2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)

Rethinking Interoperability in Contemporary Software Systems

Rebeca Campos Motta
COPPE/UFRJ
Rio de Janeiro, Brasil
rmotta@cos.ufrj.br

Káthia Marçal de Oliveira
UVHC, LAMIH CNRS UMR 8201
Valenciennes, France
kathia.oliveira@univ-valenciennes.fr

Guilherme Horta Travassos
COPPE/UFRJ
Rio de Janeiro, Brasil
ght@cos.ufrj.br

Abstract*—Interoperability is a critical software system attribute since it enables different systems interaction to support the society's daily activities. The emerging technologies composing systems-of-systems, such as smart objects, internet of things, context-aware and pervasive systems have increased its importance and scope. To better understand the evolution of interoperability and its influence in the building of contemporary software systems we undertook a *quasi*-Systematic Literature Review to collect interoperability characteristics regarding context-awareness software systems. Five dimensions (ability to exchange, cooperation, integration, system relation, and property) and seven characteristics (adaptive behavior, availability, compatibility, conformance with organization requirements, conformance with system's requirements, dynamic connection, Standardization) of interoperability emerged from the findings. They should be taken into account when developing a contemporary software system. Besides, they suggest that interoperability can be thought as *the ability of things (an object, a place, an application or anything that can engage an interaction in a system) to interact for a specific purpose, once their differences (development platforms, data formats, culture, legal issues) have been overcome.

Interoperability; Systems-of-Systems; Context-aware software systems; Systematic Literature Review; Evidence-Based Software Engineering

I. Introduction

In the last years, significant technological evolution made possible advances in several domains. For instance, the wireless network enabled the development of new systems, smart devices and the dawn of Internet of Things, opening the doors to the era of living services and global digitalization. The popularization of technology is making such things increasingly present in our daily life. Reading, running or sending a message are now technology supported activities, very similar to the future imagined by Weiser [2] in his seminal work on ubiquitous and pervasive computing.

The number of users relying on technical solutions due to the attractiveness and accessibility of technology is growing, increasing the demand for new software systems and promoting additional concerns like privacy, security, and transparency [3]. Therefore, it is necessary to increase the communication, awareness, and functionalities to address the growing demand and to allow the technical solutions to be available everywhere. At the same time, there is a need to develop computing technologies seamlessly integrated with the daily life objects. Different devices (sensors, computers, mobile technologies), services and systems-of-systems (SoS) should interoperate to achieve the envisioned objectives. The diversity of pieces of technologies is one of the distinguishing characteristics of SoS [4], which focus on diverse capabilities and functionalities. When dealing with a variety of systems and devices, connectivity challenges arise. SoS face challenges, especially when connecting new systems to the already existing ones. Aside from this diversity, some particular features, such as context sensitivity, require alternative approaches to allow the SoS development [5]–[7].

In this scenario, interoperability is an essential software system property [8] needing more attention of researchers and practitioners. According to IEEE [8] interoperability is "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged.*" However, we find ourselves in the era of global digitalization in which the emerging technologies are all around the society enabling to extend actions from the virtual to the physical world. With different current devices engaging new interactions and composing the SoS, we argue that the classical perception of interoperability needs to evolve to better support the design of contemporary software systems. Rethinking interoperability is necessary to reflect the current scenario where the integration of parts is not enough, and it should cover beyond the exchange of information to support the new functional demands.

This research aims to work in the characterization of interoperability considering the features of this new society scenario in which complex software systems become invisible and pervasive to support all daily activities. Therefore, we started capturing knowledge regarding interoperability, looking for definitions and attributes of contemporary software systems (SoS, ubiquitous, pervasive and context-aware software systems - CASS) by undertaking a secondary study supported by backward/forward snowballing procedures [10]. Next, we performed qualitative data analysis using Grounded Theory [11] to reveal evidence on five dimensions and seven characteristics of interoperability. CASS and its variations have been used as surrogate to observe contemporary systems

This work is organized as follows: Section II presents the research methodology, which combines *q*SLR and Grounded Theory. Section III discusses interoperability according to the findings. Section IV concludes our discussion.

978-1-5386-2799-0/17 $31.00 © 2017 IEEE
DOI 10.1109/JSOS.2017.5

II. RESEARCH METHODOLOGY

A. Collecting Data: quasi-Systematic Literature Review

Systematic Literature Reviews (SLR) are a type of secondary study [10] aiming to identify all relevant material that meets the research purpose. At least, four perspectives (Population, Intervention, Comparison and Outcome) are expected to guide the research protocol. Since our study does not use the comparison dimension but keeps all formalism expected in SLR, we classify it as a *quasi*-Systematic Literature Review (*q*SLR) [9]. This paper presents the *q*SLR protocol main features. For further information on keywords, selection criteria, trials and initial analysis, the complete research protocol is depicted in [12] and [13].

This research protocol followed [10] [14]. Its research goal (GQM based [15]) is to **analyze** interoperability **with the purpose of** characterizing **regarding** its characteristics and measurements **from the point of view of** software engineering researchers **in the context** of CASS. The following research questions compose the protocol: **(RQ1)** How interoperability is discussed in CASS? **(RQ2)** Which are the interoperability characteristics used in CASS? **(RQ3)** How to evaluate interoperability in CASS?

This paper targets the discussions on RQ1 and RQ2. RQ3 is not addressed at this moment. The selected search engine was Scopus, which covers most of the leading digital libraries. It has been combined with snowballing procedures to enlarge coverage and provide a representative set of papers to a characterization work [12], [16]. Four researchers participated together and interactively in each step to reduce selection bias. Therefore, doubts or changes were discussed to reach a consensus.

TABLE I. DATA EXTRACTION FORM

Interoperability definition	*Verbatim*, as presented in the article.
Interoperability dimension	The range variation of interoperability as presented in the article.
Interoperability perspective	The domain where interoperability is considered.
Interoperability characteristic	Name or list of interoperability characteristics and how they were derived.
Interoperability characteristic evaluated	Name or list of interoperability attributes evaluated in the presented technique/ article.
Interoperability measures or method	Type and value of the measurement system used to evaluate the presented interoperability or the attributes.
Pre-existent approach	To indicate whether the article presents an adaption or evolution of an existent approach.
Conditions	Limitations or restrictions for applicability.
Software systems category	Main setting of software category
Experimental study type and data	Study strategy and Any information regarding the design, experimental variables and threats to validity when available.

The execution of searches resulted in 408 articles, with six duplicates and 42 conference proceedings (removed by the selection criteria). Title and abstract reading filtered the remaining 360 papers, selecting 16 candidate papers. The snowballing performed from these candidate papers revealed nine additional primary sources. Then, the 25 candidate papers passed through full reading, data extraction, and quality assessment. In the end, the final set included 17 papers.

From the included ones, we collected some pre-defined data (Table I), which allowed to observe new ways of interaction due to the development and use of new systems and applications. In addition to the exchange of information, contemporary systems can also sense and/or react to their environment, relying on context-awareness. Also, the interaction engaged can change depending on the purpose, such as the case of pieces of software that can be part of a larger SoS. These remarks motivated us for a deep analysis of the findings to better understand the value of interoperability in this new scenario.

B. Analyzing Data: Grounded Theory Procedures

The 17 included papers provided a significant amount of qualitative data related to interoperability definition. Aside the initial analysis usually performed in SLR, the principles, and procedures of Grounded Theory (GT) [11] were used to assist us to develop and analyze the concepts related to interoperability in this study. We choose to use GT due to the lack of a unified approach in the technical literature regarding interoperability in CASS. The analysis emergent from data based on primary studies offers a high potential to fill this gap. However, this analysis does not intend to reach the deepest level of generating a theory [11]. However, we decided to follow all GT procedures to support our analysis, observing similarities and differences between the several definitions found as a mean to answer our RQ´s.

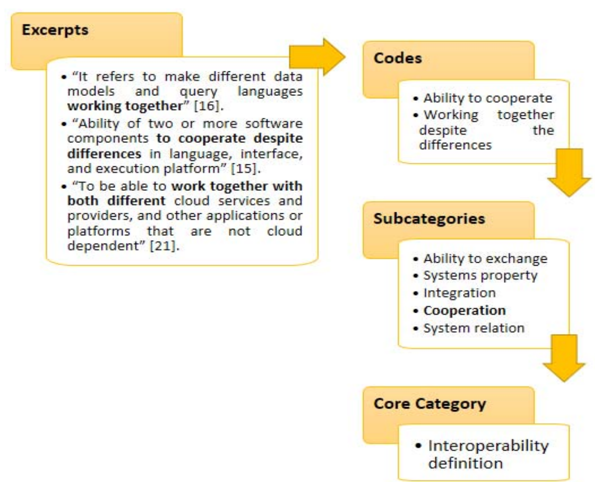

Fig. 1. An example of abstraction changes in the coding process.

We based our procedure on textual analysis, using codes to assign concepts to a portion of data. In this study, we followed [11] and performed the coding process in three steps: **open, axial***,* and **selective coding**. During the **open coding,** the goal is to identify concepts, comparing similarities and differences. When relevant for the analysis, we identified *excerpts* of data and assigned *codes* to them. Whenever coming across another *excerpt* that seemed to talk about the same concept, we grouped the respective *code* into a *subcategory*. Then, in **axial coding,** *codes* are associated to *subcategories*. This step requires some abstraction since the *subcategory* represents every *code* related to it. In **selective coding,** we identified a core category to express the main research idea and to relate *subcategories* from axial coding.

Fig. 1 shows the data analysis evolution with the increasing level of abstraction during the coding process. The *excerpts* and initial *codes* "ability to cooperate" and "Working together despite differences" were established during the **open coding** and they compose the *subcategory* "cooperation," organized in **axial coding.** "Cooperation" is part of the "Interoperability Definition" the core category from the **selective coding**.

All the coding was performed by two researchers and then integrally revised by a third one, all participants of the *q*SLR. The followed procedure was to review each extraction and the respective code in the same order they appeared, contributing to the constant comparison. Every match was then classified and justified by the reviewers as:

- **Agreement** - the code is adequate to represent the excerpt and is consistent in all the segments. In this case, we keep the original code. **Example**: ***Excerpt***: "It refers to make different data models and query languages working together." [17]. ***Initial and Final code:*** Working together.
- **Partial agreement** – the code should be modified to address the excerpt completely. **Example**: ***Excerpt:*** "the ability of systems, units, or forces to provide data, information, materiel, 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" [18]. ***Initial code:*** Ability of systems to provide, accept and use resources. ***Discussion:*** The generalization of the term resources, for both information and material, not being in the same level of abstraction. ***Final code:*** Ability of systems to provide, accept and use resources and information.
- **Disagreemen**t - suggest other code for the excerpt. **Example**: ***Excerpt:*** "Interoperability can be considered as being a problem, which can only arise when some resources are put together to interoperate. Because such resources of a system are themselves systems, interoperability simply concerns relations between systems." [19]. ***Initial code:*** Relation between systems. ***Discussion:*** The stronger concept should be "resources put together to interoperate" leading to cooperation. ***Final code:*** Ability to cooperate.

The reviewers discussed to achieve a consensus and full agreement in every code as a way to decrease potential misinterpretation and bias. This strategy was beneficial to keep the critical sense in the analysis, preserve consistency and learn from the knowledge exchange among researchers.

III. Results

A. From the Literature Review: Demographic Data

After discussions and assessments against the selection criteria, we included the 17 papers in the final set for analysis. This set shows us that the topic, interoperability in context-aware systems, is broad and widely discussed in different perspectives as Ubiquitous systems (25%), Systems-of-systems (12%), Service (19%) and Organizational domains (44%). Various points of view can lead to various interoperability interpretations in the respective domains. The way one looks at interoperability can change the definition. As a result, it can improve the understanding and the needs of interoperability for a given system. The included papers were also classified from the perspective of empirical software engineering study strategies [20]. Consequently, the following study types drove the papers organization: case study (16%), laboratory experiment (5%), survey (10%), basic research (21%), normative writing (32%, non-research writings about a phenomenon covering aspects like concept development, ideas, and suggestions), and secondary study (16%). This classification was used to strength the information and results in reliability provided by each study.

The seventeen papers contributed with the following information (Fig. 2), which is the basis for the performed analysis: 70.5% present an interoperability definition and its dimensions. 76.4% present interoperability attributes and characteristics but only 41.1% report some attribute evaluation. 11.7% present an empirical evaluation of their proposal.

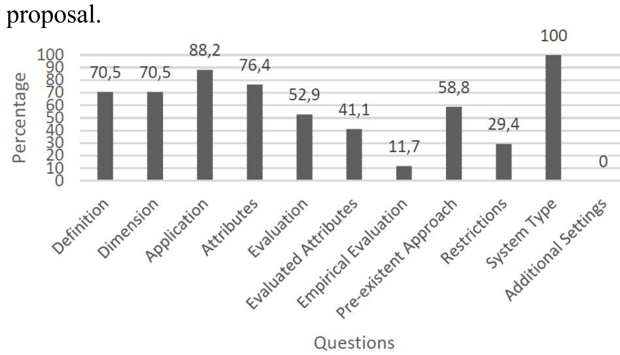

Fig. 2. General Assessment Results [9].

B. From the Analysis: Interoperability Definition

Our interest is in the interoperability views and the perspectives it has been discussed (RQ1). Although many papers provided an interoperability definition, we observed that separate definitions were assigned to the concept, having little similarity among them and with a limited vision of the contemporary scenario. Besides, many works present an attribute or characteristic (RQ2). However, we could observe discrepancies as well, either using the same term for different objects (for example, concerns the use of protocols, [21] talks about RDF and others, and [22] about WSDL, SOAP, and so on) or different terms for the same concept (for example, "Agreed upon semantics" [23] and "Semantic compatibility" [24]). These two findings motivated us to go further with the data analysis. The strength of information was also a concern. Therefore, each category has two measures to fulfill [11]: the **Groundedness (GD -** the number of occurrences, i.e. how many excerpts are related to it, there is no maximum value**)** and the **Density (DE** - how many papers support it, i.e. how far-reaching a particular code is considering the total of sources. Maximum value is 17**)** of each category. With these measures, we can observe that some subcategories are more frequent than others, leading to stronger concepts due to having more data grounding them (higher number, stronger concept).

The seventeen papers provided 96 excerpts, which were coded and organized into five subcategories, all related to the main category **interoperability definition**. With the original excerpts extracted from the primary studies and based on the performed coding and analysis, a contemporary interoperability definition emerged. The result is illustrated in Fig. 3: a larger circle with strong line represents the main category - definition - and subcategories composing them, smaller circles with thin lines. The findings presented here arose from and are grounded in research data.

Fig. 3. The core category Definition and its subcategories.

The interoperability definition (Fig. 3) emerged from the coding and discussions among the researchers during the analysis. The subcategories and examples of excerpts and codes that supported such definition are as follows:

(1) **Ability to exchange:** the exchange allows the interaction between systems. Exchange means a trade; something is given, and every transaction something is received. The concept of exchange is one of the most used to define interoperability. The trade can be data, information, and resources of different types. Here, interoperability means the system's ability to exchange something. **Excerpt example:** "Interoperability is the ability of two or more systems or components to exchange information and to use that information." [24] **Code example:** Exchanging and using resources or information. **GD:** 42. **DE:** 11.

(2) **System property:** interoperability perceived as a system's property. It can be seen as functionality or requirement. **Excerpt example:** "Interoperability was first defined as a property of IT systems." [18] **Code example**: Systems property. **GD:** 10. **DE:** 7.

(3) **Integration:** interoperability is often confounded with the integration of systems, a general misconception of these terms. In our perspective interoperability is not only integration, although parts of the system need to be somehow related. **Excerpt example:** "…interoperability often serves as a surrogate for integration when independently developed, stand-alone systems are combined to provide a new capability." [25] **Code example:** Surrogate for integration. **GD:** 7. **DE:** 6.

(4) **Cooperation:** indicates cooperation among systems, meaning to act or work together for a purpose. It covers: collaboration, operate together, work together, harmonized work, function together, function jointly, operate in synergy. **Excerpt example:** "Ability of two or more software components to cooperate despite differences in language, interface, and execution platform" [22]. **Code example:** Working together despite differences. **GD:** 28. **DE:** 8.

(5) **Systems relation:** relation 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 depending on the purpose. **Excerpt example:** "An interoperability problem appears when two or more incompatible systems are put in relation. Interoperability per se is the paradigm where an interoperability problem occurs." [19] **Code example:** Relation between systems. **GD:** 9. **DE:** 3.

C. From the Analysis: Interoperability Characteristics

More than saying that a system can interact with another one, we are concerned with the state that any software system or component for which interoperability is a need should have characteristics and mechanisms in common, because, without them, interoperability cannot be achieved. A fragmented environment of technological solutions can lead to a future characterized by "islands of functionality," where some devices are explicitly developed to interoperate only with their set or family from the same vendor, but not with others [26], [27].

Aside from this issue, we observed in our study that interoperability is usually addressed in the initial development phase and is based on previous agreements and shared knowledge. In many cases, it is not clear if or how the systems are planned, when considering interoperability. The lack of planning or alternatives to confirm the adequacy of the interaction between the systems can have an adverse impact on usage and economic aspects.

To act against the possibility of a divided future and to contribute to the decision-making towards interoperable systems, we also investigated which interoperability characteristics are presented in the technical literature (RQ2). For that, all results presented here are from the extracted material following the same analysis procedures. From the data, 43 excerpts were coded organized into seven systems characteristics related to interoperability. In other words, if a software system has some features it will be easier to interoperate with other software systems. The seven characteristics are:

(1) **Adaptive behavior:** 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. **GD:** 4. **DE:** 3.

(2) **Availability:** it is directly related to connectivity since all the interacting systems should be available during the interaction time. Availability refers to the system being available, able to be used. In a dynamic environment, availability may vary over time. **GD:** 1. **DE:** 1.

(3) **Compatibility:** deals with what ease the system can operate with shared applications. It also covers the system's compatibility on different platforms. **GD:** 14. **DE:** 4.

(4) **Conformance with organization requirements:** each organization has features that should be aligned with the other ones to interoperate. Concerned with to the decision-making and non-technological aspects. **GD:** 5. **DE:** 3.

(5) **Conformance with system's requirements:** an explicit prescription of the general requirements that should be present in the systems intending to interoperate. If systems' constraints are not satisfied, the systems may not be capable of interoperating. **GD:** 9. **DE:** 6.

(6) **Dynamic connection:** to enable the interaction between entities it is necessary to identify them. Decisions related to connecting with the identified partners according to their permissions. Authenticate and authorize interactions in automatic connection based on previous interactions (history) or re-establish lost connection. **GD:** 5. **DE:** 3.

(7) **Standardization:** bringing conformity between the systems that have to interoperate. Some decisions cannot necessarily be a current standard but should have an agreement by all parties wishing to interoperate, to ensure compatibility and integration with different systems. Despite the efforts in the field to spontaneous interoperability, where no a priori knowledge is shared, so far standardized solutions are necessary to achieve interoperability. **GD:** 6. **DE:** 5.

Despite the data simplicity as presented here, which seems to be elementary to some people, the problems with interoperability are real and out there, with also the risks associated with the development or acquisition of new technologies. The intent of these characteristics is to reflect what is present in the technical literature. More than one extensive list of features or definitive list of solutions, we would like to understand the research being conducted systematically, foster some light in the presented issues, and to be used as a guidance to **overcome the differences** (in practice or research), as the proposed definition states.

IV. Discussion

A. Regarding the Interoperability definition

During this study, we came across some widely known interoperability definitions and tried to analyze those considering aspects of contemporary systems [28]. The idea of this study is not to propose "yet another definition" without looking to knowledge built so far. That is why we undertook a *q*SLR added by a qualitative data analysis procedure.

As technology evolves, the software systems are becoming more elaborate. If we consider the context variance, for instance, it is required systems modifications at runtime to act according to the changes. For this reason, an adaptive behavior is desirable since it refers to the system ability to adapt itself dynamically to the environment in use; that calls for monitoring the systems environment and adjusting it to a different condition. Alongside, the concept of components has reached a larger scale of use; that goes from big building blocks for system-of-systems to smaller pieces of a smart environment. This characteristic requires different composition, which is based on what is available to create what is requested by the user. The heterogeneity of devices is also a concern since the application can migrate among devices and adjust itself to each one of them. With this aspect in mind, we discussed the findings of the review.

A recurrent perception during the data analysis is that interoperability is usually interchanged with the concepts of connectivity, integration or exchange. However, if a researcher or developer considers interoperability as integration, the effort and results of the work will vary from another one in which interoperability is set as communication. Moreover, the definitions can provide the foundations for any further development. Therefore, understanding this concept using general software system perspective is the first step towards its comprehension and observation in other contemporary scenarios, such as CASS and its variations.

From the excerpts analyzed and coded, it has emerged the interoperability definition in Fig. 3. This definition is generic enough to cover the definitions we found while also reporting the complexity of the interoperability concept. We decided to use the term ***things*** because we are dealing with SoS and their components as well. However, we took advantage of the topic, also considering smart things. In this way, the definition is more comprehensive. The interaction stands for any relation where a system, component, service or thing can engage with another. That way it covers exchange and trade of any kind. Together ***for a specific purpose***, the definition embraces the idea of cooperation and collaboration, frequently related to interoperability while uniting with one of the concepts uncovered during the data analysis. Each of the interacting things are different, that is why interoperability is needed. To state these differences in the definition is a step towards a deeper comprehension of interoperability. Once the differences are known, it is easier to overcome them to achieve interoperability correctly. ***Once their differences have been overcome***, represents the solutions and decisions to be made to achieve interoperability. Therefore, the definition is not only the representation of its subcategories but also includes other concepts contributing to the understanding and definition of interoperability.

An interoperability definition is given by the American Department of Defense [29]: "*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 the emerged definition with this one, we see that system, units and forces are the things that should interact. The purpose is to operate effectively together, and the interaction is to provide, accept and use services.

Another definition is presented by IEEE [8]: "*The ability of two or more systems or components to exchange information and to use the information that has been exchanged.*" Things can embrace systems and components as presented in this definition, where the interaction is to exchange and use information. These are examples of the subcategory **ability to exchange**.

In some works, interoperability is not directly defined [21], [30], instead it is classified as a system property, or quality requirement, therefore, were also analyzed and included in a particular subcategory (**system property**).

"…interoperability often serves as a surrogate for integration when independently developed, stand-alone systems are combined to provide a new capability" [25]. In this definition independently developed, stand-alone systems are very prone to have differences that do not prevent the interaction that aims to provide a new system capability and interoperability. In our perspective, interoperability is not **integration**, although the parts of the system need to be somehow related. Integration means to compose existing systems contributing to a larger one. To differ integration and interoperability concepts is important to develop interoperable systems, which can be more complex than simple plug and play. From this point, interoperability is not only about exchanging information, but it can also cover the ability to share functions, services, and different interactions.

For the **cooperation** subcategory, we mean to act or work together for a particular purpose. "*Ability of two or more software components to cooperate despite differences in language, interface, and execution platform"* [22]. In our definition, cooperation is covered by the interaction between things, which has a purpose built in its meaning, not being hampered by differences.

Systems relation subcategory represents the way in which two or more systems can interact. If a system does not interact with others interoperability is not required. This relation can evolve and change, be more complex, depending on the purpose. A definition, for example, is provided in [19]: "*because such resources of a system are themselves systems, interoperability simply concerns relations between systems.*" In the example, it considers the components of the system as systems as well, a reality observed in SoS for example. Systems relations can be presented in different ways like a simple communication link or a constraint imposed on a system, each one with its differences.

Interoperability is concerned with identifying, composing and enabling these entities, often designed and implemented separately, to work together for a purpose. Moreover, the differences precisely arise during the interaction. Some strategies are required, such as specific mechanisms, as the characteristics presented, to overcome these differences so the interaction can happen, and interoperability achieved. The idea to consider the current differences in the proposed definition is because we understand the complexity of this task. The definitions found did not go beyond saying that interoperability enables different systems to interact. However, how can this be done? The differences should be overcome or handled in such a way as not to prevent interaction. That is why it is so important to consider the characteristics together with interoperability; the concept is complete only with the two aspects together.

B. Related Work

The work of Ford *et al.* [29] identified 34 different definitions for interoperability. The definitions presented are 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 "iScore." A review focusing on interoperability evaluation models is presented in [20]. They presented ten different models and compared them regarding twelve distinct issues at several levels, technical, syntactic and semantic interoperability. It presents different interoperability definitions from the literature revised. However, they do not discuss the idea of a unified definition neither characteristics of contemporary systems.

Despite the relevance of each of these works both highlight the need for more research in the interoperability and, like us, observe the lack of consensus in the technical literature. That is the main difference among the works. We understand that it is necessary for the interoperability concept to evolve alongside with the technologies from the different interactions engaged.

C. Adequacy and Groundedness of the Analysis

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

Adequacy of the research process: the analysis is based on the data extracted. The major categories are presented in the axial coding subsection and emerged considering its GD corresponding to the data and suitability for the substantive area of interoperability. 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.

The grounding of the findings: each concept can be linked to the codes, and then to the excerpts, where it is grounded. When a concept relates to another, the relationship is presented by examples from the data analyzed. These examples also illustrate the variations of which the concepts were examined and developed.

Despite the adequacy of the research and the grounding of the findings, different researchers may have a different understanding of the data, leading to another interpretation. These are some of the limitations of this work.

D. Threats to Validity

In the *q*SLR: Since only Scopus was used as a search engine and the search string [12] is somehow limited, may be missing some relevant studies, we performed snowballing (backward and forward) to reduce this possibility. Regarding the data extraction, the form evolved during the *q*SLR execution being revised by the authors and external researchers. Extraction bias was mitigated by always having a second researcher to revisit any extracted information.

In the GT Analysis: A common threat when analyzing data is the interpretation bias. To reduce bias, two researchers followed the GT procedures, and the results have been entirely revised by another, separately. The meaning of a concept was clarified by using a dictionary to avoid misinterpretation. QDAMiner Lite, a qualitative analysis tool, heavily supported this process.

V. Conclusion

This article presented the results of a secondary study on interoperability and how it is addressed in contemporary systems. 17 studies were selected through a *q*SLR and analyzed using GT. The results show differences in concept understanding and lack of consensus, despite the extension of the research associated with interoperability issues.

This study also identified the absence of a common interoperability definition, since we could observe in the open coding stage 96 different segments related to interoperability definitions (open coding). Therefore, these codes were grouped into subcategories (axial coding) representing five key concepts (**ability to exchange**, **system´s property**, **integration**, **cooperation** and **system´s relation**) shaping our definition of Interoperability: **Ability of things to interact for a specific purpose, once their differences have been overcome,** including a list of seven software systems characteristics regarding interoperability. These characteristics allow seeing the importance of technical decisions during systems development. To deliver interoperable solutions, the concerns about interoperability should start at the software project conception phase.

With this work, we addressed the purpose of characterizing interoperability, with the proposed definition and the identified characteristics. This characterization is the first step towards an extensive study regarding interoperability considering aspects such as evaluation methods and interoperability levels. We hope the body of knowledge organized in this paper can contribute to the progress in this field of investigation and towards a future with simple, seamless and fluid interaction among contemporary software systems.

Acknowledgment

We thank Victor Vidigal for the contribution in this study. This work is part of “CAcTUS - Context-Awareness Testing for Ubiquitous Systems” project supported by CNPq - Universal (484380/2013-3). Prof. Travassos is a CNPq Researcher (305929/2014-3).

References

[1] D. Chen, “Enterprise Interoperability Framework,” *EMOI-INTEROP*, 2006.

[2] M. Weiser, “The Computer for the 21st Century,” *Scientific American*, vol. 265. p. 94–104, 1991.

[3] L. Rocha, J. Lima, M. Maia, W. Viana, M. de Castro, and R. Andrade, “Ubiquitous software engineering: Achievements, challenges and beyond,” *25th Brazilian Symp. Softw. Eng*, p. 132–137, 2011.

[4] J. Boardman and B. Sauser, “System of Systems - the meaning of,” *SMC*, no. April, p. 1–6, 2006.

[5] E. Niemelä and J. Latvakoski, “Survey of requirements and solutions for ubiquitous software,” *Proc. 3rd Int. Conf. Mob. ubiquitous Multimed*, no. October, p. 71–78, 2004.

[6] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, p. 2787–2805, 2010.

[7] K. Rose, S. Eldridge, and C. Lyman, “The internet of things: an overview,” *Internet Soc.*, no. October, p. 53, 2015.

[8] IEEE, “IEEE Standard Glossary of Software Engineering Terminology (Std 610.12-1990),” *IEEE Comput. Soc.*, 1990.

[9] G. Travassos, P. dos Santos, P. Mian, A. Neto, and J. Biolchini, “An Environment to Support Large Scale Experimentation in Software Engineering,” *13th IEEE Int. Conf. Eng. Complex Comput. Syst*, p. 193–202, 2008.

[10] B. Kitchenham, S. Charters, “Guidelines for Performing Systematic Literature Reviews in Software Engineering,” *Engineering*, vol. 2, p. 1051, 2007. Available at: https://goo.gl/qJNW2q

[11] A. Strauss and J. Corbin, “Basics of qualitative research: Techniques and procedures for developing grounded theory,” *SAGE*, 1998.

[12] R. Motta, K. Oliveira, and G. Travassos, “Characterizing Interoperability in Context-aware Software Systems,” in *Brazilian Symposium on Computing System Engineering*, 2016.

[13] R. Motta, V. Ribeiro, G. Travassos, and K. Marçal, “qSLR Protocol - Investigating Interoperability in Context-aware Systems,” 2015. Available at: https://goo.gl/BqSClK

[14] J. Biolchini, P. Mian, A. Natali, T. Conte and G. Travassos, “Scientific research ontology to support systematic review in software engineering,” *Adv. Eng. Informatics*, vol. 21, no. 2, p. 133–151, 2007.

[15] V. Basili, “Software modeling and measurement: the Goal/Question/Metric paradigm,” *Quality*. p. 24, 1992.

[16] S. Matalonga, F. Rodrigues, G. Travassos, “Challenges in Testing Context-Aware Software Systems,” *9th Workshop on Systematic and Automated Software Testing*, 2015, no. Oct, p. 51–60.

[17] D. Chen, G. Doumeingts, and F. Vernadat, “Architectures for enterprise integration and interoperability: The Past, present and future,” *Comput. Ind.*, vol. 59, no. 7, p. 647–659, 2008.

[18] E. Wyatt, K. Griendling, D. Mavris. "Addressing interoperability in military systems-of-systems architectures." *Systems Conference*, IEEE International, p. 1-8, 2012.

[19] Y. Naudet, T. Latour, W. Guedria, and D. Chen, “Towards a systemic formalization of interoperability,” *Comput. Ind.*, vol. 61, no. 2, p. 176–185, Feb. 2010.

[20] J. Wynekoop and S. Conger, "A review of computer aided software engineering research methods*". Department of Statistics and Computer Information Systems*, College of the City University NY, 1992.

[21] G. Athanasopoulos, A. Tsalgatidou, and M. Pantazoglou, “Interoperability Among Heterogeneous Services,” *IEEE Int. Conf. Serv. Comput.*, p. 174–181, 2008.

[22] J. Fang, S. Hu, and Y. Han, “A service interoperability assessment model for service composition,” in *Proceedings - IEEE International Conference on Services Computing*, 2004, p. 153–158.

[23] R. Rezaei, T. Chiew, S. Lee, and Z. Aliee, “Interoperability evaluation models: A systematic review,” *Comput. Ind.*, vol. 65, no. 1, p. 1–23, 2014.

[24] J. Ullberg, R. Lagerström, and P. Johnson, “A Framework for Service Interoperability Analysis using Enterprise Architecture Models,” *IEEE Int. Conf. Serv. Comput.*, p. 99–107, 2008.

[25] A. M. Madni and M. Sievers, “System of Systems Integration : Key Considerations and Challenges,” *Systems Engineering,* v. 17, n. 3, p. 330-347, 2014.

[26] C. Jaren, E. Scott, “The internet of things: an overview - Understanding the issues and challenges of a more connected world,” *Internet Soc*, no. October, 2015.

[27] W. Edwards and R. Grinter, “At Home with Ubiquitous Computing: Seven Challenges,” *Proc. 3rd Int. Conf. Ubiquitous Comput.*, p. 256–272, 2001.

[28] R. Spínola, F. Pinto, and G. Travassos, “Supporting requirements definition and quality assurance in ubiquitous software project,” in *Communications in Computer and Information Science*, 2008, vol. 17, p. 587–603.

[29] T. Ford, J. Colombi, S. Graham, and D. Jacques, “A Survey on Interoperability Measurement,” *Int. Command Control Res. Technol. Symp.*, no. 937, 2007.

[30] M. Memon, S. Wagner, C. Pedersen, F. Beevi, F. Hansen, “Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes.,” *Sensors*, vol. 14, no. 3, p. 4312–41, 2014.