study. Then, the next three sections present, respectively the qSLR (section 4), the qualitative analysis with GT (section 5) and the discussion of the results answering the research questions (section 6). Section 7 an example of the conceptual framework proposed, and Section 8 presents an overview of current challenges for interoperability. The limitations and threats to validity are presented in Section 9. Finally, in Section 10, we present the conclusions and discuss future directions of our research.

2 INVESTIGATION STRATEGY

To perform our research, we defined and followed an investigation plan composed of four main activities, as



Fig. 1. Investigation Strategy.

illustrated in Fig. 1. The first activity, research background was defined to set down the main concepts related to context-awareness and interoperability; and to look for other literature reviews on the same subject (Section 3 presents the result of this activity.)

The second activity is dedicated to investigating issues regarding interoperability in CASS by performing a *quasi*-systematic literature review (qSLR) [10], [14]. We classify our work as a qSLR because we have no means to compare outcomes or apply meta-analysis. A systematic literature review is a type of secondary study [15] 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. To perform the qSLR two main steps were structured based on [16]: *Planning* to define the research protocol; and, *Execution*, where after some trials refining the search string, papers are selected considering the criteria defined in the protocol. Following the same approach used in other SLRs [1], [17], we used the SCOPUS database for the selection of papers. One can raise that this could suggest a possible limitation in the search space. However, working with SLR for more than fifteen years [18], we are convinced that Scopus combined with backward and forward snowballing procedures [19] can mitigate the lack of search engines and provide a representative set of papers to a characterization work. Moreover, Scopus mostly overlaps other search engines, such as Web of Science [20] and other engines present different limitations (e.g., IEEE Xplorer was limited to 15 search terms, and the ACM Digital Library and Google Scholar do not favor repeatability). Therefore, for our final selection, we combined the Scopus results, applied on 17 July 2015 with snowballing procedures (backward and forward) executed till April 2018. These steps are detailed in Section 4. We also performed an initial analysis to provide a general overview of the data retrieved. However, to draw a better view of the findings to answer the proposed research questions, a broader interpretation of this information was required, where Grounded Theory was used.

Grounded Theory (GT) supported our third research activity. GT is an inductive methodology that originated in

the 1960s in the field of health studies, which can be used with qualitative or quantitative data [11]. Even that the GT appeared more than fifty years ago, it is used in several domains, such as software engineering area [12], [21], [22] for building a body of knowledge about topics of interest showing its adequacy to this domain. In our case, GT suits well the characterization studies with broader research questions, such as the ones presented in the qSLR (section 4), as it aids in the interpretation and clarification of the results.

We based our procedure on textual analysis, using codes to assign concepts to a portion of data following the approach developed by Strauss [13]. This approach is considered the most widely used for qualitative research in the Software Engineering area [23]. It is composed of three steps: **Open coding**, to identify concepts, comparing similarities and differences by assigning codes from excerpts of data identified in the text; **axial coding** to reassemble data that fractured during open coding in categories; and **selective coding** where a core category is identified that expresses the central idea of the research. Then, we relate each category previously defined in the axial coding to this core category defining the framework for interoperability for CASS. All papers selected in the qSLR was used for the GT analysis (Section 5 presents the GT procedures in detail as well as the resulting defined framework.)

Finally, the fourth activity is the *final analysis of the results* revisiting the research questions considering the qualitative analysis and codes resulted from the application of the GT and analyzing the operationalization of the interoperability framework (Sections 6 and 7 present these results.)

3 RESEARCH BACKGROUND

This section starts by presenting a brief description of the Context-Aware Testing for Ubiquitous Systems (CAcTUS)¹ Project, which motivated this research. Then, we present some basic concepts related to context-awareness characteristic and interoperability that are the ground for the following discussions and the results presented in this paper.

3.1 The CAcTUS Project

This Brazilian National Council for Scientific and Technological Development (CNPq) research project was performed based on the aim of understanding test strategies for the quality assessment of actor-computer interaction in context-awareness systems, as it is one of the chief characteristics of ubiquitous systems [1], [24], [25]. Three universities conducted the project: Universidade Federal do Rio de Janeiro, and Universidade Federal do Ceará, in Brazil, and Université Polytechnique Hauts-de-France in France.

One of the project goals was to organize a body of knowledge from which the researchers conducted secondary

studies to present the state-of-the-art and reveal evidence regarding CASS. Three qSLR shared the definitions presented in [26] and aimed to investigate test case design for CASS [1], testing methods for CASS [17], and interoperability for CASS, focusing on the different interactions and different actors in context-aware applications (presented in this work).

This qSLR focuses on how interoperability is considered in CASS. The initial research protocol was the same as the other two; however, due to differences in the intervention, we evolved some research protocol features, extending the population to cover different concepts as well CASS. Further information on this review is discussed in the next sections.

3.2 Context-Awareness

Context-awareness is one of the characteristics of ubiquitous and pervasive systems [27]. Ubiquitous computing aims to develop computing technologies seamlessly integrated with objects, so they become indistinguishable [28]. It seeks to create useful systems to be present in different situations encountered in the real world and observe their impact from the user's perspective [29], [30].

The use of context has been discussed in the technical literature for some time now, with several definitions [27], [2], [31]–[34]. In the CAcTUS project, we amended its definition to enhance the interaction between actors and computers importance, considering the term **actor-computer interaction**, which is used in this work. Moreover, our interpretation is that “context-awareness” is a dynamic property representing a piece of information, which can evolutionarily affect the overall behavior of CASS in the interaction between the actor and the computer [1].

A CASS continuously monitors the world information around it and uses such information to enhance its natural behavior or as a trigger to its adaptation [2]. Applications regarding the Internet of Things, smart cities, automated manufacturing, intelligent environments, and many others rely on the interaction of different technologies with various devices and need to discover services and resources and communicate with various other computing resources. Multiple devices (sensors, computers, mobile technologies) and services should interact automatically to achieve the envisioned objectives. However, these applications may have been constructed with a wide variety of development platforms, middleware systems or programming languages, or have different communication interfaces. All of these represent challenges for their interoperability.

3.3 Interoperability

Interoperability is a critical software systems' property [35] and is currently referred to as a significant technical challenge [25],[36]. This challenge is even more evident considering that Weiser's' vision of ubiquitous computing is

¹ <http://lens.cos.ufrj.br/cactus/>

turning into reality with the increasing number of systems developed for universities, hospitals, and houses [37], [38]. To deploy applications in such different fields for the most different purposes, it is necessary to support a variety of components with features and functions suited to the application objectives. The difficulty lies in achieving interoperability between things, components, and systems. In this scenario, middleware interoperability solutions have been developed, evolving from one-to-one bridging to interoperable platforms and transparent interoperability approaches [39].

Initially, the concept of interoperability was limited to information systems, but with the progress of research and interest of several organizations, it became harder than initially imagined. Middleware research has provided different approaches (such as Jini2, UPnP3, LonWorks4, HAVi5, Dojot6, and others) seeking to provide a high-level abstraction with flat configuration, to make services available without high development effort [40] by enabling interoperability. Alongside with middleware and other technical solutions, standardized modeling approaches have been used to deal with interoperability. This alternative relies on the use of a common language with formal or semi-formal models, for example, model alignment, model integration, reverse modeling and human modeling that can facilitate a system specification [41], [42]. The work [43] is related to model-based approaches reports the interoperability challenge for enterprise information systems, since the systems are not built focusing in interoperability. Therefore, despite the different initiatives, interoperability still has several challenges. For this reason, it is essential to understand the general view of interoperability, starting with its definitions, differences, and similarities between the studies we came across throughout this work.

Terminology problems [44] (like inconsistency and out-of-date information) are recurrent in this research area, and, as expected, there have been several definitions of interoperability proposed by researchers, standards bodies or governments over the years. For instance, Ford et al. [45] identified thirty-four distinct definitions for interoperability. In that work, the authors give an analysis and history of interoperability also presenting some research on interoperability evaluation methods. In conclusion, they offer some recommendations for the field of interoperability measurement research, as to pursue mathematical methods, but they also encourage the researchers to extend the existent body of knowledge.

The challenge of interoperability and the need to develop more interoperable solutions motivates different initiatives such as INTEROP Network of Excellence (INTEROP NoE) focused on research and development for Architectures and

Platform, Enterprise Modeling and Ontologies at the European Level [46]. For them, there are numerous gaps between the existing paradigms and the interoperable systems. These gaps pointed out by the initiative shows different research and investigation opportunities. In our present work, we seek to contribute with the research looking at the CASS domain, in order to complement the existing knowledge.

“The ability of two or more systems or components to exchange information and to use the information that has been exchanged” from IEEE [35] is one of the most popular definitions referring to interoperability [45]. From a systemic perspective, a system is a set of parts forming a unified whole; these parts are named as components in this paper. The understanding of components includes different points of view, as it can go from small modules to software services in an enterprise, for example. However, this definition is limited to information exchange. The complexity and evolution of today’s software systems being smarter and bigger can lead to a discussion of whether it is only information when concerning perception or recognition, and the information comprehension as well. We argue that the existing definitions and solutions should cover the concept of actor-computing interaction to address CASS interoperability concerns.

The interoperability comes as an essential requirement to cope with heterogeneity and dynamicity, specific challenges for CASS. It aims at making compatible and harmonious software systems and devices regardless of the differences in infrastructure development, directly addressing their heterogeneity. Aside from surpassing heterogeneity issues, it is also necessary to overcome the dynamicity challenges (for instance the information from a user is dynamic, it can change its meaning and form over time [47]) that come with the changes in the interaction context. Therefore, interoperability should also be achieved considering possible adaptations in the software system behavior.

3.4 Literature reviews and discussions

Before the qSLR execution, a search was performed in 2014 looking for other literature reviews dealing with interoperability in CASS. To the best of our search efforts, no much works were investigating both subjects together and providing aggregated results. Nevertheless, we found three essential works: two specifically about interoperability [45], [48] and one about context-aware systems [49].

As presented in the previous section, Ford et al. [45] identified thirty-four different definitions of interoperability. They offer several definitions of general interoperability concepts, and some of them define what they consider specific interoperability types (technical, logistic, and operational interoperability). The presented definitions are

² Sun Microsystems. “JINI Architecture” - www.sun.com/jini

³ UPnP Forum. “Universal Plug and Play” - www.upnp.org.

⁴ Echelon Co., “LonTalk Protocol Specification” – goo.gl/Rq8rDr.

⁵ The Havi Organization. “Havi Version1.1 Specification” – www.havi.org.

⁶ Dojot – www.dojot.com.br

widespread through other technical papers and based on their references; they extract some interoperability characteristics as a foundation for a proposed methodology for measuring interoperability, called the “Interoperability Score.” Rezaei et al. [48] performed a systematic literature review focusing on interoperability evaluation models. They presented ten different models and compared them regarding twelve distinct issues at various granularity levels of technical, syntactic, and semantic interoperability. These works [45], [48] show that there are still differences in the understanding and a lack of consensus on the concept, despite the research-extension associated with the interoperability topic.

Perera et al. [50] present a literature review considering context-awareness from the Internet of Things perspective. The authors define the Internet of Things and possible application domains, together with some research gaps and show the relationship between the context and this paradigm. Middleware solutions are considered a fundamental problem, especially for sensors network, when it is necessary to transfer data and to incorporate different solutions in one system. Furthermore, interoperability is presented together with other several relevant topics. The literature review is not specific for interoperability.

The review by Hong et al. [49] is specific for CASS and suggests a new classification framework for those systems based on their architecture. It consists of five layers: concept and research, network, middleware, application, and user infrastructure. Despite addressing the topic of context-awareness and considering the interaction of different devices, it does not expressly present a view of interoperability.

These initial literature reviews motivated us to contribute to the field by filling the research gap and merge these two topics of interest: interoperability and CASS.

More recently, an interesting discussion on interoperability for the Next Generation of Enterprise Information Systems has been offered by Panetto *et al.* [51]. The authors argue that such systems can be “a key technological enabler of complex, loosely connected or disconnected functional networks of devices.” However, there are many challenges to be faced, such as the dynamic reconfigurability, smart reconfiguration, automatization of ontological structures, services adaptation, and interoperability. It reinforced our perception of the importance of doing more investigations on the topic.

4 QUASI-SYSTEMATIC LITERATURE REVIEW

This activity is dedicated to investigating issues regarding interoperability in CASS by undertaken a *quasi*-systematic literature review (qSLR) [10] following three main phases: planning, execution, and analysis.

In the *planning phase*, two researchers were responsible for defining the research protocol, including research questions, search string, inclusion and exclusion criteria,

quality criteria, database, and tool support. Two external researchers revised the protocol. Table 1 presents a summary of our research protocol.

Our research goal (organized based on [52]) is defined as: *Analyze interoperability for the purpose of characterizing with respect to its concepts, attributes, and evaluation methods from the point of view of software engineering researchers in the context of context-aware software systems.*

From this goal, we set three research questions (see Table 1). The first question (RQ1) intends to identify how the works on CASS discuss or deals with interoperability, how it has been integrated with all aspects concerned with the context and its meanings. The second question, more precise one (RQ2) to identify all elements (or issues) concerned with interoperability and that should be taken into account to cover it in CASS. Also, the third question (RQ3) focusses on interoperability evaluation and all elements, characteristics, and features involved in its quality assessments. Based on that, we defined the search string using the PICO strategy [53]. For that, Population was defined as CASS, which means considering all papers that deal with CASS, setting all words that can characterize context-aware systems. The Intervention is interoperability (our focus); Comparison is empty due to a qSLR has no baseline for comparison; and, Outcome is set as all elements (characteristics, features, models, among others) and evaluation issues of interest.

Inclusion criteria were explicitly defined to assure that the paper discussed interoperability in CASS eliminating those that have words in the abstract but do not explicitly deal with the subject. To not lose any relevant work, we set as exclusion criteria only documents not representing scientific papers or not available.

We performed four trials until setting our final search string (see Table 1). In the first trial, the original structure and first search string were used based on a previous literature review conducted in CACTUS [26]. The objective was to look for primary sources of information related to interoperability regarding CASS, to form a first gold standard. In the second trial, after string tuning, it was possible to observe the difference between what we want to find (concepts and aspects describing interoperability in CASS) from the returned results (interoperability used as a buzzword). Due that, it has been decided to extend the search (trials 3 and 4), to see how interoperability is addressed in context-aware systems and how to evaluate it, and not only concepts and aspects.

During the execution some articles were found considering “*assisted living*” systems, “*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 with the research considering this perspective. We did not consider these new terms as synonyms of the initial target population (context-aware systems), but they were used to extend the research range and reach different concepts related to CASS.

Table 1. Protocol Summary.

Research questions	(RQ1) How is interoperability discussed in CASS? (RQ2) Which are the interoperability elements in CASS? (RQ3) How to evaluate interoperability in CASS?	
Search string	Population	("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
	Intervention	(interoperability OR interconnection OR interoperation OR interaction OR integration OR exchange) AND
	Comparison	-
	Outcome	("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")
Search Strategy	SCOPUS (www.scopus.com) + Snowballing (backward and forward)	
Inclusion Criteria	- to talk about interoperability; - to discuss systems including context-awareness attributes or applied to CASS.	
Exclusion Criteria	- not available for retrieval; - studies in duplicity; - register of proceedings.	
Study type definition	Articles presenting an example, application, proof of concept or study related to interoperability in CASS.	
Acceptance criteria	Four distinct readers: - all readers accept => paper is accepted - all readers exclude => paper is excluded - the majority of accept, others in doubt => paper is accepted else => discuss and consensus	
Tool	JabRef (www.jabref.org)	
Research Material	- Detailed information about the review planning and execution available at https://bit.ly/2SoF1uS . - GT analysis package with QDA Miner Lite available at https://bit.ly/2SnUA61 . - Detailed analysis and discussions available at https://bit.ly/2S2NUFg	

The terms were submitted separately as population, with intervention and outcome fixed in the search string, and re-submitted to retrieve publications.

In the *execution phase*, the search using the final search string presented in Table 1 reached 408 articles, being 42 proceedings registers, which were removed considering the exclusion criteria. The acceptance criteria presented in supported the selection of papers.

From the 16 selected studies, backward and forward snowballing sampling [15] were performed following the guidelines described in [19]. For the backward snowballing, one researcher checked the references in the articles. If their titles seemed appropriate for the review, they were considered. For the forward snowballing, one researcher searched the citations to each article and read the titles regarding their suitability for this study. They were evaluated against the selection criteria by the other researchers; whether acceptable, it was selected for a full reading.

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. The selected studies were evaluated against quality criteria. It is essential to state that quality in our evaluation is concerned with the adequacy, or to what extent the technical paper can contribute to the objectives of our research (see the questions for quality criteria in Table 2).

We decided to focus our quality criteria on interoperability since it is our primary focus of this research. This quality assessment is not regarding the trust in the source ("evidence strength").

Since there was no upper limit (e.g., if the study was validated in several settings, it counts as 0.5 points each), we defined as a cut-off point the first quartile, calculated in 3 points. We kept, therefore, only papers with a score above this cut-point (**Fig. 2A**). After the quality assessment, 17 studies compose the final set of selected papers ([47], [48], [54]–[68]). Considering the general assessment results (**Fig. 2B**), 70.5% of the selected studies present an interoperability definition and its dimensions.

We can observe a gap between the proposals and what is evaluated since 76.4% of the studies present interoperability elements, but only 41.1% offer some evaluation. Moreover, only 11.7% describes an empirical assessment in the correspondent paper. From this, we understand that the field still misses the reporting of empirical evidence considering an evidence-based perspective.

With the quality assessment, we intended to observe the adequacy and contributions of each study considering our research goal. There is no interest or reason for making any claim regarding their overall quality. Finally, based on the data collected with the extraction form, we proceeded with the **analysis phase** using GT procedures as presented in the next section.

Table 3. Data Extraction Form Example.

Field	Description
Reference	[44]
Abstract	This paper presents an assessment model of service interoperability for service composition, which helps users to judge whether their composite services can be invoked properly.
Interoperability definition	“Ability of two or more software components to cooperate despite differences in language, interface, and execution platform.”
Interoperability dimension	Interoperability is usually sub-classified to signature level, protocol level, and semantic level. They also include quality level and a new context level in Ubiquitous Computing Environments.
Interoperability perspective	Web Services rely on some simple, yet extensible standards and protocols - WSDL, SOAP, UDDI, and so on. The characteristics such as self-described, standard-based and HTTP binding will make web services easier to resolve basic interoperability problems including platform heterogeneity, program language heterogeneity, and operating system heterogeneity, among others.
Interoperability characteristic	Syntax consistency of service description, registration and invocation; Standards; Order in which a service expects its operations to be invoked; Compatible understanding of the semantic; Satisfy quality constraints; Context information.
Evaluated attribute	The proposed evaluation is for calculating the interoperability degree, regarding the interoperability levels.
Interoperability measures or method	In the following, we give a reference assessment model of service interoperability by fuzzy quantization of the interoperability.
Pre-existent approach	“Since 2002 we have worked on a service composition project called FLAME2008.”
Conditions or Restrictions	“As our research is at an early stage, we just calculate the interoperability between two services. Multiple services’ interoperability is our future work.”
Software systems category	Application software - Service based systems
Development target	Web Services
Experimental study	Basic research
Experimental data	Not applicable

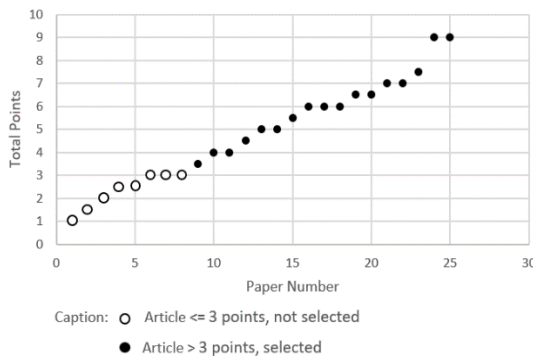


Fig. 1. Cut-off point

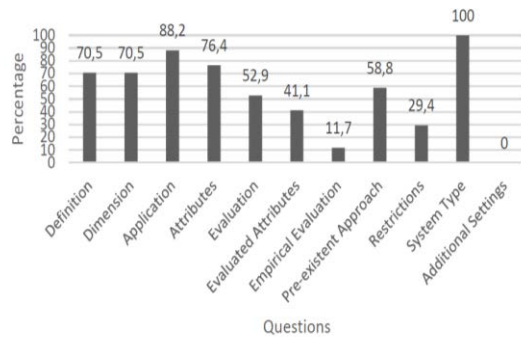


Fig. 2. General Assessment Results.

Table 2. Quality Evaluation 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 (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
Is the interoperability evaluation described? (1 pt.)
Does the interoperability evaluation include the proposed attributes? (1 pt.)
Is there an empirical assessment of the interoperability approach? (1 pt.)
Criteria related to the background or applicability of the approach
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 the interoperability approach generalization
Is there any description of the interoperability approach application in additional settings? (0.5 for each setting)

5 QUALITATIVE ANALYSIS

We based our procedure on textual analysis. It uses codes to assign concepts to a portion of data following the GT approach developed by Strauss [13] composed of three steps: open coding; axial coding, and selective coding.

To support these steps, we considered the groundedness (g) and density (d) metrics proposed in [11]. The groundedness indicates how many excerpts are related to one particular code. The density shows how many articles support it. We consider recommendations with low groundedness and density (such as g:1; d:1) as evidence. They were chosen to increase the number of observations in this analysis. With these metrics, we can observe that some subcategories are more frequent than others, leading to stronger concepts due to more available data to ground them.

All the matching from text to code, in the open coding and axial coding steps, were executed by two researchers and then integrally revised by a third one.

The 17 papers selected from the *q*SLR and presented in the previous section provided a significant amount of data to be analyzed (e.g., several elements related to interoperability, definitions, and dimensions). The GT methodology comes as a mechanism to understand data and their relationships, considering the application domain and systems features. The principles and procedures of GT were used to assist us in developing and analyzing the concepts related to interoperability in CASS.

The QDA Miner Lite tool was used to support the process. It is a free version of 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. The analysis of data started after the first selection of papers and finished after evaluating all data resulted from the snowballing procedures, following subsections present in detail each coding step.

5.1 Open coding

The open coding process followed the process presented in Fig. 3. In the beginning, one researcher **identified** the

relevant excerpts by marking (using the QDA Miner tool) all the excerpts related to an interoperability definition. The excerpts could be a word, a phrase, or a paragraph relevant to the concept under observation. To uncover and develop a concept, the researcher should keep an open mind and report the thoughts related to it. After analyzing the excerpts, two researchers defined the codes. The constant comparative analysis is a standard procedure to execute in the coding activity continuity. Whenever coming across another excerpt that seemed to describe the same concept or share a common attribute, these were grouped into the same code.

This activity requires some abstraction since the code should be written in such a way that it represents the grouped excerpts. Table 4 presents an example of code. It includes some excerpts from the original texts composing the code, which in turn is related to the definition of interoperability.

Table 4. Open coding example.

Data Excerpts	Defined Code
<i>"It refers to make different data models and query languages working together"</i> [45].	Working together despite differences
<i>"Ability of two or more software components to cooperate despite differences in language, interface, and execution platform"</i> [44].	
<i>"To be able to work together with both different cloud services and providers, and other applications or platforms that are not cloud dependent"</i> [39].	

After defining codes for all information extracted from the papers, a third researcher, who did not participate in the coding process, reviewed all selected excerpts and corresponding codes. The researcher reviewed each extraction and the respective code in the order they appeared, contributing to their constant comparison and marking for each code one of the following options: agreement, partial agreement, and disagreement.

We obtained 85% of an agreement, 5% of partial agreement and 11% of disagreement, showing that in general, the coding process was adequate. Consequently, the three researchers participated in meetings for discussion and consensus. Every match was then classified and justified by the reviewers as follows:

Agreement - in this case, we kept the original code.

Example: Excerpt: *"It refers to make different data models and*

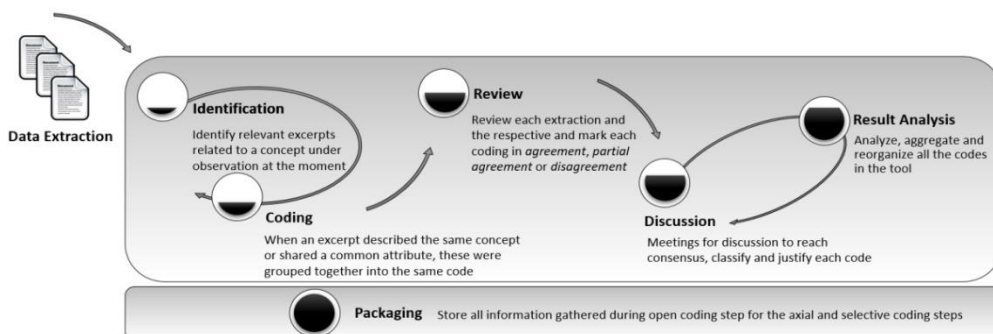


Fig. 3. Coding Process Activities.

query languages working together.” [55]. Initial and Final code: Working together.

Partial agreement – modified the code to represent the excerpts better. **Example:** Excerpt: “*the ability of systems, units, or forces to provide data, information, material, and services to and accept the same from other systems, units, or forces and to use the data, information, and material, and services so exchanged to enable them to operate effectively together*” [65]. Original code: Ability of systems to provide, accept, and use resources. Discussion: the resources term generalization, as it stands for information and material, is not at the same abstraction level. Final code: Ability of systems to provide, accept, and use resources and information.

Disagreement - in this case, suggested other code for the excerpts. **Example:** Excerpt: “*Interoperability can be considered as being a problem, which can only arise when some resources are put together to inter-operate. Because such resources of a system are themselves systems, interoperability simply concerns relations between systems.*” [62]. Original code: Relation between systems. Discussion: the stronger concept should be “resources are put together to inter-operate” leading to cooperation. Final code: Ability to cooperate.

Finally, the first researcher reorganized all the codes in the tool. During the **analysis**, from the 17 selected papers, we identified 411 relevant excerpts labeled with 109 codes in the final of open coding.

5.2 Axial coding: Context and Interoperability

In the axial coding process, we related the codes, including them into 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, and dimensions. The subcategories, together with their respective codes, are part of the category under consideration whole concept.

To perform this activity, we analyzed 109 codes looking at their semantics, which means, what is the main subject they relate, allowing the definition of subcategories. This procedure of establishing subcategories is not a following analytic activity, once the ideas of categories come to mind throughout the open coding process. We can illustrate it with the following algorithm:

```
For i=1 to 109
  For j=i+1 to 109
    Analyze the code (i) against the code (j)
    If they have the same meaning verify if they were
    already associated with a subcategory
      If subcategory exists then
        Analyze if the sub-category also represents
        the code being evaluated and if necessary
        generalize/refine the code or the
        subcategory to consider the new concept
      else
        create a new subcategory labeling in such
        way that considers code (i) and code (j)
      endif
    endif
  endfor
endfor
```

After that, the process of reassembling subcategories into categories is quite similar to the algorithm previously presented, noticing that the focus of analyses is the defined subcategories rather open codes. During the definition of subcategories, potential ideas of categories already emerged in the analysis, making the last step simpler.

Based on this procedure, the 109 codes were grouped into 41 subcategories later organized into ten categories. In the next subsections, we present two of these categories with their related subcategories: Context Category and Interoperability Definition Category. The other categories (Interoperability Level, Purpose, Perspective, Benefits, Issues, Challenges, Evaluation Methods, and Attributes) are detailed similarly in Appendix A. All ten categories followed the same analysis procedure, and this division was due to organization purposes, remaining the primary categories for our discussion.

5.3 Context Category

This research considers context as a new factor, and our expectation, in the beginning, was to observe how the context can influence the interoperability of software systems, adding complexity and overlooked difficulties. Based on the definition provided, **context** is any piece of information that may be used to characterize the situation of an entity (logical and physical objects present in the environment) [1].

In the analysis, 11 excerpts were associated with five open codes equivalent to five subcategories. Fig. 4 shows the Context Category representation. These codes were grouped into five context variables, identified in the study: **business information** (g:1; d:1); **extrinsic environment information** (g:1; d:1); **interaction time** (g:3; d:2); **intrinsic environment information** (g:5; d:4); and **user knowledge** (g:1; d:1). **Business information** and **user knowledge** extend the initial definition of context since they do not necessarily come from devices. However, it is essential to consider, as the analysis presents, due to their contribution to semantic interoperability.

Based on the data analysis, context is presented in the studies as a characterizing factor since the analysis performed, different excerpts were associated with context (coded). However, it was not considered an influencing factor once the influence value is not apparent (e.g., positive or negative). Regarding expectations, we observed that each variable information is retrieved in a particular manner, for example, through sensors. Also, each variable contributes differently to the system; for instance, it can use the data from the sensors to retrieve context information and to trigger a predefined action.

Excerpts	Open Code and Subcategory
“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.” [45]	Business Information
“Refers to any information acquired by a sensor (e.g., position, time, temperature, speed, etc.” [44]	Extrinsic Environmental Information
“For any interoperable system the operator has control of the medium and equipment; the environment represents those items which are outside the operator's direct control.” [39]	
“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 dynamically modify their information processing approaches and, possibly, their representations.” [38]	Interaction Time
“For any interoperable system the operator has control of the medium and equipment; the environment represents those items which are outside the operator's direct control.” [39]	Intrinsic Environmental Information
“Many pervasive computing devices have limited processing and memory footprints, and wireless communication protocols offer differing reliability and quality of service. ” [50]	
“Semantic interoperability problem lies in the different understanding. The same problem can bring quite divergent views because of users' knowledge context.” [44]	User Knowledge

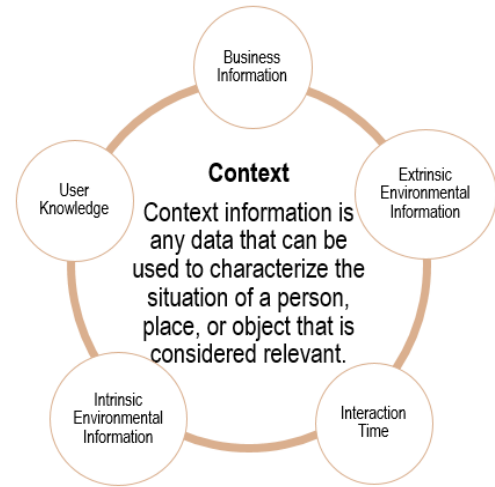


Fig. 4. Detailed Context Category.

5.4 Definition Category

The work of Ford *et al.* [69] identified 34 distinct definitions for interoperability. It shows that there are still differences in understanding and a lack of consensus on the concept, despite the extension of the research associated with the topic. This study also observed a lack of consensus on interoperability definition. We could observe in the open coding stage 96 different excerpts with ten open codes associated with interoperability definitions.

To better observe the identified definitions, these codes were grouped into five subcategories (axial coding) representing fundamental concepts shaping the definition of interoperability. Appendix B presents a detailed description of this category. The five subcategories are the **ability to exchange**, **system's property**, **integration**, **cooperation**, and **system's relation** (Fig. 5), described here.

The subcategory ability to exchange has six associated codes: exchange meaningful information despite the used instruments (g:2; d:2); exchange resources or information (g:14; d:8); exchanging and using resources or information (g:13; d:7); finding and sharing information from heterogeneous data sources (g:2; d:2); information exchange over a network (g:9; d:4); and provide, accept and use resources and information (g:2; d:2).

For the **system's property** subcategory, there are three codes associated: **functionality provided by middleware services** (g:2; d:2), **systems property** (g:3; d:3) and **system**

quality requirement (g:5; d:4). For the **integration** subcategory, there is one code associated: **surrogate for integration** (g:7; d:6). For the **cooperation** subcategory, there are two codes associated: **ability to cooperate** (g:18; d:6) and **to work together despite differences** (g:10; d:4). For the **systems relation** subcategory, there is one code associated: **relation between systems** (g:9; d:3).

Finally, the applied coding process 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**. We recognize that the term "things" is comprehensive and overloaded meaning (like the IoT perspective); but it was chosen to represent the broad range of interoperable items observed in the articles, such as: systems [55], [64], [66], services [48], [54], [57], objects [48], [61], equipment [48], [59], products [48], and software and applications [55], [56]. This definition also relates to other categories: a) **To interact**, we understand as related to the interoperability levels a system can engage the interaction; b) **For a specific purpose**, it is related to the Purpose category, presented next; c) **Once their differences have been overcome** represents the solutions and decisions that must be made to achieve interoperability, described in the Attributes category. Therefore, the definition is not only the representation of its subcategories but also includes other concepts that contribute to the understanding and definition of interoperability in software systems.

Excerpts	Open code	Subcategory
“Interoperability pertains to the capability of organizations to effectively communicate and transfer meaningful data (information) despite the use of a variety of information systems over significantly different types of infrastructure, possibly across various geographic regions and cultures.” [39]	Exchange meaningful information despite the used instruments	Ability to exchange
“Information gets exchanged between collaborating parties, and it is used in a meaningful way despite differences in language, interface or operation environments.” [56]		
“The interoperability of data is to find and share information from heterogeneous bases, and which can reside on different machines with different operating systems and databases management systems.” [47]	Finding and sharing information from heterogeneous data sources	
“The interoperability of data deals with finding and sharing information from heterogeneous data sources, and which can reside on different machines under different operating systems and database management systems.” [45]		
“The ability to operate in synergy in the execution of assigned tasks” [55]	Ability to cooperate	Cooperation
“The ability of two or more entities to communicate and operate together in a meaningful way.” [56]		
“Ability of two or more software components to cooperate despite differences in language, interface, and execution platform.” [44]	Working together despite differences	
“It is concerned with identifying, composing and making various applications function together (designed and implemented independently).” [45]		

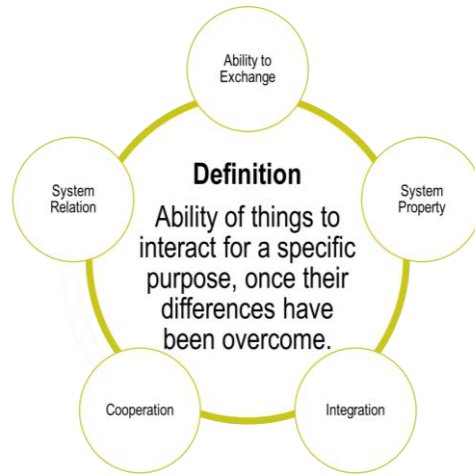


Fig. 5. Detailed Definition Category.

5.5 Selective coding: towards an Interoperability Theoretical Framework

During the open coding, we analyzed data extracted from the literature review identifying what was relevant. This exercise led to codes that were later organized into categories and subcategories.

Then, the axial coding was the step to relate the codes to their categories, including to describe how the concepts are grounded in data, with interpretation and abstraction. This process was briefly described in the previous sections. Finally, in selective coding, all the concepts associated with interoperability are integrated. These two central activities support the identification or proposal of a core category. The core category is essential in the phenomena under study and for all identified categories. We organized interoperability (Interoperability Definition Category) as the core category and observed how the other categories relate to it and with each other (Table 5).

The relations were identified by analyzing the excerpts occurring together and the implicit meaning of codes, also considering their groundedness and density [11]. With these measures, we can observe that some subcategories are more frequent (higher number) than others leading to stronger concepts due to more data to ground them.

For instance, it is observable that the three top concepts related to interoperability are attributes, issues, and

evaluation methods. Once the concepts are related, they were organized into an Interoperability Theoretical Framework - ITF Fig. 6 [13]. It allowed the representation of concepts together with their definitions and relations.

The relations in the ITF are described as *mn* relationship (*gn, dn*), where *mn* means the relationship number, the **relationship** is the connection role, *gn* is the relationship groundedness, and *dn* is the relationship density, as previously defined. The data grounds the name of each relationship.

As emerged from data analysis, the concepts were 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 concern 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).

A dotted line indicates the behavioral concepts in Fig. 6. This division is a consequence of the analysis from GT and the discussions between the authors. The structural part comprehends Context (Section 5.2.1), Attribute (Appendix A8), Level (Appendix A1), Perspective (Appendix A3), and Purpose (Appendix A2). The behavioral part comprehends Evaluation Methods (Appendix A7), Challenges (Appendix A6), Issues (Appendix A5) and Benefits (Appendix A4).

Table 5. Categories Overview.

Category	Definition
Definition	The applied coding process allowed us to propose the following interoperability definition, grounded in data: Ability of things to interact for a specific purpose, once their differences have been overcome. The definition is not only the representation of its subcategories but also includes other concepts that contribute to the understanding and definition of interoperability, relating to other categories, being the core of the Selective Coding.
Purpose	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 own system's purpose motivates the goal or objective of the interoperability to happen.
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.
Perspectives	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 category refers to the interpretation and application of the interoperability concept in different fields, domains, and areas.
Benefits	The analyzed studies reported some advantages observed in interoperable systems.
Issues	When dealing with interoperability some problems and concerns can arise. In the performed analysis under our dataset we identified different segments that were associated with interoperability issues.
Challenges	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.
Evaluation Methods	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.
Attributes	Typical systems' attributes related to interoperability and the decisions to be taken and mechanisms recurrently used to address these attributes.
Context	Context information is any data that can be used to characterize the status of a person, place or object that is considered relevant.

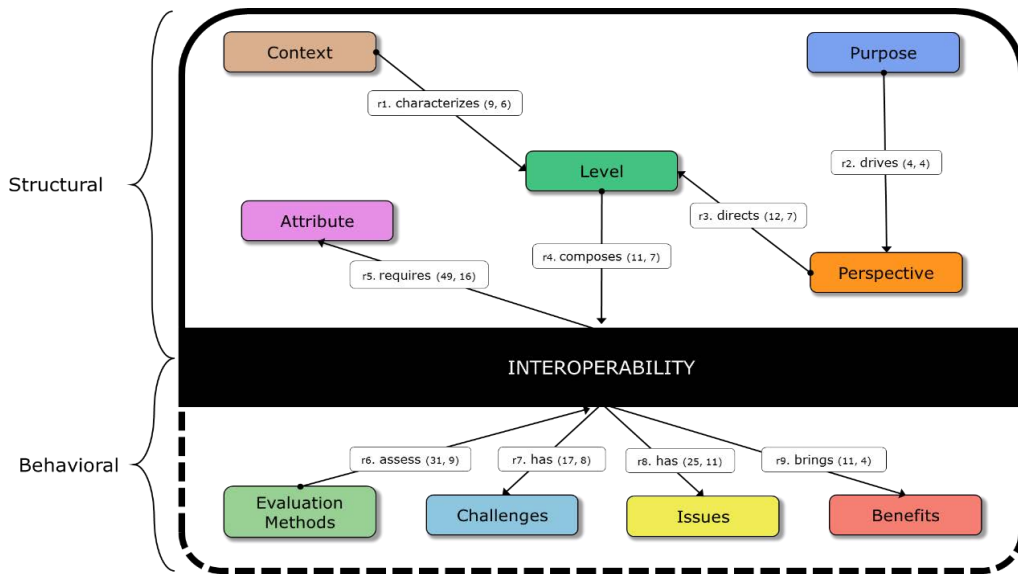


Fig. 6. Interoperability Characterization in a Theoretical Framework.

Once the interoperability is realized (from the structural viewpoint), it can manifest itself and be measured, improved, and observed (from the behavioral viewpoint). Each relationship is discussed below:

r1. Context characterizes Level: Context information describes distinct features in Levels. We understand that interoperability can be observed at different levels (**technical, syntactic, semantic, and organizational**), depending on the level of support provided. From the analyzed data, context information (**business information, extrinsic environment information, interaction time, intrinsic environment information, and user knowledge**) does not prevent the

interaction from happening. The structure required for an interoperable system will continue to exist regardless of the context information. However, what is changed is the adequacy (**the quality of being good enough for a particular purpose**) of a given interaction changes.

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. Besides, it is a need for interoperability

that drives each perspective (**organizational, systems, and services**).

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 considered, according to the purpose and reflected in the levels to achieve interoperability accurately.

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 materialize, which makes the interoperability adequate for that particular case.

r5. Interoperability requires Attributes: The perspective drives the levels composing interoperability in a given interaction. Each level, in its turn, adjusts the requirements to enable interoperable systems. The required interoperability attributes depend on the levels composing it. Codes in this concept are **system attributes** and **interoperability mechanisms**. They allow translating the needs of each level, presented in an abstract form, in possible solutions and more practical decisions.

r6. Evaluation Methods assess Interoperability: The concept, and therefore, this relationship, arises from one of the *qSLR* research questions: **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.

r7. Interoperability has Challenges: Like any other research topic, interoperability has its challenges as well. Of the possible challenges our study revealed, such as interoperability of new systems with legacy ones, or future work and open issues, are highlighted in the analyzed studies. We separate them between research and practice.

r8. Interoperability has Issues: Management and developing decisions lead to consequences. Interoperability is no different. We decided to code the concepts as issues because it could encompass problems, concerns, and difficulties when addressing software systems 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.

r9. Interoperability brings Benefits: From the analyzed data, we recurrently came across positive consequences,

presented in the concept of advantages, which can happen because 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.

For the sake of completeness, further discussions and examples are available in [82].

5.6 Exemplifying the Interoperability Theoretical Framework

To keep the representation grounded on data, we choose to demonstrate the use of the ITF to represent the situations captured in examples retrieved from *qSLR* papers.

Example 1 (Fig. 7):

*“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 the interoperation between electronic services with well-defined, self-descriptive interfaces.”* [66]

What the authors call “service level,” we interpreted as a **service perspective**, because it requires particular decisions and has specific concerns. The service perspective drives the **technical** and **semantic level** composing the interoperability requirements needed in the example. Because of this relationship, 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 decisions.

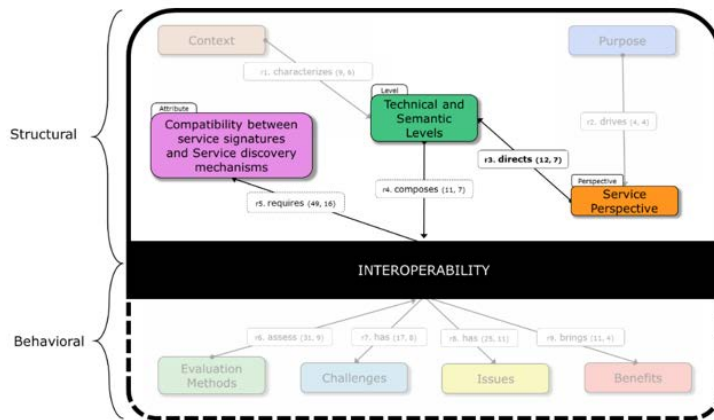
Example 2 (Fig. 7):

*“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.”* [47]

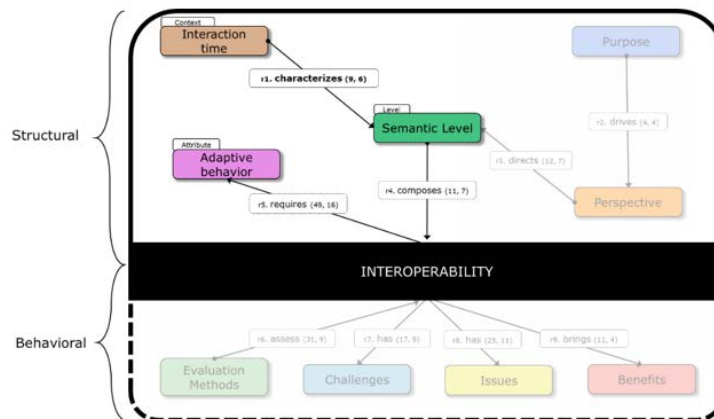
To our interpretation, the **semantic level** enables mutual understanding between interoperable systems. Over time, the meaning and form of information can change, which can affect the interaction itself. If mutual understanding is required, but the information cannot be understood, the interaction is meaningless.

Because of this relationship, to address the interoperability adequately, some **attributes** may be necessary. In this example, to address this concern, adaptive behavior is necessary to the system can respond to any change during **interaction time**.

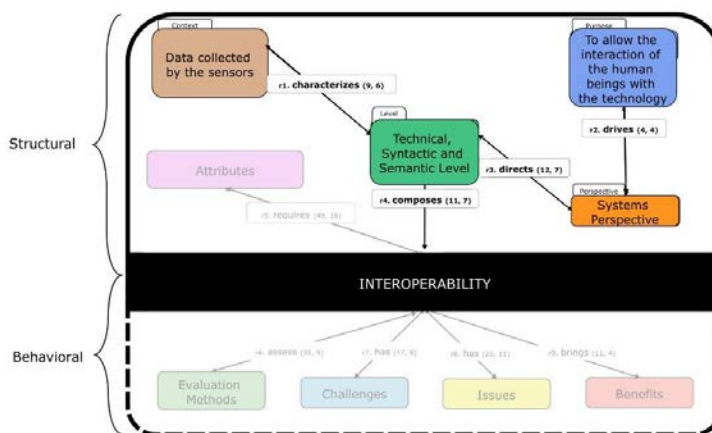
to organize it. We observed that the ITF is generic enough to



Example 1



Example 2



Example 3

Fig. 7. Examples of Interoperability Situations using the ITF.

Following the previous example, we decided to revisit the not analyzed articles (below the first quartile - Fig. 2A) to strength confidence in using the ITF to deal with these cases. Although the eight articles [70]–[77] did not fully meet the quality assessment, they are aligned with this work. The idea of revisiting these articles once the ITF has been proposed is to observe how it supports data from outside the dataset used

represent interoperability concepts in other particular situations described in the paper. The process consisted of taking back the articles and using the extraction form as the basis for the matching between the concepts in the framework and the ones captured in the articles.

Example 3 (Fig. 7) To exemplify, one of the articles states:

“The perception of the outside world requires that devices are embedded in the environment with the purpose to allow the interaction of the human beings with the technology. Data collected by the sensors have to be transmitted by communication networks and preprocessed by complex components, which collate and harmonize data from different devices.” [72]

In this case, interoperability is required to enable the interaction between technologies, especially sensor devices. These sensors generate data that need to be collected, transmitted, and preprocessed, reflecting a need for interoperability support in **technical**, **syntactic**, and **semantic levels**. Moreover, this support should consider changes in the "outside world" **context**, translated from the data collected by the sensors. With this setting, to make information useful, the attributes and mechanisms must consider obtaining accurate information about the environment and its users and enable the system to act, through various types of actuators, **to achieve its objective**.

6 ANSWERING THE RESEARCH QUESTIONS

The answers presented in this section intends to contribute to evolving the established knowledge, taking into account the findings observed during the qSLR execution, together with the qualitative data analysis supported by GT practices.

6.1 (RQ1) How is interoperability discussed in CASS?

A recurrent perception in the qSLR findings and during data analysis is that interoperability can be interchanged with the following concepts: connectivity, integration, or exchange. The decision on first observing the studies' definitions was to understand the selected works and solutions mindset and perspectives. If a study considers interoperability as integration, the work effort and results will vary from another one in which is set as communication.

Furthermore, the definitions can provide the foundations for any further development. Therefore, understanding this concept using general software systems 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 have proposed the following definition of interoperability of software system:

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

This definition is generic enough to cover the definitions we found while also reporting the interoperability concept complexity. The use of term *things* was decided because we are dealing with systems, including their hardware and software components as well as considering smart things. The interaction stands for any relationship where a system, component, software service, or thing can engage with another. That way it covers the exchange of any kind.

When we add the text “for a particular purpose, the definition intends to embrace the idea of cooperation and collaboration, frequently related to interoperability while uniting with one of the concepts uncovered during data analysis. This difference is a step towards comprehending interoperability to affirm in its definition. Once the differences are known, it is easier to overcome them to achieve interoperability accurately.

One of the most used interoperability definitions is given by the American Department of Defense [69]:

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 efficiently together, and the interaction is to provide, accept, and use services. Another definition previously presented is one by IEEE [78]:

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 regards the exchange and use of 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. Some strategies are required as we have observed in the mechanisms and attributes presented to overcome these differences in a way that the interaction can happen and achieve interoperability.

After observing the general interoperability understanding and justifying our definition, one of the challenges in investigating CASS is precisely the term: context. An effort was made during the coding process to differentiate this term in each study. Rezai *et al.* [48] and some referenced works [79], [80] consider interoperability to the use of different representations, purposes, and contexts.

This vision refers to the various contexts of development and usage, not regarding contextual variation, which can affect the system behavior. In other works [55], [56] context-awareness 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 they do not consider context as defined in our research. Gjoreski *et al.* [58] present a CASS with activity recognition where the system behavior acts according to the context. Interoperability is evaluated when setting up the scenario and observed in the initial phase on the recognition testing.

Other papers [55], [61] 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 CASS, the solution lies in the use of ontologies, especially for semantic issues [81]–[83]. Chen [57], based on ISO 14258 [84], 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 can be a solution for the current issues of heterogeneity and dynamicity part of the CASS and ubiquitous computing scenario. It is often considered the Federated approach [42], [55] where there is no standard format established, and interoperability should be achieved on the fly. It is still a current challenge for different reasons. For Spalazese [85], the possible heterogeneity of interconnected actors in the systems challenges their connection and communication. It can lead to mismatches among the used protocols. Another challenge in this regard is related to the methods and standards for specifying semantics to be suited to automated interpretation [86].

Nonetheless, from the data set we analyzed, this still is a research challenge in the field with a few early efforts towards suitable solutions [86], [87]. Sullivan and Lewis [60] also state that the key to interoperability in CASS cannot rely on prior agreements, but somehow to be able to adapt at runtime to be interoperable. In all of these works [64], [81], [82], [83] 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 (cell phones, tablets); poor application portability and increasing software platform dependency. As the solutions arise, also comes a growing number of organizations to be involved in achieving the seamless interoperability implied by the ubiquitous computing vision.

Based on our context definition [1], the selected articles provide five different context information:

- **Business Information** – it concerns with the context information that guides the interoperability levels in the business (organizational) perspective bringing significance to the interactions.
- **Extrinsic Environment Information** - extrinsic means: 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 of a whole. It refers to variables like position, time, climate, 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** – intrinsic means: 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 entities are interacting can influence interoperability.
- **User knowledge** - the user’s background can affect interoperability as knowledge compromises the user’s comprehension and perspectives regarding a software system (related to semantic).

The information previously described represents what the selected articles provide 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 [34]. 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 properties that should be respected and adequately addressed. 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 the levels composing the interoperability between two systems will require attributes and mechanisms to deal with context information.

Given that, even whether the interaction between the systems is enough so that the variation of context can influence the system behavior, the infrastructure required for interoperability remains as it is part of the system itself. For the observed context information, the basic system structure to be interoperable would continue to exist regardless of it. What can be changed is the adequacy of a given interaction.

Recalling the central question, how interoperability is addressed in context-aware software systems? From our findings, the answer to this question is still in an embryonic stage. To explain it in one sentence, it is clear that after many years, Weiser’s [27] 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 in its infancy, 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 CASS is often addressed in the initial development phase, when composing software systems, it is based on previous agreements and shared knowledge while the context is clear information with limited influence in the software systems behavior when related to interoperability.

6.2 (RQ2) Which are the interoperability elements in CASS?

Initially, this question aimed at providing a list of interoperability elements regarding context-aware software systems, from what we could distinguish them from other

system properties. From this analysis, we classified the elements into two groups: **systems attributes** (Table 6) and **interoperability mechanisms** (Table 7) relating to interoperability. The existence of **Systems attributes** makes it easier for the system to interoperate with others. We can claim that interoperability cannot be (at high abstraction level) directly observed but perceived through its related attributes and results.

Table 6: Systems Attributes.

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 is: when identifying the bandwidth reduction to the point of harming the audio and video transmission, the streaming service continues, but video quality is reduced.
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 is: GPS satellites and the mobile phone should be available at all times to collect the correct and up-to-date position data.
Compatibility	Compatibility of a system deals with how easy 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 is: extrinsic environment information, like temperature, should be in a compatible scale between the sensor and the 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 non-technological aspects. One example of how the adaptive behavior is relevant is: including the context of access control and support resolutions based on the current situation, dynamically adjusting user role and permissions according to organization rules.
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 is hardware components from sensors, mobile client, and servers should be supported regardless the variety of settings and programming languages.
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 the permissions. Authenticate and authorize interactions in automatic connection based on previous interactions or re-establish lost connection. One example of how the dynamic connection 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.
Standardization	It refers to bringing conformity between the systems that have to interoperate. Some decisions cannot necessarily be a 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 reach spontaneous interoperability, there is no sharing of a priori knowledge until now whether standardized solutions are necessary to achieve interoperability. One example of how standardization is relevant is the use of practices and standards for geographic information, like a standardized system of coordinates.

The **interoperability mechanisms** (Table 7) are decisions to be taken and solutions recurrently used throughout the software systems to achieve interoperability. Before the coding process, we recovered 87 excerpts coded as interoperability mechanisms from the selected studies. Due to the short description of some interoperability mechanisms and high recurrence of others, they were also coded.

Table 7: Interoperability Mechanisms.

Interoperability Mechanisms	Description
Data definition	It is important to make decisions related to the data type, data structure, data representation, data format, language to interact with the data, – and this also is related to the standards to be used and the consistency between the relating entities.
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.
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.
Human actions regarding interoperability	It addresses unexpected business issues requiring human reasoning.
Share common information and concepts	Information specifications provide the basis 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.
Use "bridges" to enable interaction	A posteriori solution that acts as an intermediate between inter-related systems enabling interoperability.
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.
Use compensation to enable interaction	A posteriori solution is to use some compensation as an alternative to something that went wrong.
Use negotiation to enable interaction	Negotiation mechanisms are used when different actors influencing various systems have divergent 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.
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.
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.
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.
Use suitable architecture	Architectural decisions have consequences affecting the whole system infrastructure.
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, the use of compatibles models is necessary to achieve interoperability.
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.

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, since they are grounded in data and reflect what was recovered. The technology usually evolves, and the most appropriate solutions vary over time and from system to system. 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 for software systems development, or it can be included in available software systems. 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.

6.3 (RQ3) How to evaluate interoperability in CASS?

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 third research question, we report the findings related to the evaluation of interoperability in such systems.

The following six methods were detailed described in the *qSLR* articles:

- **Enterprise Interoperability Maturity Model** - each area of concern would be defined by a set of objectives and goals relevant to interoperability. The maturity levels would be defined for each area of concern [48], [55] depending on the presence or lack of indicators.
- **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), 0 (machine translation required), 1 (no human or machine actions necessary to be interoperable) [48], [61].
- **Interoperability Assessment Methodology** - it introduces nine components, which are: requirements, node connectivity, data elements, protocols, information flow, information utilization, interpretation, latency, and standards. Each component include either a "yes/no" response or a mathematical equation [48].
- **Levels of Information Systems Interoperability (LISI)** - with five interoperability levels (Isolated, Connected, Functional, Domain, and Enterprise), in which each interoperability level exists in a specific environment. The model attributes contain Procedures, Applications, Infrastructure, and Data [48], [55], [61], [65].
- **Organizational Interoperability Maturity Model** - this model extends the LISI model into more abstract layers of command and control support. Five levels of organizational maturity are defined in this model (independent, ad-hoc, collaborative, combined, and unified) to describe the ability to interoperate [48].
- **Quantification of Interoperability Methodology** - it states that "interoperability of systems, units, or forces can be factored into a set of components that can quantify interoperability" and it identifies seven necessary components as languages, standards, environment, procedures, requirements, human factors, and media [48], [65].

Ten other methods were only introduced in the papers without an in-depth discussion. They were included in the "Other evaluation methods" subcategory.

In some cases [55], [56], interoperability deals with systems working together, considering (and removing) the interaction and interoperability barriers. Tolk has widely discussed interoperability [88], [89] and proposes the Levels of Conceptual Interoperability Model (LCIM) [90] proposed to deal with conceptual interoperability issues. The model, in the recent version [91], counts with seven interoperability levels that go from "no interoperability" to "conceptual interoperability" and provides methods and requirements that must be considered to achieve each level. Another work is from Chen, the Enterprise Interoperability Framework [55], [92] developed within the frame of the INTEROP NoE initiative. The framework is composed of Concerns (Data, Service, Process, Business), Barriers (Conceptual, Technological, Organizational) and Approach (Federated, Unified, Integrated). In both, the idea is to identify the barriers in early phases before the implementation, but they do not consider the contextual information, as defined in our research. In another work [93], Rezaei *et al.* present an interoperability maturity model for ultra large scale systems, considering some aspects from LISI, Enterprise Interoperability Maturity Model, and others.

Despite finding different assessment methods, most of them do not address the context. However, two initiatives found in two other articles [54],[58] attempt to tackle this issue.

Fang *et al.* [54] present a proposal for dealing with interoperability 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 (standard service's function understanding) by fuzzy interoperability quantization. Additionally, the quality and context levels are considered. The limitations of this work are the focus on service composition and a lack of information on its empirical evaluation. Despite considering the context, this level is not assessed because it is a work in progress. Gjoreski *et al.* [58] present a laboratory experiment, which evaluated eight software systems in recognition of seven activities representing an older adult's lifestyle and provides the context for the 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.

Considering our results, we did not find an evaluation method designed for CASS or considering context information while assessing interoperability. Therefore, we outline some alternatives for interoperability evaluation in

CASS using the proposed ITF (Section 5.3, Fig. 6) as a guide for assessment. The **first step** is to define the *purpose* of the system and the *context* to determine how it will affect the system. From this initial characterization, following the ITF structure, in the **second step**, we can delineate the *perspective* and required a *level* of interoperability for such a system. This definition leads to an overview of the structural interoperability organization from which we can outline the required *attributes*. The **third step** consists in check for the gap between the interoperability ideally desirable and the attributes and mechanism actually in use in the target system. With this perception, we can observe the adequacy of the system for what is desired and directions to improve. This is a brief overview of long-term research to investigate better and produce a proper evaluation method considering context, as proposed (Fig. 8).

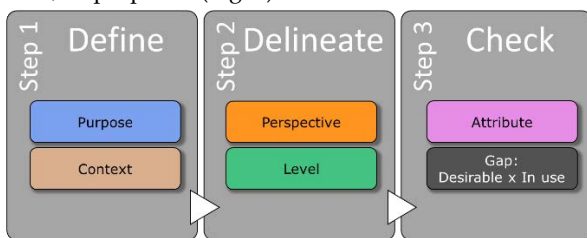


Fig. 8. Evaluation using ITF.

Another evaluation way can be extending some of the existing evaluation methods to consider the context information and the infrastructure required to acquire, store, control, and use the information. This alternative compares existing methods regarding the seven system 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 the ITF can provide proper support for the evaluation issue, which configures an opportunity for future work. However, this research question requires further investigation. For instance, a more in-depth analysis of each model represents a research opportunity to enrich the set of interoperability attributes regarding RQ2.

7 EXEMPLIFYING THE INTEROPERABILITY THEORETICAL FRAMEWORK

One of this work's contributions is the organization of a body of knowledge regarding CASS interoperability, represented by the ITF (Section 5.3, Fig. 6). The framework's structured information represents a set of concerns to be considered when *things need to interact* within and with software systems. As it was based on a literature review, the framework proposal is grounded in available knowledge in technical papers. Also, the structured organization and aggregation of the different elements presented in the ITF demonstrate a contribution of the proposition. We can consider the concepts as the issues to be considered when discussing

interoperability in software systems. Also, one of the differentials of the ITF against previous proposals is the inclusion of context, which is crucial for CASS. Thus, we can map the interactions between the actors and the execution environment as well, since they directly influence the system.

For researchers, to observe the established levels of interoperability (technical, syntactic, semantic, and organizational) at a smaller grain may lead to opportunities for further investigation. For instance, understanding the level of application of current software technologies, what are the open issues in each of these fronts, and looking for standards that can guide the building of solutions in each interoperability level represent possible investigation opportunities. Such knowledge structure can also be used to present the level of technological concerns regarding the multi-disciplinarity of context-aware and contemporary software systems. The complexity and demand increase every day, so there is no more room for independent software solutions. Instead, there is a real need for interaction. The proposed framework comes as an instrument to support a more comprehensive view of interoperability. It allows the instantiation of various scenarios, contemplating different perspectives concerns, including those related to the actor-computer interaction.

For practitioners, the characterization of their projects, using the knowledge structure, can draw attention to interoperability issues not observed before. Organizing a project under the issues raised by the ITF can support the observation of required levels of interoperability and the anticipation of integration situations beyond the solution itself, such as team, effort, and risks. For instance, if the project requires high semantic interoperability, the team in question should observe the mechanisms to attend it, and its skills must agree, being able to use specific solutions such as ontologies. This team arrangement can vary whether the project is driven by complex business rules, indicating a need for achieving organizational interoperability. It is possible to assume that the more levels of interoperability are needed, the more effort will be required to provide a solution. Besides, the more actors involved in the interaction, the higher the risks incurred. In such cases, the ITF can be used to organize the project decision making by allowing visibility to issues that might not have been previously thought or discussed. Therefore, it is possible to operationalize the ITF, as it can be used to support the project decision-making. For instance, it can be interesting to investigate, organize, and make available further options of system attributes and interoperability mechanisms as a meta-model or tool for different software system types usually built at the software organization.

Even a relatively simple traffic control system example as described in the introduction can present interoperability issues. The data is generated by different devices from roads and users, such as RFID, sensors, actuators, Global Position Systems (GPS), and laser scanners. Then it is collected from

real-time traffic data; the system can recognize the current traffic, its flow conditions and direct the routes for emergency vehicles. For that, a semantic analysis is required to deal with the input from different data sources. The purpose of supporting traffic decisions drives (R2) the organizational perspective since it relies on the city public control. This perspective directs (R3) the interoperability level required for the system, together with the context. The context should consider business information like hospital locations and ambulances routes, extrinsic information such as the data retrieved from the devices and the interaction time since the connection should be available at any time. Context characterizes (R1) interoperability to work in a semantic level requiring (R5) attributes like compatibility, availability, information assurance, and their respective supporting mechanisms.

One of the possible ways of using the ITF can be seen in Fig. 9. As a result of RQ3, 14 methods are cited, and six methods discussed in more detail. It is possible to compare the coverage of each method with the ITF as a possibility to extend it. For that is a necessary further investigation in each method and observe their results and impacts and the ITF adequacy regarding them (Part A: illustrates the comparison of the methods marked in red and green to represent the coverage regarding ITF). From this match and framework evolution, we can implement the ITF to serve as a guide to check if each of the framework elements was considered when implementing interoperability (Part B: A concrete implementation of ITF that can be configured by a system and evaluate its suitability regarding the concepts, attributes, and mechanisms). This idea is still at an early stage, but we characterize it as future work.

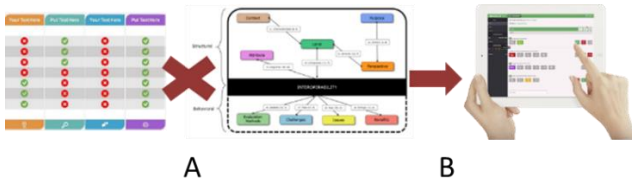


Fig. 9. Operationalization Idea.

It is essential to highlight that not all concepts must be instantiated for all software systems. The ITF should be used according to what is relevant to a particular system. For CASS, it is necessary to define the context information, which will help to determine which behaviors the software system shall support. Another example regards the system purpose, it can vary from a system to another, and the ITF should be used according to it. Also, it is not every system interaction that requires all interoperability levels (technical, syntactic, semantic, and organizational); the ITF can guide such identification and serves as a complementary requirements elicitation instrument.

We argue that different types of actors, interactions, and relevant contexts should be considered in the software systems' quality evaluation [2]. In this way, the ITF can be an

option to deal with conflicts, improve the identification of relevant information, and used together with verification and validation techniques.

8 BACK TO THE FUTURE

Although interoperability is a not-so-new issue, it continues to be a core concern in new systems. From the results achieved in this research, investigating interoperability in CASS, we start to look to new CASS systems such as the Internet of Things (IoT), smart cities, smart homes, ubiquitous computing, and many other contemporary software systems. This new generation of software systems requires the integration of different engineering domains (e.g., software engineering, human-machine interaction) and the higher need for the software to be embedded in the product [94], [95]. Communication and interoperability are essential for the system materialization [96], [97] in those cases.

Focusing on IoT systems, a paradigm that explicitly considers the interaction issues, we have investigated the different concerns on these systems, including interoperability [98]. We consider inputs from a literature review, practitioners and a report from the Brazilian government, to broaden the range of the research contributing to a more comprehensive representation. Considering these different sources, interoperability got unanimous importance and is a core challenge for IoT. Considering the results of [98] and our additional research in this direction, we point out some interoperability challenges:

- As a result of the environment, interaction, and collaboration of different devices, a new behavior can be generated. It is necessary to provide interoperable solutions that can deal with **dynamic properties** and **emergent behavior** [99], [100]. This demand is connected to requiring full interoperability of interconnected devices by providing a degree of **smartness** to enable their **adaptation** and **autonomous behavior** [101].
- IoT allows different objects to be seen as a resource for increasingly complex systems with sensors, monitoring, and management through software. Internet of Vehicles, for example, will require **seamless interoperability** among vehicular sensors, computing platforms, and consumer devices [102].
- The wide diversity of data will require a new breed of intelligent algorithms with the ability to adapt and self-learn as well as envisage and analyze events at multiple levels of abstraction to achieve interoperability through **self-learning** and incremental learning **intelligent algorithms** [103].
- Do-it-yourself approaches [104], [105] are becoming more popular, as a way to empower the user in the IoT scenario. This will require a **"Plug n' Play"** interoperability where smart objects can be deployed in any environment with an interoperable backbone [106].

- The **Practitioners** pointed out that aside the primary concern with the interaction of so many different devices, an important issue is how to address the programming of multi-devices, considering interoperability for development as well.
- The report from the **Brazilian government** raises the fact that if left free to the market, the standards developed by technology giants may result in monopolies, leading to the exclusion (or cost-intensive inclusion) of technologies in the global IoT ecosystem.

This highlights challenges and opportunities for interoperability in the nearest future. In this scenario, if IoT is the paradigm that enables computing capabilities in things around us, interoperability is the attribute that will enable the interaction among heterogeneous devices, with varied requirements of different applications. The results of this research and the proposed ITF can be evolved to adapt to solutions more complex than CASS. Even so, this work remains relevant as discussions about interoperability are as crucial as ever.

9 LIMITATIONS AND THREATS TO VALIDITY

As in any empirical study, various menaces and risks can compromise their results. In this particular, the methodology involves different investigation strategies and procedures, such as Grounded Theory. In GT, the coding process is challenging and is essential 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 to the research goal. Conceptualizing, observing the similarities and differences among the excerpts, keeping the consistency among the relating concepts and working in data interpretation are significant activities to any qualitative analysis.

Nevertheless, such issues are sources of risk to the validity of the study since different researchers may have different data understanding, leading to other concepts and the interpretation of their relationships. To mitigate this risk, we worked focusing on assuring three crucial aspects, as suggested by GT authors [13]:

- **Research process adequacy** - the original sample was selected based on the data extracted from the *q*SLR. The categories are presented in the axial coding subsection and emerged considering its groundedness corresponding to the data and suitability for the area. The relationships are given considering density and discussed in the results subsection. These conceptual relations lead to the core category, Interoperability, the central theme of this research. Details of the process are presented in Appendix A.
- **The findings groundedness** - each concept can be linked to the codes, and then to the excerpts, where it is grounded, which can be easily achieved with the support of the QDA Miner tool. When a concept relates to another, the relationship is presented by examples from the data

analyzed. These examples also illustrate the variations and conditions of which the concepts were examined and developed, as in the interoperability definition.

- **The quality of data** - the papers were selected by following a well-defined protocol, based on the *q*SLR guidelines. Their contents were evaluated considering their capacity to contributing to the research questions. The strength of their contributions was not evaluated. However, the papers offer empirical discussions on interoperability, which reduces our capacity to measure the degree of influence each one may affect the final result. Therefore, based on the qualitative analysis, we choose to consider groundedness and density as surrogates to indicate our findings strengths.

Additionally, the combination of empirical strategies and procedures also contributes to other threats to validity [107], as follows:

- **Internal validity** - as a central feature of an SLR, there is a possibility to miss some studies. Moreover, the natural bias the researchers can have in selecting the papers can influence the results. Therefore, a well-defined protocol and process can lead to consistency studies selection and minimize the research bias when performed by different researchers, contributing to mitigating this risk. Besides, it has been revised by two external researchers. Another recurrent threat in SLR regards inconsistent terminology, and the keywords used to be restrictive. To tackle it, first, we searched for other literature reviews and observed the terms used to compose our search string. Also, the protocol and structures shared good practices and similarities with other *q*SLR undertaken in the CACTUS project [1][17]. Some initial trials presented several network-centric works inserted in the integration layer, such as middleware as a solution for interoperability. Considering the selection criteria and our concern about having more information regarding semantic, systemic, and business visions, such works were not included. We take the risk of losing primary sources at the level of technical interoperability for the sake of a general and broader interoperability discussion. Another threat is that despite several trials and string adjustments, the use of only one research database can lead to missing studies as well, despite the overlapping, the search engines can present [20]. Backward and forward snowballing were performed to mitigate it. Moreover, all phases of this review were peer-revised with doubts discussed among the researchers.
- **Conclusion validity** - it is vital for an SLR study to present findings traceable to data. The presented results were drawn from data analysis, systematically retrieved from the *q*SLR, ensuring a high degree of traceability between data and conclusions. However, the significant validity threat for any qualitative study is the interpretation bias. To reduce such 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. The meaning of a concept was clarified using a dictionary to avoid any misinterpretation, mitigating the interpretation bias.

- **External validity** – this threat is related to the possibility to generalize the results. Our study relies on the methods from the technical literature as guidance for data analysis, and the results represent synthesized concepts related to interoperability in CASS view, as found in the primary sources. The results applicability, at the moment, is limited to the data analyzed, so it is necessary to confirm its validity with additional data. The study results were considered concerning the research goal. Thus, the findings presented are valid, considering the current set.

One work limitation is the set of selected articles as the research basis. Due to the time lapsed from the search, analysis, and report we decided to re-apply the search string to keep track of what has been published so far, revisiting and reevaluating the findings against our selection criteria. From title selection, we kept 11 articles, and five remained after abstract selection. After full-paper reading, the article by Vassev and Hinchey [108] drew attention but was not included since it did not address the heterogeneity or interoperability of the systems directly. None of the articles contributed to the research questions or to confirm the ITF within the present work scope, therefore were not included in our set.

Forward snowballing: We used the 17 papers selected in our review to perform forward snowballing searching in the Scopus database for the publications that cited them from 2016 to the execution date, in May 2018. It resulted in 284 articles cited at least one of the works in our set. These articles were then evaluated against our selection criteria. From title selection, we kept 34 articles, and two remained after abstract selection [109], [110]. The work of Gambi *et al.* [109] focused on architectures and frameworks proposed for Smart Homes and Ambient Assisted Living, mainly three leading solutions are described: UniversAAL, Domoinstant, and AllJoyn. They conclude the paper by reporting that interoperability, at different levels, is still an open problem. In the work of Weichhart *et al.* [110], the discussion is on Enterprise Systems where they detail the need for an infrastructure to support the interoperability of the components to be connected to form an enterprise information system. As a solution, they adapted the existing Ontology of Enterprise Interoperability for Complex Adaptive Systems, implemented in a domain specific language.

Both works are different from our proposal, therefore were not included in our set. However, these activities have led us to realize that the topic remains relevant, as well as the proposal presented in the article. It also contributed to mitigating the time spent limitation in the research.

10 CONCLUSION

Considering that context-aware systems should be able to interact with different things, 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 when building such software systems. Therefore, this paper presented the results of a secondary study about interoperability regarding context-aware software systems. It detailed the performed activities aimed to contribute to the quality assessment of actor-computer interaction in ubiquitous systems and in a more focused way to observe the interoperability regarding CASS. Seventeen studies were selected from a quasi-systematic literature review and analyzed by using the Grounded Theory methodology. From this analysis, a body of knowledge about interoperability was organized and represented through an evidence-based Theoretical Framework composed of ten central concepts: interoperability definition, context, attributes, level, perspective, purpose, evaluation methods, issues, challenges, and benefits. Each one of these concepts was conceptually defined and grounded in data collected from seventeen primary sources. These results show a broader view of interoperability for context-aware software systems which evolves the classical sense of just “exchange information and use such information.” This study shows that to address interoperability for CASS means to take care of structural concepts (context, perspective, purpose, the level of provided support and system attributes), and behavioral concepts (evaluation method, challenges, issues, and benefits). These concepts highlight quality attributes that should be considered when planning, analyzing, and design interoperable CASS. Besides, they can provide relevant information to organize a checklist or reviewing script to detect interoperability defects or to guide the evolution of software systems regarding changes focused on interoperability. Also, it can support the planning of integration tests aiming at validating interoperability attributes.

As future work, we plan to build software technologies to evaluate interoperability based on measurements on top of the Interoperability Theoretical Framework proposed in this paper. To that end, we are initially focused on the Internet of Things applications [98]. Moreover, the ITF is being used in the development and testing of context-aware software systems developed in the context of our projects in collaboration with the industry.

11 ACKNOWLEDGMENT

We thank Victor Vidigal Ribeiro for his contributions to this research. This work is part of the CNPq CACATUS project – Universal 14/2013 (484380/2013-3). Prof. Travassos is a CNPq Researcher (305929/2014-3). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

12 REFERENCES

- [1] I. de S. Santos, R. M. de C. Andrade, L. S. Rocha, S. Matalonga, K. M. de Oliveira, and G. H. Travassos, "Test case design for context-aware applications: Are we there yet?," *Information and Software Technology*, vol. 88, pp. 1–16, Aug. 2017.
- [2] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [3] H. E. Byun and K. Cheverst, "Utilizing context history to provide dynamic adaptations," 2010.
- [4] R. O. Spínola and G. H. Travassos, "Towards a framework to characterize ubiquitous software projects," *Information and Software Technology*, vol. 54, no. 7, pp. 759–785, Jul. 2012.
- [5] A. I. Maarala, X. Su, and J. Riekk, "Semantic Reasoning for Context-Aware Internet of Things Applications," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 461–473, Apr. 2017.
- [6] E. Niemelä and J. Latvakoski, "Survey of requirements and solutions for ubiquitous software," *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia MUM 04*, no. October, pp. 71–78, 2004.
- [7] "ISO 25000:SQuaRE," *ISO/IEC/IEEE 25000:2011*, pp. 1–21, 2011.
- [8] iso, "ISO / IEC 19500 Information technology – Common Object Request Broker Architecture," vol. 2012, 2012.
- [9] C. Iso and others, "ISO/IEC 2382-1: 1993 Information Technology - Vocabulary - Part 1: Fundamental terms," 1993.
- [10] G. H. Travassos, P. S. M. dos Santos, P. G. Mian, A. C. D. Neto, and J. Biolchini, "An Environment to Support Large Scale Experimentation in Software Engineering," *13th IEEE International Conference on Engineering of Complex Computer Systems (iceccs 2008)*, pp. 193–202, 2008.
- [11] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, vol. 1, no. 4. 1967.
- [12] J. Carver, "The Use of Grounded Theory in Empirical Software Engineering," in *Empirical Software Engineering Issues. Critical Assessment and Future Directions*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 42–42.
- [13] A. Strauss and J. Corbin, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Newbury Park: Sage Publications, Inc, 1990.
- [14] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "Characterizing Interoperability in Context-Aware Software Systems," in *2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC)*, 2016, pp. 203–208.
- [15] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," in *Proceeding of the 28th international conference on Software engineering - ICSE '06*, 2006, vol. 2, p. 1051.
- [16] J. Biolchini, P. G. Mian, A. Candida, and C. Natali, *Software and Data Technologies*, vol. 10. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [17] S. Matalonga, F. Rodrigues, and G. H. Travassos, "Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review," *Journal of Systems and Software*, vol. 131, pp. 1–21, Sep. 2017.
- [18] P. Mian, T. Conte, A. Natali, J. Biolchini, and G. Travassos, "A Systematic Review Process for Software Engineering," p. 6.
- [19] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*, pp. 1–10, 2014.
- [20] A. Martín-Martín, E. Orduna-Malea, M. Thelwall, and E. Delgado López-Cózar, "Google Scholar, Web of Science, and Scopus: A systematic comparison of citations in 252 subject categories," *Journal of Informetrics*, vol. 12, no. 4, pp. 1160–1177, Nov. 2018.
- [21] D. S. Cruzes and T. Dyb, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology*, vol. 53, no. 5, pp. 440–455, 2011.
- [22] T. Conte, R. Cabral, and G. Travassos, "Aplicando Grounded Theory na Análise Qualitativa de um Estudo de Observação em Engenharia de Software—Um Relato de Experiência," *\ldots sobre a Engenharia de Software \ldots*, no. 2006, pp. 26–37, 2009.
- [23] S. Adolph, W. Hall, and P. Kruchten, "A Methodological Leg to Stand On: Lessons Learned Using Grounded Theory to Study Software Development," p. 13, 2008.
- [24] S. Mota, "Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software," Universidade Federal do Rio de Janeiro, 2013.
- [25] L. Rocha, F. Lima, M. Maia, W. Viana, M. Castro, and R. Andrade, "Ubiquitous software engineering: Achievements, challenges and beyond," *Proceedings - 25th Brazilian Symposium on Software Engineering, SBES 2011*, no. November, pp. 132–137, 2011.
- [26] F. Rodrigues, S. Matalonga, and G. H. Travassos, "Systematic literature review protocol: Investigating context aware software testing strategies," 2014. .
- [27] R. O. Spínola, F. C. R. Pinto, and G. H. Travassos, "Supporting requirements definition and quality assurance in ubiquitous software project," in *Communications in Computer and Information Science*, 2008, vol. 17 CCIS, pp. 587–603.
- [28] M. Weiser, "The Computer for the 21st Century," 1991.
- [29] G. D. Abowd, "Software engineering issues for ubiquitous computing," *Proceedings of the 1999*

- International Conference on Software Engineering*, pp. 75–84, 1999.
- [30] L. Tang, Z. Yu, X. Zhou, H. Wang, and C. Becker, “Supporting rapid design and evaluation of pervasive applications: Challenges and solutions,” in *Personal and Ubiquitous Computing*, 2011, vol. 15, pp. 253–269.
- [31] M. Bazire and P. Brézillon, “Understanding Context before Using It,” *Modeling and Using Context: 5th International and Interdisciplinary Conference*, no. November, pp. 29–40, 2005.
- [32] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, p. 263, 2007.
- [33] G. Chen and D. Kotz, “A Survey of Context-Aware Mobile Computing Research,” *Science*, vol. 3755, pp. 1–16, 2000.
- [34] S. Lee and J. Chang, “Survey and Trend Analysis of Context-Aware Systems,” *Information-An International Interdisciplinary Journal*, vol. 14, no. 2, pp. 527–548, 2010.
- [35] I.-I. STANDARD, “ISO/IEC / IEEE 24765 - Systems and software engineering – Vocabulary,” vol. 24765, 2017.
- [36] C. A. Da Costa, A. C. Yamin, and C. F. R. Geyer, “Toward a General Software Infrastructure for Ubiquitous Computing,” *Tese (UFRGS)*, pp. 64–73, 2008.
- [37] C. Kidd, R. Orr, and G. Abowd, “The aware home: A living laboratory for ubiquitous computing research,” *Cooperative buildings. ...*, pp. 191–198, 1999.
- [38] J. A. Kientz and B. Jones, “The Georgia Tech Aware Home,” pp. 3675–3680, 2008.
- [39] V. Issarny, A. Bennaceur, and Y. D. Bromberg, *Middleware-layer connector synthesis: Beyond state of the art in middleware interoperability*, vol. 6659 LNCS, 2011.
- [40] K. D. Moon, Y. H. Lee, Y. S. Son, and C. K. Kim, “Universal Home Network Middleware Guaranteeing Seamless Interoperability among the Heterogeneous Home Network Middleware,” *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 546–553, 2003.
- [41] G. Zacharewicz *et al.*, “Model-based approaches for interoperability of next generation enterprise information systems: state of the art and future challenges,” *Information Systems and e-Business Management*, vol. 15, no. 2, pp. 229–256, May 2017.
- [42] R. G. Uninova, C. A. Uninova, E. S. Uninova, and Y. D. Ub, “MDA / MDI Model Transformation : Application to MDSEA M8 issue,” pp. 1–53, 2012.
- [43] Z. Liu and G. Zacharewicz, “Model transformation from BPMN format to openerp workflow,” in *13th International Conference on Modeling and Applied Simulation, MAS 2014*, 2014, pp. 225–230.
- [44] O. Noran, “Achieving a sustainable interoperability of standards,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 327–337, 2012.
- [45] T. Ford, “Interoperability Measurement,” pp. 1–198, 2008.
- [46] H. Panetto, M. Scannapieco, and M. Zelm, *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*, vol. 3292. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
- [47] A. M. Madni and M. Sievers, “System of Systems Integration: Key Considerations and Challenges,” pp. 330–347, 2013.
- [48] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. Shams Aliee, “Interoperability evaluation models: A systematic review,” *Computers in Industry*, vol. 65, no. 1, pp. 1–23, 2014.
- [49] J. Hong, E. Suh, and S.-J. Kim, “Context-aware systems: A literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.
- [50] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context-Aware Computing for The Internet of Things: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [51] H. Panetto, M. Zdravkovic, R. Jardim-Goncalves, D. Romero, J. Cecil, and I. Mezgár, “New perspectives for the future interoperable enterprise systems,” *Computers in Industry*, vol. 79, pp. 47–63, Jun. 2016.
- [52] V. R. Basili, G. Caldeira, and H. D. Rombach, “Goal Question Metric Paradigm.” 1994.
- [53] M. Pai *et al.*, “Clinical Research Methods Systematic reviews and meta-analyses : An illustrated , step-by-step guide,” *the National Medical Journal of India*, vol. 17, no. 2, pp. 86–95, 2004.
- [54] J. Fang, S. Hu, and Y. Han, “A service interoperability assessment model for service composition,” in *Proceedings - 2004 IEEE International Conference on Services Computing, SCC 2004*, 2004, pp. 153–158.
- [55] D. Chen, G. Doumeingts, and F. Vernadat, “Architectures for enterprise integration and interoperability: Past, present, and future,” *Computers in Industry*, vol. 59, no. 7, pp. 647–659, 2008.
- [56] L. Kutvonen, “Enhancing the maturity of open service ecosystems and inter-enterprise collaborations,” in *Lecture Notes in Business Information Processing*, 2013, vol. 144 LNBIP, pp. 6–21.
- [57] D. Chen, “Enterprise Interoperability Framework,” 2006.
- [58] H. Gjoreski *et al.*, “Competitive Live Evaluations of Activity-Recognition Systems,” *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 70–77, Jan. 2015.
- [59] G. Athanasopoulos, A. Tsalgaidou, and M. Pantazoglou, “Interoperability Among Heterogeneous Services,” *IEEE International Conference on Services Computing*, pp. 174–181, 2008.

- [60] D. O. Sullivan and D. Lewis, "Semantically Driven Service Interoperability for Pervasive Computing," pp. 17–24, 2003.
- [61] J. Ullberg, R. Lagerström, and P. Johnson, "A Framework for Service Interoperability Analysis using Enterprise Architecture Models," *2008 IEEE International Conference on Services Computing*, pp. 99–107, Jul. 2008.
- [62] Y. Naudet, T. Latour, W. Guedria, and D. Chen, "Towards a systemic formalisation of interoperability," *Computers in Industry*, vol. 61, no. 2, pp. 176–185, Feb. 2010.
- [63] T. Ford and J. Colombi, "The interoperability score," *Proceedings of the Fifth conference on systems engineering research*, pp. 1–10, 2007.
- [64] C4ISR Architecture Working Group, "Levels of information systems interoperability (LISI)," *March*, vol. 30, no. March p. 1998, 1998.
- [65] E. J. Wyatt, K. Griendling, and D. N. Mavris, "Addressing Interoperability in Military Systems-of-Systems Architectures," 2012.
- [66] T. Ruokolainen and L. Kutvonen, "Interoperability in Service-Based Communities," vol. 68, pp. 317–328, 2006.
- [67] M. Memon, S. R. Wagner, C. F. Pedersen, F. H. A. Beevi, and F. O. Hansen, "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes," *Sensors (Basel, Switzerland)*, vol. 14, no. 3, pp. 4312–41, Jan. 2014.
- [68] A. Zutshi, A. Grilo, and R. Jardim-Goncalves, "The Business Interoperability Quotient Measurement Model," *Computers in Industry*, vol. 63, no. 5, pp. 389–404, Jun. 2012.
- [69] T. Ford, J. Colombi, S. Graham, and D. Jacques, "A Survey on Interoperability Measurement," *International Command and Control Research and Technology Symposium*, no. 937, 2007.
- [70] T. Mcbryan, M. R. Mcgee-lennon, and P. Gray, "An Integrated Approach to Supporting Interaction Evolution in Home Care Systems," pp. 1–8, 2008.
- [71] C. Rodríguez-Domínguez, K. Benghazi, M. Noguera, J. L. Garrido, M. L. Rodríguez, and T. Ruiz-López, "A communication model to integrate the Request-Response and the Publish-Subscribe paradigms into ubiquitous systems," *Sensors (Basel, Switzerland)*, vol. 12, no. 6, pp. 7648–68, Jan. 2012.
- [72] A. Queirós, A. Silva, J. Alvarelhão, N. P. Rocha, and A. Teixeira, "Usability, accessibility, and ambient-assisted living: a systematic literature review," *Universal Access in the Information Society*, vol. 14, no. 1, pp. 57–66, 2015.
- [73] M. Damas, H. Pomares, S. Gonzalez, A. Olivares, and I. Rojas, "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*, vol. 19, no. 1, pp. 54–60, Jan. 2013.
- [74] M. Reynolds and S. Member, "Telemonitoring Systems Interoperability Challenge: An Updated Review of the Applicability of ISO/IEEE 11073 Standards for Interoperability in Telemonitoring," pp. 6161–6165, 2007.
- [75] A. Flores, M. Polo, J. C. Augusto, and M. Varea, "Towards context-aware testing for semantic interoperability on PvC environments," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2004, vol. 2, pp. 1136–1141.
- [76] H. Jimenez and D. N. Mavris, "Characterization of Technology Integration Based on Technology Readiness Levels," *Journal of Aircraft*, vol. 51, no. 1, pp. 291–302, Jan. 2014.
- [77] C. E. Kuziemyky and J. H. Weber-jahnke, "An eBusiness-based Framework for eHealth Interoperability," vol. 1, no. 2, pp. 129–136, 2009.
- [78] IEEE, "IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990)," *IEEE Computer Society*, 1990.
- [79] F. Lampathaki, S. Mouzakitis, G. Gionis, Y. Charalabidis, and D. Askounis, "Business to business interoperability: A current review of XML data integration standards," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1045–1055, 2009.
- [80] S. Koussouris and F. Lampathaki, "Digging into the real-life enterprise interoperability areas definition and overview of the main research areas," *Proceedings of CENT*, pp. 19–22, 2011.
- [81] D. Beckett and B. McBride, "RDF/XML Syntax Specification (Revised)," *W3C Recommendation*, 2004. [Online]. Available: <http://www.w3.org/TR/REC-rdf-syntax/>.
- [82] D. McGuinness and F. Harmelen, "OWL WebOntology Language Overview," *W3C Recommendation*, 2004. [Online]. Available: <http://www.w3.org/TR/owl-features/>.
- [83] N. Ryan, "Smart environments for cultural heritage," *Computing*, 2005.
- [84] ISO, "ISO 14258 – Concepts and rules for enterprise models," 1999.
- [85] R. Spalazzese, P. Inverardi, and V. Issarny, "Towards a formalization of mediating connectors for on the fly interoperability," in *2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA 2009*, 2009, pp. 345–348.
- [86] G. Blair, N. Bencomo, and R. B. France, "Models@run.time," *Computer*, vol. 42, no. 10, pp. 22–27, 2009.
- [87] J. Juárez, J. A. Rodríguez-Mondéjar, and R. García-Castro, "An ontology-driven communication architecture for spontaneous interoperability in Home Automation systems," *19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2014*, 2014.

- [88] A. Tolk, S. Y. Diallo, and C. D. Turnitsa, "Applying the Levels of Conceptual Interoperability Model in Support of Integrability, Interoperability, and Composability for System-of-Systems Engineering," *Journal of Systematics, Cybernetics, and Informatics*, vol. 5, no. 5, pp. 65–74, 2007.
- [89] A. Tolk, "What Comes After the Semantic Web - PADS Implications for the Dynamic Web," in *20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, 2006, vol. 2006, no. February 2006, pp. 55–55.
- [90] J. A. Muguira and A. A. Tolk, "The Levels of Conceptual Interoperability Model A Traditional View of Levels of Information Systems Interoperability The Levels of Conceptual Interoperability," no. September 2003.
- [91] W. Wang, A. Tolk, and W. Wang, "The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S," *Proceedings of the 2009 Spring Simulation Multiconference*, p. 9, 2009.
- [92] D. Chen and N. Daclin, "Framework for enterprise interoperability 1," *Proc. of IFAC Workshop EI2N*, no. August 2016, pp. 77–88, 2006.
- [93] R. Rezaei, T. K. Chiew, and S. P. Lee, "An interoperability model for ultra large scale systems," *Advances in Engineering Software*, vol. 67, pp. 22–46, Jan. 2014.
- [94] J. Miranda *et al.*, "From the Internet of Things to the Internet of People," *IEEE Internet Computing*, vol. 19, no. 2, pp. 40–47, Mar. 2015.
- [95] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, Jun. 2017.
- [96] A. Gyrard, M. Serrano, and G. A. Ateazing, "Semantic web methodologies, best practices and ontology engineering applied to Internet of Things," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, vol. 1, pp. 412–417.
- [97] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
- [98] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "On challenges in engineering IoT software systems," in *Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18*, 2018, pp. 42–51.
- [99] M. Jackson, "Behaviours as Design Components of Cyber-Physical Systems," in *Software Engineering*, Springer, 2015, pp. 43–62.
- [100] N. F. Soria Zurita and I. Y. Tumer, "A Survey: Towards Understanding Emergent Behavior in Complex Engineered Systems," in *Volume 7: 29th International Conference on Design Theory and Methodology*, 2017, p. V007T06A015.
- [101] IEEE, *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society Press, 2004.
- [102] S. K. Datta, J. Häerri, C. Bonnet, and R. P. F. D. Costa, "Vehicles as Connected Resources: Opportunities and Challenges for the Future," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 26–35, 2017.
- [103] R. Nawaratne, D. Alahakoon, D. De Silva, P. Chhetri, and N. Chilamkurti, "Self-evolving intelligent algorithms for facilitating data interoperability in IoT environments," *Future Generation Computer Systems*, vol. 86, pp. 421–432, 2018.
- [104] F. Salamone, L. Belussi, L. Danza, T. Galanos, M. Ghellere, and I. Meroni, "Design and development of a wearable wireless system to control indoor air quality and indoor lighting quality," *Sensors (Switzerland)*, vol. 17, no. 5, 2017.
- [105] F. Salamone, L. Belussi, L. Danza, M. Ghellere, and I. Meroni, "An open source low-cost wireless control system for a forced circulation solar plant," *Sensors (Switzerland)*, vol. 15, no. 11, pp. 27990–28004, 2015.
- [106] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
- [107] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [108] E. Vassev and M. Hinchey, "Integration and promotion of autonomy with the ARE framework," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9952 LNCS, pp. 689–703, 2016.
- [109] E. Gambi, L. Montanini, L. Raffaelli, S. Spinsante, and L. Lambrinos, "Interoperability in IoT infrastructures for enhanced living environments," *2016 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2016*, 2017.
- [110] G. Weichhart, W. Guédria, and Y. Naudet, "Supporting interoperability in complex adaptive enterprise systems: A domain specific language approach," *Data and Knowledge Engineering*, vol. 105, pp. 90–106, 2016.

13 APPENDIX A - AXIAL CODING DESCRIPTION

In the axial coding process, the fractions of data from the open coding can be reassembled and organized into the categories and subcategories with their descriptions, properties, and dimensions. The subcategories, together with their respective codes, are part of the whole category concept under consideration. Therefore, the 109 codes were grouped into 41 subcategories and organized into ten categories. This appendix briefly complements the other categories. The codes indicate their groundedness (g: number of excerpts) and density (d: number of articles).

13.1 Level Category

From the 17 papers used as population, 14 of them discussed the **interoperability level** concept. Different authors and projects give different terms such as facets, maturity, aspects, degree, category, dimensions, among others, to present the same idea we identified as Level. We identified 34 excerpts associated with ten codes concerned with interoperability levels. The data analyzed (excerpts and their codes) provided enough information to organize them into subcategories and observe four different Interoperability Levels (Fig. 10): **technical**, **syntactic**, **semantic**, and **organizational**. In Fig. 10, a larger circle with high lines represents the Level category and smaller circles with thin lines their subcategories. All the following categories will use the same notation.

Excerpts	Open code	Subcategory
"Technical interoperability is concerned with connectivity between the computational services, allowing messages to be transported from one application to another." [37]	Concerned with connectivity	Technical
"This type of interoperability often focuses on communication protocols and the infrastructure required for those protocols to function." [30]		
"This level includes the syntax consistency of service description, registration, and invocation." [35]	Concerned with syntax consistency	Syntactic
"The messages transferred by communication protocols should possess a well-defined syntax and encoding, even if only in the form of bit-tables." [30]		
"After the consistency of the transmitted data set is ensured, the interpretation of the data by each individual processor must be investigated." [30]	Concerned with interpretation of the exchanged information	Semantic
"Semantic interoperability is the ability of two or more systems to interpret the meaning of exchanged information automatically, and to accurately produce useful results for end users (of both systems)." [29]		
"Interoperability is the ability for two systems to understand one another and to use the functionality of one another." [36]	Concerned with reciprocal understanding between interacting entities	Semantic
"Semantic interoperability means that the message content becomes understood in the same way by senders and receivers, both regarding information representation and messaging sequences." [37]		
"Business rules and policies must be agreed upon at the organization level" [47]	Concerned with business rules, policies and constraints	Organizational
"The ability of entities to align and match their business and legal rules for conducting legitimate automated transactions." [30]		
"Organizational interoperability creates cohesion amongst approaches to governance, finance, legislation and business processes." [36]	Concerned with align process between interacting entities	Organizational
"The ability to align processes of different entities (enterprises), for them to exchange data and to conduct business seamlessly." [30]		

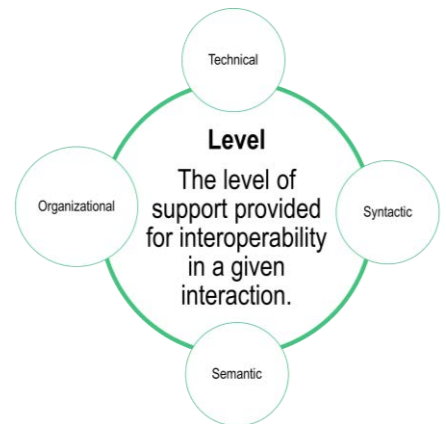


Fig. 10. Detailed Level Category.

Subcategories definitions:

- **Technical:** it is concerned with the connectivity, communication, and operation regarding the interacting entities, and all middleware elements. It is important to consider the architectural style, issues regarding authentication and authorization,

the use of technical standards, protocols for communication and transport, and an interface between the *things* - when applicable to enable interaction. This level facilitates the interaction between *things*. Three associated codes: **concerned with communication** (g:4; d:4), **with connectivity** (g:4; d:4), and **with operations** (g:1; d:1).

- **Syntactic**: this level is concerned with communication, data exchange, and syntax consistency. It is essential to consider the use of standards and protocols and provide the proper infrastructure needed and issues regarding data definition - when appropriate - to enable this kind of interaction. This level allows the interaction of *things* through data. Two associated codes: **concerned with communication and data exchange** (g:3; d:2) and **with syntax consistency** (g:2; d:2).
- **Semantic**: it concerns with the interpretation and mutual understanding between interacting entities. It is essential to consider, when applicable: the use of semantic web and a priori solution to be decided at design time, the use of standards, models and ontologies, and the sharing of compatible concepts to enable the interaction at this level. This level allows mutual data understanding between *things*. Two associated codes: **concerned with the interpretation of the exchanged information** (g:2; d:2) and **with the mutual understanding between interacting entities** (g:10; d:8).
- **Organizational**: this level is concerned with business rules, policies and constraints, process alignment, and the actions necessary to make the entities to collaborate. To enable interactions at this level, it is essential that the assets and structures use compensation or negotiation, use standards, and protocols, share common concepts and process - when applicable. This level enables the mutual understanding of the data meaning between *things*, under determined conditions. Three associated codes: **concerned with actions necessary for the collaboration** (g:4; d:4), **with align process between interacting entities** (g:2; d:2), and **with business rules, policies and constraints** (g:2; d:2).

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. Therefore, 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.

With this definition, we can distinguish in what levels systems interoperate. The **technical** concerns are related to system connectivity, communication, and operation. Moreover, the technical concerns are very different from the **organizational subcategory**, which is concerned with security and sensitive data. We highlight that although these levels have been identified for articles in the CASS scope [54], [67], they can also be applied to other similar concepts that rely on contextual information.

13.2 Purpose Category

From the 17 papers used as the population in this analysis, 14 of them introduced the Purpose concept. We identified 11 excerpts associated with six open codes regarding the interoperability purpose. The codes are equivalent to six subcategories related to purpose, shown in Fig. 11. Six associated codes: **to achieve a purpose** (g:3; d:3), **to develop business** (g:3; d:3), **to improve people's quality of life** (g:1; d:1), **to improve positioning with systems implementation** (g:1; d:1), **to improve services provision to citizens and business** (g:1; d:1) and **to achieve cost-effective partnerships** (g:2; d:1).

Excerpts	Open Coding and Subcategory
"Interoperability is intended to create a capability that serves an important human purpose and/or satisfies a mission requirement." [29]	To achieve a purpose
"It refers to work in a harmonize way at the level of organization and company in spite of, for example, the different modes of decision-making, methods of work, legislations, the culture of the company and commercial approaches, etc. so that business can be developed." [38]	To develop business
"It uses technology to improve elderly people's quality of life by increasing their autonomy in daily activities and helping them feel secure, protected, and supported." [39]	To improve people's quality of life
"... The steps needed to improve their positioning with regard to system implementation and services provision to citizens and businesses." [49]	To improve positioning with systems implementation
"... The steps needed to improve their positioning with regard to system implementation and services provision to citizens and businesses." [49]	To improve services provision to citizens and business
"This enables enterprises to, for instance, build partnerships, deliver new products and services, and/or become more cost efficient" [49]	To achieve cost-effective partnerships

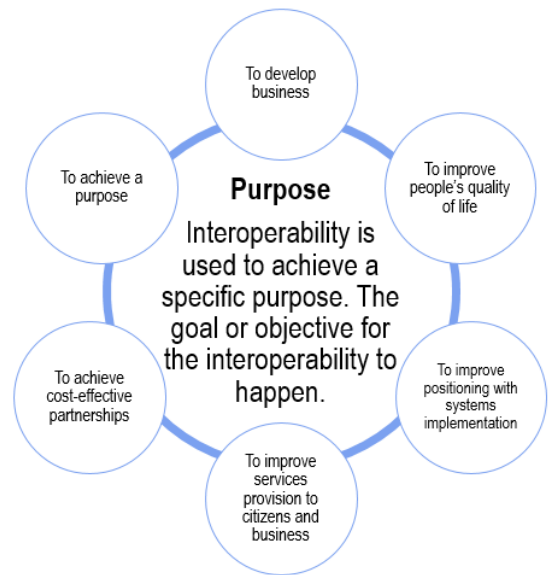


Fig. 11. Detailed Purpose Category.

We understand this concept as interoperability being used to achieve a particular purpose. Every software system is developed for a 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 own system's purpose motivates the goal or objective for the interoperability to happen. This statement may sound obvious; however, from the data, we observed that the purpose could affect how to address interoperability.

The codes **to achieve a purpose** and **to improve people's quality of life** were turned into subcategories and are based on excerpts that contribute to supporting the existence of a concept of purpose in the interoperability topic, despite not specific like the others. The other subcategories: **to develop business**, **to improve positioning with systems implementation**, **to improve services provided to citizens and business**, and **to achieve cost-effective partnerships** are different between them and more troublesome to achieve. The efforts in development, concerns in decision-making, and results can vary based on the purpose. For CASS, it is fundamental to understand the situation of the user, and his/her respective contexts serve to improve people's quality of life [58].

13.3 Perspective Category

We identified 21 excerpts associated with 12 open codes regarding the interoperability perspective. Based on those codes, we can identify the following views: **organization**, **system**, and **service** (Fig. 12).

We understand this concept as the way interoperability is seen and how it can affect decisions at different levels. Depending on the perspective, the system purpose or needs can change. The need or the fundamental purpose of which any system is built and the functional requirements are specified reflects the organization demands. This requirement can be seen differently depending on the scenario, which will require actions and solutions accordingly. Therefore, this category refers to the interoperability concept interpretation and application in different domains.

Internally, several discussion sessions were held to organize the data in the presented form. We understand the **organization** to cover business, military, and health perspectives, based on the data retrieved. However, this does not prevent other perspectives from being included in this concept. The perspective here is to consider the organization aims, culture, and values when different computing technologies have to interact to contribute to them. For CASS, the context should reflect the interacting organizations, and the interoperability enables the conformance between the various context representations. Three associated codes: **concerned with applications that have to interoperate** (g:2; d:1), **organization policies and capabilities** (g:4; d:2), **use technology to interoperate with others to conduct business** (g:1; d:1).

We also understand that the software service implements systems communication, but we decided to organize into separate perspectives based on the codes and grounded data. **System** perspectives refer to physical, technical infrastructure and stakeholders' concerns (for which the needs are translated into functional requirements). Four associated codes: **concerns be**

available for runtime composition in a self-administrative manner (g:1; d:1), concerns instant and seamless interaction among systems (g:1; d:1), concerns systems to interact without special effort from stakeholders (g:1; d:1) and relates to syntactic and semantic level (g:2; d:2).

Service relates to computer-based applications providing a software service and the concerns regarding this provision, that meaning to make the software service available. Five associated codes: concerns to be available for runtime composition in a self-administrative manner (g:2; d:2), concerns to register dynamically, aggregate and consume composite services of an external source (g:3; d:3), concerns to protocol, signature and semantic (g:2; d:2), concerns to service-oriented architectures (g:1; d:1) and relates to technical and semantic level (g:1; d:1).

Excerpts	Open Code	Subcategory
"The concept of Business Interoperability goes beyond IT, into organizational aspects of businesses, and includes the level of people-to-people interactions." [49]	Use technology to interoperate with others to conduct business	Organization Perspective
"Solutions that allow an enterprise to operate, make decisions, exchange information within and outside its boundaries." [36]		
"System interoperability needs to address both syntactic interoperability and semantic interoperability." [29]	Relates to syntactic and semantic level	System Perspective
"The ability of two specific business system implementations to interoperate. This includes the use of compatible data structures, functionality, and orchestrations." [36]		
"The Interoperability of services: It is concerned with identifying, composing and making function together various applications (designed and implemented independently)." [38]	Concerns to dynamically register, aggregate and consume composite services of an external source	Service Perspective
"At the service level, both technical (compatibility between service signatures) and semantic interoperability (semantics and behavior of services) between service endpoints must be established." [47]	Relates to technical and semantic level	

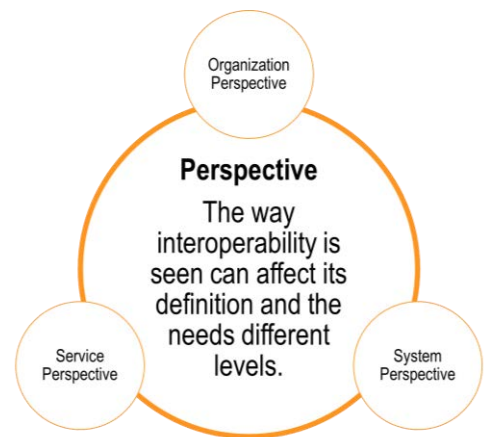


Fig. 12. Detailed Perspectives Category.

13.4 Benefits Category

We found 11 excerpts that generated seven codes in the open coding step. We grouped these codes into four subcategories during axial coding stage (Fig. 13).

For the subcategory improves understanding, two associated codes: allows better understanding and meaningful interaction of systems (g:3; d:2) and enables information to be universally used (g:1; d:1).

For the increases flexibility subcategory, three associated codes: compose new systems based on existent capabilities (g:1; d:1), enables reuse (g:3; d:2) and increase systems flexibility (g:1; d:1). For the hides complexity subcategory, one associated code: hides systems complexity from the user (g:1; d:1). For the creates cohesion subcategory, one associated code: creates cohesion among entities (g:1; d:1).

We understand this concept as the advantages observed when interoperability is achieved. Some of the benefits and the interoperability itself contributed to the system's capabilities, as required by the user and desired in ubiquitous computing. For CASS to achieve the interoperability benefits it is necessary to adapt to specific contexts like "to be universally accessible and comprehensible by all transaction parties" [48], it considers accessibility, which may vary depending on the user.

Excerpts	Open Code	Subcategory
"Similarly, interoperability can reduce the cost of creating new capabilities by allowing existing systems to be reused in multiple ways for multiple purposes." [29]	Compose new systems based on existent capabilities	Increase Flexibility
"Interoperability offers a number of advantages including increased flexibility, by allowing mixing and matching of systems." [29]	Increase systems flexibility	
"An unheralded advantage of interoperability is that it hides overall system complexity from users by creating the illusion of an integrated system." [29]	Hides systems complexity from the user	Hide Complexity
"Interoperability creates cohesion amongst approaches to governance, finance, legislation and business processes." [36]	Creates cohesion among entities	Create Cohesion
"Interoperability can enable a better understanding of telemonitoring systems for lifestyle, genetics, physiology, and pathology for the management of personal health." [48]	Allows better understanding and meaningful interaction of systems	Improve Understanding
"The ability of data to be universally accessible, reusable and comprehensible by all transaction parties, by addressing the lack of common understanding..." [30]	Enables information to be universally used	

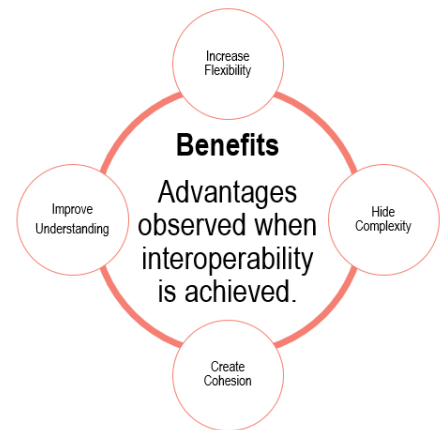


Fig. 13. Detailed Benefits Category.

13.5 Issues Category

When dealing with interoperability, some problems and concerns can arise. The analysis performed identified 49 different excerpts and 12 open codes that were associated with interoperability issues. The issues were grouped into three subcategories, as shown in Fig. 14. These subcategories represent where the issues can be observed.

For the **related to levels** subcategory, three associated codes: **related to semantic** (g:20; d:8), **related to syntactic** (g:3; d:2) and **related to technical** (g:13; d:7).

For the related to planning subcategory, three associated codes: increase costs (g:1; d:1), modifying a system can bring problems not previously planned (g:1; d:1) and interoperability needed after implementation (g:2; d:1).

For the related to perspectives subcategory, five associated codes: complexity drivers to achieve interoperability (g:1; d:1), different incompatibilities have different impacts (g:2; d:2), it is a concern that involves different perspectives (g:2; d:2), non-technology aspects of interoperability (g:3; d:2) and privacy and security issues (g:1; d:1).

This concept represents the issues reported when addressing or not interoperability. 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**.

Excerpts	Open Code	Subcategory
"Although SOA is providing the framework for integrated cross-company services or technical interoperability, it does not address the semantic interoperability problem." [36]	Related to Semantic	Related to Levels
"Naming conflicts where different names used to represent the same concepts, either homonyms or synonyms. Domain conflicts occur when different reference systems are used to measure value. Examples are different currencies." [41]		
"As important, costs can escalate from having to make systems interoperable." [29]	Increase costs	Related to Planning
"Consider errors generated by the modification of systems and changes in performance." [43]	Modification can bring problems not planned	
"Human factors and procedures address the non-technology aspects of interoperability, and incorporate the effect of established doctrine and operational plans as well as the naturally unpredictable behavior of any operation involving humans." [46]	Non-technical aspects of interoperability	Related to Perspectives
"The real interoperability challenges are stemming from various sources, such as organizational incompatibilities buried deeply into the structures of collaborating enterprises." [47]		



Fig. 14. Detailed Issues Category.

13.6 Challenges Category

Beyond the benefits and issues, there are also challenges associated with interoperability. The analyzed studies present some challenges to be tackled (18 excerpts associated with eight open codes) so that interoperability evolves as a field across the several concepts presented so far. We organized them into **research** or **practice challenges** (Fig. 15).

For the research subcategory, three associated codes: address specific challenges for ubiquitous computing (g:1; d:1), interoperability "on the fly" (g:5; d:5) and the field need further development (g:3; d:3). For the practice subcategory, three associated codes: dynamically locate and integrate application functionality (g:1; d:1), legacy and new objects interoperability (g:2; d:2), protocols are not silver bullet (g:2; d:2), runtime monitoring (g:2; d:2) and unify protocols and architecture to IoT (g:2; d:2).

Research challenges relate to the academic development, to improve and focus mainly on those fields where compatibility is still low, like the areas lacking or conflicting standard developments or lacking uniform implementation of standards. For CASS, it still necessary some research to allow interoperability 'on the fly' and develop solutions for dynamic adaptation [55].

Practice challenges are related to current interoperability in use, for example, legacy and new systems interoperability – a chronic problem in the field.

Excerpts	Open Code	Subcategory
"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." [36]	Interoperability "on the fly"	Research
"Although SOA is providing the framework for integrated cross-company services or technical interoperability, it does not address the semantic interoperability problem." [41]	The field needs further development	
"Mechanisms aiming to ease and enable the device's cooperation and service discovery among distant/remote domains and spaces, both at semantic and at syntactic levels." [50]	Dynamically locate and integrate application functionality	Practice
"Using a common interoperability mechanism to dynamically locate and integrate application functionality from a large number of disparate sources, in this case, software embedded in our physical surrounding or running on mobile devices, rather than on the web." [41]		



Fig. 15. Detailed Challenges Category.

13.7 Evaluation Methods Category

We found 31 excerpts associated with seven open codes regarding assessment methods To quantify interoperability and identify causes of interoperability problems are some of our interest in the research regarding assessment. This category is associated with RQ3.

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. That means that for the development of CASS, several specific evaluation scenarios should be defined according to the context while using an evaluation method.

To organize the information related to these methods in subcategories and improve our understanding, we had to check the original references to such methods. Fig. 16 presents the evaluation methods found: **Athena** (g:2; d:2), **i-Score** (g:2; d:2), **IAM** (Interoperability assessment methodology – g:2; d:2), **LISI** (Levels of Information Systems Interoperability – g:6; d:4), **OIMM** (Organizational Interoperability Maturity Model – g:2; d:1), **QIM** (Quantification of Interoperability Methodology – g:3; d:2). The **other methods** of subcategory represent ten methods cited only one time (g:14; d:9).

Excerpts	Opens Code and Subcategory
"Several methods for assessing interoperability have previously been suggested, on a general scope the assessment methods include LISI, SoSI, LCIM, and i-Score." [42]	iScore
"This method uses the current architecture data and can involve more than one interoperability type. 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." [30]	
"Several methods for assessing interoperability have previously been suggested, on a general scope the assessment methods include LISI, SoSI, LCIM, and i-Score." [42]	LISI
"The LISI model focuses on enhancing interoperability levels of complexity within the systems [55, 58]. The five interoperability levels (0-4) are Isolated, Connected, Functional, Domain, and Enterprise, in which each interoperability level exists in a specific environment." [30]	

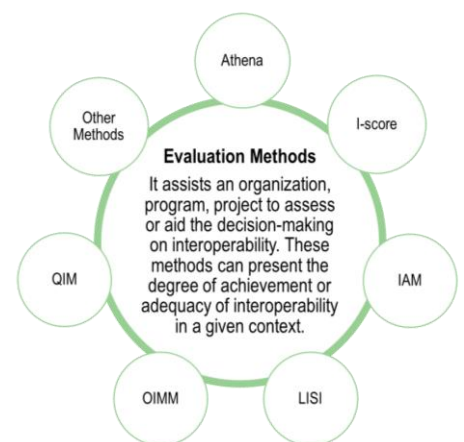


Fig. 16. Detailed Evaluation Methods Category.

13.8 Attributes Category

We found 129 excerpts associated with 26 open codes regarding interoperability elements that allowed defining two subcategories related to elements (Fig. 17): **system attributes** and **interoperability mechanisms**. This category is concerned

with systems characteristics related to interoperability and the decisions (as interoperability mechanisms) to achieve them, a form to translate the requirements into possible solutions and decisions.

This category is of particular importance within the research scope since it directly answers RQ2. During the review, we actively search the articles for elements and interoperability features in the systems. With the coding, it was possible to clarify our vision. This characterization became clearer after coding. We perceived that some elements are system attributes related to the interoperability. It also was possible to differentiate them from mechanisms, which are the technical solutions and decisions taken to reach the attributes.

System Attributes are typical systems' characteristics related to interoperability: adaptive behavior (g:4; d:3), availability (g:1; d:1), compatibility (g:13; d:5), conformance with organization requirements (g:5; d:3), conformance with system's requirements (g:9; d:6), dynamic connection (g:5; d:3) and standardization (g:6; d:5). Interoperability Mechanisms are the decisions to be taken, and mechanisms recurrently used to address.

Excerpts	Open Code and Subcategory
"This includes the use of compatible data structures, functionality, and orchestrations." [36]	Interoperability Mechanisms
"For results to be meaningful and valid, interoperable systems need to share common/compatible semantics. This requirement implies that interoperating systems must have compatible mechanisms for exchanging, representing, modifying, and updating semantic descriptions of information items." [29]	
"Collaborate in order to automatically authenticate and authorize entities and to pass on security roles and permissions to the corresponding electronic identity holders, regardless the system that they originate from." [30]	System Attributes
"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." [30]	

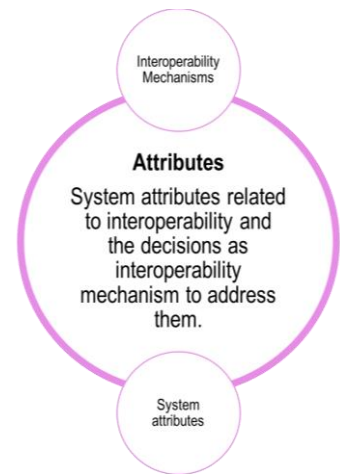


Fig. 17. Detailed Attributes Category.

14 APPENDIX B - INTEROPERABILITY DEFINITION CATEGORY DESCRIPTION

This appendix provides an *Interoperability Definition Category* description embracing all the codes belonging to it, to keep the excerpts consistency. It also presents the groundedness and density. We illustrate the subcategory with one excerpt example and the discussion regarding their relation and interpretation. The categories related to their subcategories is the type **part of**, used to represent the whole concept under discussion.

Category: Interoperability Definition		
<p>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.</p>		
<p>Groundedness: 42 excerpts are coded in this subcategory that represents six codes. Density: 11 cases support it.</p>	<p>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" [47]</p>	<p>Discussion: the interaction information exchange. The purpose is to use this information in a meaningful way. Differences in language, interface or operation environments cannot be an obstacle.</p>
<p>System property: interoperability as a system's property. Can be seen as functionality or requirement.</p>		
<p>Groundedness: 10 excerpts are coded in this subcategory that represents three codes. Density: 7 cases support it.</p>	<p>Example: "Interoperability was first defined as a property of IT systems" [46]</p>	<p>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 consider: interoperability as an ability of the system.</p>
<p>Integration: in our perspective interoperability is not only integration, although the parts of the system need to be somehow related. Integration means to compose existing systems contributing to a larger one.</p>		
<p>Groundedness: 7 excerpts are coded in this subcategory that represents one code. Density: 6 cases support it.</p>	<p>Example: "...interoperability often serves as a surrogate for integration when independently developed, stand-alone systems are combined to provide a new capability" [29]</p>	<p>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.</p>
<p>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.</p>		
<p>Groundedness: 28 excerpts are coded in this subcategory that represents two codes. Density: 8 cases support it.</p>	<p>Example: "ability of two or more software components to cooperate despite differences in language, interface, and execution platform" [35]</p>	<p>Discussion: interaction is regulated by cooperation, which has a purpose built in its meaning, not being hampered by differences.</p>
<p>Systems relation: interoperability meaning systems put into relation. A relationship represents 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.</p>		
<p>Groundedness: 9 excerpts are coded in this subcategory that represents one code. Density: 3 cases support it.</p>	<p>Example: "Because such resources of a system are themselves systems, interoperability simply concerns relations between systems" [43]</p>	<p>Discussion: in this example, it considers the components of the system 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.</p>