

Interoperability Requirement to Enhance Collaboration in Software Product Lines: A Systematic Mapping

Frâncila Weidt Neiva¹, Heleno de Souza Campos Junior¹, José Maria N. David¹, Regina Maciel Braga¹, Marco Antônio Pereira Araújo¹, Fernanda Claudia Alves Campos¹, Rita Suzana Pitangueira Maciel²

¹Department of Computer Science, Federal University of Juiz de Fora – Minas Gerais, Martelos – CEP 36036-330, Juiz de Fora-MG – Brazil

²Department of Computer Science, Federal University of Bahia – Bahia, Pituba – CEP: 41810-825, Salvador-BA - Brazil

fran.weidt@gmail.com, camposheleno0@gmail.com, {jose.david, regina.braga, marco.araujo, fernanda.campos}@ufjf.edu.br, ritasuzana@dcc.ufba.br

Abstract. *[Background]: Software Product Lines (SPLs) have been widely discussed in literature as part of solution to build systems solutions in a collaborative way. Providing interoperability support is an essential requirement to achieve effective results in collaborative systems field. Despite the importance of interoperability in SPLs, research related to this field is scarce and scattered in literature. Fulfilling this requirement is not a trivial task and a mapping study about which interoperability solutions have been tackled in SPLs has not been done yet. [Goal]: This paper aims at identifying and classifying research related to interoperability in SPLs. [Method]: Conduct a systematic mapping to identify and classify SPL interoperability solutions. [Results]: The classifications showed that syntactic was the main addressed interoperability level and that architecture and variability issues were the focus of the SPL researches. [Conclusions]: The results have suggested that an expressive number of solutions related to the syntactic interoperability level were proposed. However, some challenges remain opened, such as providing higher interoperability level support, for example, regarding pragmatic, dynamic and conceptual interoperability solutions.*

1. Introduction

The use of software plays a key role in different sectors of society causing an implicit dependency. From this dependency, the need to increase quality and agility, and to reduce costs in software development process has arisen. To address this need some trends have been adopted as globalization of software engineering and use of software architectures based on Software Product Lines (SPLs) [Sugumaran et al. 2006]. The adoption of these trends increases collaboration complexity through the creation of new dependencies between distributed software development teams and organizations. SPLs are now developed and used beyond the boundaries of a single organization [Bosch 2009]. In a collaborative effort, distributed teams and organizations create software products. Regarding significant differences in practices in different teams and organizations, SPL environments are frequently heterogeneous in terms of variability modeling approach, platform, domain vocabulary, and so on. If left untreated this heterogeneity may induce “island solutions” that restricts communication and hinders

collaboration [Dhungana et al 2011]. A software product line is a set of software-intensive systems sharing a common set of features that satisfy specific needs of a particular market segment or mission, and is widely used to build systems solutions in a collaborative way [Clements and Northrop 2001]. These systems are developed from a common set of core assets in a prescribed way. In other words, a software product line comprises systematic reuse of assets to be instantiated or to generate a personalized product. Systematic reuse benefits the agile software development in conjunction with costs reduction, however interoperability challenge increases.

Software developers, usually working globally distributed, use assets built by others developers in a SPL. The selection and combination of different assets that compose a product in a SPL must be interoperable, which is especially difficult when different developers are involved. Interoperability is an important non-functional requirement that is not trivial to be fulfilled. Providing interoperability between SPL assets in order to build the best possible product, frequently demands that all environment be prepared and adapted to enable interoperation. To achieve interoperability requirement is necessary to design an interface which enables communication between assets [Siegmund et al. 2009].

Communication process is usually explored in semiotics, mainly in the study of meaning-making, the philosophical theory of signs and symbols in linguistics field. Semiotics is divided into three dimensions: syntactic, semantics and pragmatics. Syntax acts as a sign, semantics is related with which the sign refers to, and pragmatics is related to the effect of the sign on the interpreter. This effect can be noticed depending on the context where the sign is used [Morris 1938]. These three aspects are important in the communication process, which is fundamental to reach an effective collaboration as revealed in [Ellis 1991].

Using semiotics, we can also explain interoperability in SPLs. SPLs assets need to be combined and to communicate accordingly. In other words, they need to interoperate in order to reach specified goals. Interoperability can be understood as the ability of heterogeneous applications to share procedures and data on the same equipment and/or on distinct platforms [Bernstein 1996]. Interoperability in SPLs is not a trivial requirement. Each SPL has its own particularities and needs. As a result, each SPL requires that different interoperability levels be achieved. For example, suppose a developer working in a SPL that must combine assets that better address his/her expectation. So, he/she must know enough semantic information about assets, as restrictions about its use, which concepts they represent and which nonfunctional requirements they provide. In these situations, to assure that SPL will allow composition not only considering syntactic interoperable assets, but also higher interoperability levels, it is necessary to aggregate additional meaning into a communication protocol. Furthermore, these assets are usually built by globally distributed developers, even developers from different companies. The developer may not have previous information about most of the assets, once they did not develop them or use them before. Despite of this fact, the assets have to fulfill developers' requirements in order to solve their problems properly. In this case, supporting syntactic interoperability is not enough. In order to do this, it is important to consider the context where these assets are inserted, the constraints about their use, the concepts that they represent, and the nonfunctional requirements that they have to fulfill, among others [Neiva et al. 2016]. In this way, a SPL that supports interoperability in different levels is

essential to an effective collaboration among the developers, once it enhances the use and interoperation of assets created by different developers. Without interoperability support, the reuse in SPL is hindered and developers tend to build and use its own assets instead of those already developed by others.

The problem is that despite the importance of the interoperability requirement in SPL domain to enhance collaboration, the proposed solutions related to this research field are scarce and scattered in literature. Besides, to the best of our knowledge there is no study that maps the knowledge in literature about this subject. In this way, this paper aims at systematically mapping the interoperability solutions that have been proposed in software product line literature, mainly in collaborative systems area, identifying trends and research opportunities.

In contrast to the usual process of literature review, systematic reviews and mappings are designed to reduce bias and provide a reliable picture of the current state of the art of a specific research field. Through a predefined protocol, this sort of studies follows precise and strict methodological steps to select and analyze relevant papers.

This paper is organized as follows, Section 2 presents relevant concepts to understand the classifications used in the systematic mapping. Section 3 presents our systematic mapping planning and execution. Section 4 presents the results of our systematic mapping. Section 5 presents the threats to the validity of our study. Finally, Section 6 concludes this paper, with a discussion about our findings.

2. Background

Kitchenham et al. (2004) suggested that we have to consider adopting an evidence-based approach to software engineering. The author argued that evidence-based software engineering emphasizes the need to find and aggregate evidence on a specific topic using secondary studies, such as systematic literature reviews and mapping studies, as the methodological framework for identifying and aggregating evidence [Kitchenham 2010].

Systematic maps and reviews have distinct goals. Systematic reviews aim at identifying the best practices related to specific procedures, technologies, methods or tools based on empirical evidences. This is not a goal for systematic mappings since they do not analyze in details the articles content. Instead, the systematic mappings aim at classifying and conducting a thematic analysis of literature in a specific topic of software engineering [Petersen et al. 2008] [Kitchenham et al. 2011].

This paper contributes to an evidence-based approach to investigate interoperability requirement that is essential to be fulfilled in collaborative systems based on SPLs. This systematic mapping aims to identify studies across the years related to this subject, the most active authors in the area and the main vehicle for publications. In this context, we are interested in classifying the studies according to the interoperability level addressed based on Levels of Conceptual Interoperability Model (LCIM) [Tolk and Muguirra 2003] and according to the software product line element addressed (i.e. architecture, variability, artifact repository, and configuration management). The following two subsections provide the background on the LCIM and the software product line elements, respectively.

2.1 Levels of Conceptual Interoperability Model (LCIM)

There are several models that deal with interoperability in a technical domain [C4ISR 1998], [NATO 2003]. Technical domain is about systems that were already implemented. One of these models is the “Levels of Information Systems’ Interoperability (LISI)” [C4ISR 1998]. LISI model focuses on technology (IT) issues, and are mainly concerned with communication, data exchange and service (application) interoperability.

Other model is the NC3TA Reference Model for Interoperability (NMI). This model is defined and documented within the context of the NATO C3 Technical Architecture (NC3TA) [NATO 2003]. NC3TA establishes a set of standards, architecture and reference models. It aims at evaluate if a system is able to automatic exchange well-defined structural data. The main goal of the LISI and NC3TA models is to establish measures of merit to evaluate the degree of interoperability between two existing systems.

However, to analyze interoperability only considering a technical domain is not enough to achieve an effective collaboration. To cope with this problem, Tolk and Muguira (2003) proposed the “Levels of Conceptual Interoperability Model (LCIM)”. LCIM model is able to determine whether meaningful interoperability between systems is possible in initial phases of the systems development through the alignment of conceptual models. It was firstly developed to be applied in military simulation systems within the U.S. Department of Defense but has been successfully applied in different domains as U.S. Department of Energy solutions for the power-grid [GridWise 2007]. Also, it was featured as a reference model in several contributions, such as [Tolk et al. 2007], [Dobrev et al. 2007], [Zeigler et al. 2007] and [Morris et al. 2004].

LCIM is a framework that deals with conceptual interoperability issues beyond technical interoperability and divides conceptual interoperability into layers. LCIM levels are characterized as follows: Level 0 (No Interoperability) - Stand-alone systems are included in this level; Level 1 (Technical Interoperability) - A communication protocol enables data exchange between systems; Level 2 (Syntactic Interoperability) - A standard structure for information exchange is introduced; Level 3 (Semantic Interoperability) - A common information exchange reference model is used. On this level, the meaning of data is shared in an unambiguous way; Level 4 (Pragmatic Interoperability) - The use of data – or the context of its application – is understood by the participating systems. The context in which the information is exchanged is unambiguously defined; Level 5 (Dynamic Interoperability) - In this level, the systems are able to comprehend the state of changes that occur in the assumptions and constraints that each system is making over time, and is able to take advantage of those changes. Besides, this level ensures that those involved in a message exchange expect that the effects caused by this exchange are the same; Level 6 (Conceptual Interoperability) - This is the highest level and it occurs when the conceptual model is aligned. In other words, the assumptions and constraints generated through the semantic abstraction of the real world are aligned.

2.2 Software Product Line Elements

Interoperability requirement can be related to different elements in a SPL, such as architecture, variability, artifact repositories and configuration management. For

example, software product line architecture usually defines components that have to interact to enable products development. Different architectural designs can be employed to organize these components, such as Common Object Request Broker Architecture (CORBA) and Service Oriented Architecture (SOA) [Barbosa et al. 2011].

A brief description of these concepts, based on [Pohl et al. 2005], follows:

- **Architecture in SPL:** The architecture provides a common, high level structure for all product line applications.
- **Variability in SPL:** Variability defines what can vary in software product lines. In other words, it introduces variation points for the product line.
- **Artifacts Repository:** Artifacts repository comprises a common platform where the artifacts are stored. The artifacts are interrelated by traceability links to ensure the consistent commonality definition and the variability among all artifacts in a software product line [Clements and Northrop 2001].
- **Configuration Management in SPL:** Configuration management aims to manage different versions of the artifacts that are used to develop and maintain software system.

The definitions of the interoperability levels and the different elements involved in SPL presented in this section were used to guide the mapping of interoperability solutions in SPL. Interoperability in SPL enhances the communication between the assets in order to enable products composition in an effective way.

Next section details the systematic mapping conducted in this research.

3. Systematic Mapping

This section presents a systematic mapping of the literature that we conducted to investigate interoperability solutions that have been published in the software product line domain. We organized our systematic mapping based on the main activities proposed by [Kitchenham et al. 2004], such as: planning, conducting and reporting the study.

3.1 Mapping Planning

The planning activity aims to identify the mapping objectives and to define a protocol. The protocol specifies the method to be used in the systematic mapping in order to reduce the researcher bias [Steinmacher et al. 2012]. Moreover, the systematic mapping must be reproducible and the protocol contributes to fulfill this requirement. This section summarizes our protocol.

Research Questions. This systematic mapping aims at answering the questions below:

MQ1) How many studies were published across the years?

MQ2) Who are the most active authors in the area?

MQ3) Which publication vehicles are the main targets for research production?

MQ4) How are the solutions distributed according to its interoperability level?

MQ5) Which software product line artifacts are related to the interoperability solution?

Based on the research questions, we take into account the PICOC method proposed by [Petticrew and Roberts 2008] to define our scope:

- Population (P): Software product lines.
- Intervention (I): Interoperability.
- Comparison (C): No comparison intervention.
- Outcomes (O): Solution.
- Context (C): Software.

To select the papers to be analyzed in order to answer the research questions some criteria were defined.

Inclusion and Exclusion Criteria. The process used to include and exclude a paper was organized in one inclusion criteria (IC) and three exclusion criteria (EC). The inclusion criterion was:

IC1: The study presents an interoperability solution to support software product lines.

The exclusion criteria were:

EC1: The study is not written in English;

EC2: The study is not available on the web; and

EC3: The study is an editorial or summary of keynotes, workshops or tutorials.

After the definition of the research questions and the papers inclusion and exclusion criteria, the following steps define the sources of the research papers and the search string that was performed in these sources.

Sources: Search was done in six electronic databases namely: Science@Direct (<http://www.sciencedirect.com>); El Compendex (<http://www.engineeringvillage.com>); IEEE Digital Library (<http://ieeexplore.ieee.org>); ISI Web of Science (www.isiknowledge.com); Scopus (<http://www.scopus.com>); Inspec (<http://www.theiet.org>) and ACM Digital Library (<http://dl.acm.org>).

The electronic databases were selected based on previous studies [Kitchenham and Charters 2007], [Prikladnicki and Audy 2010] and [Steinmacher et al. 2012].

Search String: The search string was defined based on the PICOC terms and synonyms related to software product line and interoperability, as follows: (*“product line” OR “domain architecture” OR “dssa” OR “product family” OR “application family” OR “core artifacts”*) AND (*“interoperability” OR “mediation” OR “mediator” OR “wrapper” OR “connector”*) AND (*“software”*).

[Silva et al. 2010] and [Siegmond et al. 2009] were known potential interest studies that were used to control if the search string was finding relevant studies.

The mapping was conducted and the details for this step are showed in the following subsection.

3.2 Conducting the Mapping

In this activity we executed the search string on the selected sources and afterwards we performed the study selection process. This activity was done by two researchers in order to avoid selection bias according to personal experience or interest. The resulting process is illustrated in Figure 1 and it comprised the following steps:

The papers titles were read, and those papers that were considered clearly irrelevant to the research question, or appeared more than one time in the results, were excluded. The included papers were analyzed based on their abstracts and keywords, considering the research questions. The papers were fully read and analyzed according to the research questions. As a result, the selected papers had information extracted into a proper form.

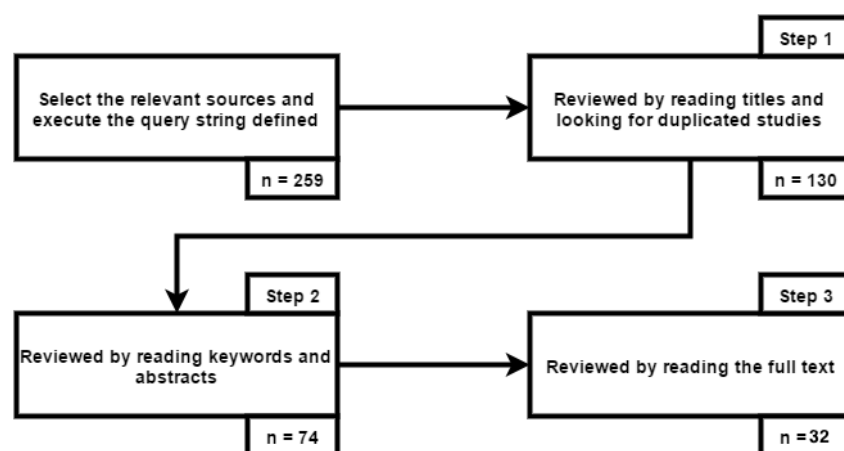


Fig. 1. Paper selection process – based on [Steinmacher et al. 2012].

In the first step, from 259 papers returned by the query, 130 (50,19%) were selected. The next step involved keywords and abstract reading. After the analysis, 74 papers were selected, which represents a reduction of 56,92%. Finally, on the last step, the review was performed by reading the full text. From 74 papers included in the previous step, 32 (43,24%) of them remained.

4. Systematic Mapping Report

The 32 selected papers were analyzed and the information to answer the mapping questions (MQ) was extracted. A summary on what could be observed on each of those questions follows.

Figure 2 represents graphically the answer to MQ1 (How many studies were published over the years?). The distribution of publications over the years is similar but we realized that there was an upward trend in the amount of publications after 2004.

This was probably due to the fact that the first conceptual model to deal with interoperability issues was proposed in 2003 by Tolk and Muguira (2003), motivated by the research agenda of USA Department of Defense Net-Centric Data Strategy. Besides, the Web 2.0 term became widespread in 2004 by O'Reilly Media enterprise increasing the interest in interoperability challenges.

It is important to highlight that the search was performed in 2015 December, so it is possible that papers which were published in this year had not been yet indexed.

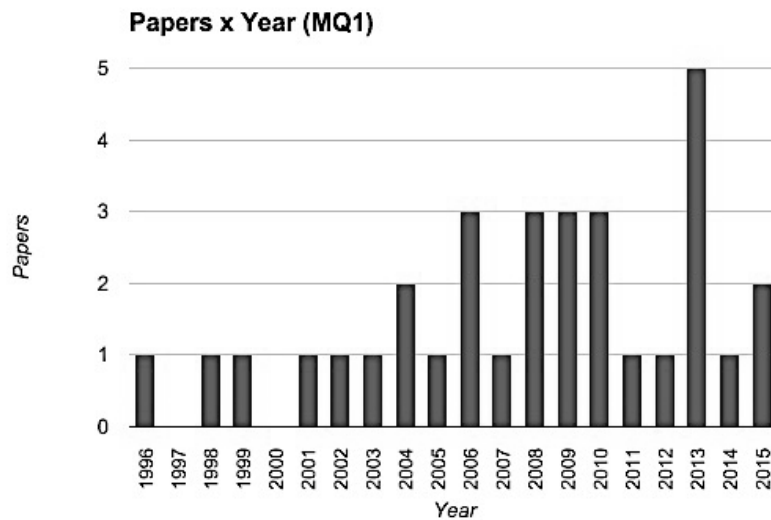


Fig. 2. Number of relevant papers per year.

Considering MQ2 (Who are the most active authors in the area?), we selected the authors that appeared twice or more times on the selected papers. The result is illustrated in Figure 3 and we can observe that the most active researchers have three publications. This fact evidences that interoperability in software product lines has not been widely explored by any group of authors.

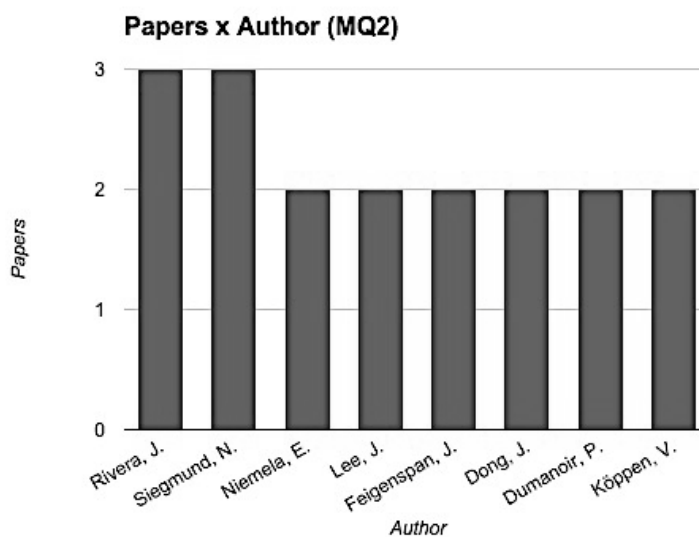


Fig. 3. Number of relevant papers per author.

Regarding MQ3 (Which publication vehicles are the main targets for research production in the area?), Figure 4 presents where the primary studies were published. Most of papers were published in conferences (46,87%), followed by workshops (25%) and journals (21,87%). In Computer Science, the number of Conferences is significantly higher compared with the number of Journals. This fact explains why most papers were published in conferences.

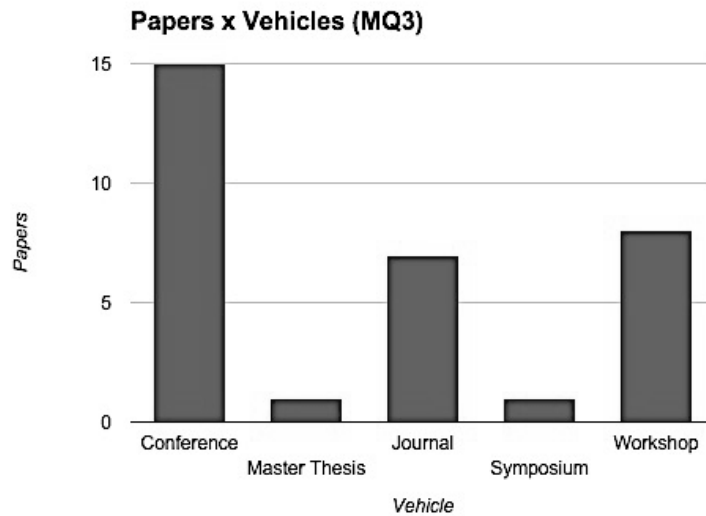


Fig. 4. Number of relevant papers per publication vehicles.

To answer MQ4 (How are the solutions distributed according to their interoperability level?), we used the LCIM to classify the papers focusing on their addressed interoperability level. Figure 5 illustrates the amount of papers and the addressed interoperability levels. First, we can conclude that syntactic was the main target level addressed. Second, we can observe that few interoperability solutions to support SPL in a pragmatic interoperability level were published. The pragmatic interoperability solutions have only been published after 2011. Pragmatic interoperability, as defined in Section 2.1, is the fourth level according to LCIM, thus there are opportunities to investigate how to satisfy higher interoperability levels in SPLs. Furthermore, we have realized that researches which investigate pragmatic interoperability in SPL are scarce in literature.

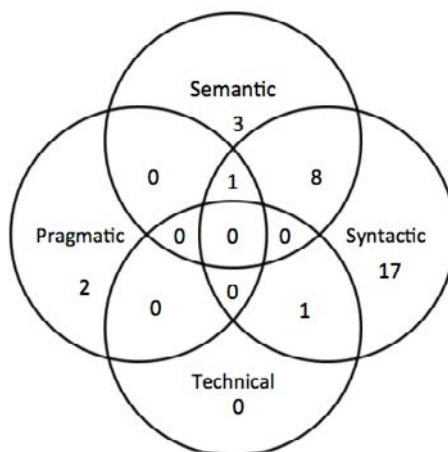


Fig. 5. Venn diagram for interoperability levels.

Finally, to answer MQ5 (Which software product line elements are related to the interoperability solution?), we classified the papers according to the addressed SPL element as defined in section 2.2. Figure 6 illustrates a classification that includes architecture, variability, artifacts repository and configuration management. This Venn diagram allows observing that architecture and variability are the main elements

addressed by interoperability solutions. The proposed solutions used the development of architecture designs as a strategy to enable the SPL to achieve interoperability. Also, once the variability introduces variation points in SPL, it is important that these variation points are aligned with interoperability purposes in order to identify variations that maintains interoperation among assets.

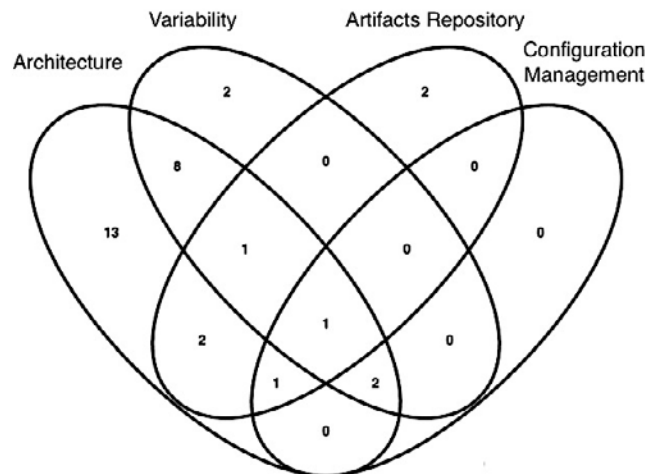


Fig. 6. Venn diagram for SPL elements.

In the next subsection, a summary of MQ4 and MQ5 research questions results is presented. Besides, the classified papers are linked to their references to allow knowing which paper belongs to which classification.

4.1 Summary of the classification according to LCIM and SPL elements

In this subsection we summarize the classification of interoperability level found in the papers according to the LCIM and its relation with the SPL element.

Table 1 provides an overview of the selected papers considering the interoperability level and the SPL elements, allowing to observe which SPL elements are supported by which interoperability level. Due to space limitations, the complete references list to the selected papers could not be presented in this paper. However, they can be found in <https://goo.gl/audSsK>. All external references are marked with a “*”.

Table 1. Distribution of papers according to LCIM and SPL element.

LCIM	SPL elements	References
Technical	Architecture	[Hodson 2007*].
	Variability	[Hodson 2007*].
Syntactic	Architecture	[Abu-matar 2009*] [Carbon et al. 2008*] [Dong 2013a*] [Dong 2013b*] [Dumanoir et al. 2009*] [Ganesan 2010*] [Garcia et al. 2013*] [Goedicke et al. 2002*] [Graubmann 2003*] [Hatzisymeon et al. 2005*] [Hodson 2007*] [Hupf and Davis 2007*] [Kaiser et al. 1996*] [Krishna et al. 2006*] [Losavio et al. 2015*] [Nascimento et al. 2011*] [Niemela and Holappa 1998*] [Niemela 1999*] [Papakonstantinou 2001*] [Razavian and Khosravi 2008*] [Rivera et al. 2008*] [Siegmund et al. 2010*] [Siegmund et al. 2009*] [Smiley et al. 2014*].
	Variability	[Abu-matar 2009*] [Ganesan 2010*] [Goedicke et al. 2002*] [Hartmann et al. 2010*]

Semantic		[Hodson 2007*] [Losavio et al. 2015*] [Nascimento et al. 2011*] [Niemela 1999*] [Razavian and Khosravi 2008*] [Smiley et al. 2014*].
	Artifacts Repository	[Chiang and Lee 2004*] [Dumanoir et al. 2006*] [Hupf and Davis 2007*] [Kaiser et al. 1996*] [Niemela 1999*] [Rivera et al. 2008*] [Smiley et al. 2014*].
	Configuration Management	[Kaiser et al. 1996*] [Nascimento et al. 2011*] [Niemela 1999*].
	Architecture	[Dong 2013a*] [Galindo et al. 2015*] [Graubmann 2003*] [Hatzisymeon et al. 2005*] [Hupf and Davis 2007*] [Lesaint and Papamargaritis 2004*] [Losavio et al. 2015*] [Nascimento et al. 2011*] [Niemela 1999*] [Papakonstantinou 2001*] [Silva et al. 2010*].
	Variability	[Galindo et al. 2015*] [Hartmann et al. 2010*] [Lesaint and Papamargaritis 2004*] [Losavio et al. 2015*] [Nascimento et al. 2011*] [Niemela 1999*].
	Artifacts Repository	[Hupf and Davis 2007*] [Niemela 1999*].
	Configuration Management	[Lesaint and Papamargaritis 2004*] [Nascimento et al. 2011*] [Niemela 1999*].
Pragmatic	Architecture	[Lee 2013*] [Nascimento et al. 2011*].
	Variability	[Lee 2013*] [Nascimento et al. 2011*] [Siegmond et al. 2012*].
	Configuration Management	[Nascimento et al. 2011*].

Table 1 allows us to observe that the number of syntactic solutions is higher than semantic solutions comparing all SPL elements. None SPL elements are supported by dynamic and conceptual interoperability levels. These observations are probably related to the fact that in order to achieve higher interoperability levels the lower levels must be established [Tolk and Muguira 2003]. Naturally, in literature we found more studies discussing lower interoperability levels and the lack of studies addressing high levels can be considered as a good research opportunity.

We also observed during paper analysis that syntactic interoperability solutions were commonly associated with a service-oriented architecture (SOA) [Abu-Matar 2009*] [Dumanoir et al. 2009*] [Losavio et al. 2015*] [Nascimento et al. 2011*] and the exchange of Extensible Markup Language (XML) files [Hatzisymeon et al. 2005*] [Hodson 2007*] [Hupf and Davis 2007*] [Rivera et al. 2008*]. These associations can be explained because (SOA) was proposed as a solution to enable distributed applications development that is an important factor to enhance collaborative systems design [Papazoglou et al. 2007]. XML standard in its turn, is used in Web Services initiatives. Web Services technologies were built to follow SOA concepts. XML is an important technology to provide interoperability across the web.

In semantic interoperability level, interoperation is enriched with semantics. To add semantic to the environment, the use of ontologies is a common strategy [Hupf and Davis 2007*] [Nascimento et al. 2011*] [Silva et al. 2010*]. Ontologies are used to store semantic information and are widely used in semantic web applications as occurred in [Silva et al. 2010*].

In pragmatic interoperability level, collaboration must achieve the expectations of the involved parties. For this purpose, interoperation is enriched with pragmatic information, such as context information. Context information can be modeled through ontologies, in conjunction with semantic information.

SPL architecture and variability were the main target for interoperability solutions, considering all interoperability levels. Providing architecture designs that

considers interoperability issues since its conception is a smart strategy to achieve interoperability in SPL [Batisda and Nieto 2007]. SPL variability must maintain the interoperation among the assets in order to produce well-functioning products.

Interoperability in SPL benefits different domains. In our set of selected papers, we identified interoperability in SPLs applied in e-business [Papakonstantinou and Vassalos 2001*], e-health [Losavio et al. 2015*] [Smiley et al. 2014*], e-learning [Razavian and Khosravi 2008*] and military domain [Rivera et al. 2008*] [Hupf and Davis 2007*] [Dumanoir et al. 2006*] [Dumanoir et al. 2009*], for example. While SPL has been adopted to increase reuse and productivity, the need of collaboration involving scenarios of distributed teams and systems reinforced interoperability as an essential requirement.

There are influences that can threaten the validity of this mapping study and must be identified. The next section presents these threats to the mapping study validity.

5. Threats to Validity

This systematic mapping intended to raise the interoperability solutions to support software product line domain applications. However, it presents some threats to validity and limitations. The results of this mapping study may have been influenced by some uncontrollable limitations. First and foremost, there might be bias regarding to the selected papers during this study. To prevent this bias, two researchers actively reviewed all the papers, agreeing on each selection.

Additionally, errors can be introduced in the protocol definition and the search string may not contain all the relevant keywords, even considering that the protocol was reviewed. As a result, this fact may cause the loss of some valuable studies. To reduce this risk, other researchers reviewed the mapping planning presented in Section 3. The search string was also evaluated using two papers [Silva et al. 2010] and [Siegmund et al. 2009] to control the returned results. We aimed to verify if they would be found by the electronic databases. The two papers appeared in the results generating evidences about the search string correctness.

Furthermore, we did not consider the Springer Link database, so it is possible that relevant studies were not indexed by the selected electronic bases. However, [Kitchenham 2010] conducted a systematic mapping only using the SCOPUS database. The author argued that SCOPUS indexes IEEE, ACM and Elsevier publications, which means that it indexes many of the leading publications. Once SCOPUS was in our set of selected databases, we believe that the selected electronic databases were enough to obtain a big picture of the interoperability research area.

6. Final Remarks

Interoperability is a key requirement to achieve an effective collaboration in collaborative systems. To cope with market needs, collaborative systems can be built using SPLs. In this paper, we presented a systematic mapping to identify and classify the interoperability solutions in software product lines. We aimed to obtain a good coverage of publications related to the targeted area. During the mapping conduction, we started with 259 papers found by the selected electronic sources. These results were filtered, which resulted in 32 papers.

From the obtained results, we identified that after the year 2004 the number of papers published in the interested area has slightly increased, reaching a peak in 2013. We also identified that there is no research group widely exploring the area. The most active researchers in the area have 3 publications.

Analyzing the content of the selected papers, we could notice that most of them (53,12%) only focused on syntactic interoperability level to support SPLs. The semantic interoperability levels were exclusively addressed in 9,37% of the papers and 25% were related to syntactic and semantic level. We can observe that other interoperability levels such as dynamic and conceptual did not appear in the selected papers.

Interoperability enables the communication between the assets to compose a product, maximize reuse and encourage developers to use assets built by others developers. Despite interoperability importance, there is a low number of research in this area. The majority of papers focused on the lowest interoperability levels and addressed architecture and variability in SPLs. Higher interoperability levels represent a long-term research agenda that need to be explored in depth.

Despite this mapping study representing an important contribution itself, it is one piece in a bigger project in collaboration system area. We are making progress in advanced studies to depth understanding strategies to achieve higher interoperability levels (pragmatic, dynamic and conceptual levels). We aim to enhance a meaningful collaboration, regarding collaborative experiments supported by a SPL infrastructure. In these advanced studies, we are considering, for example, organizational factors and the context where the involved companies and developers are inserted.

We already obtained results achieving lower interoperability levels: syntactic, semantic and pragmatic interoperability [Neiva et al. 2015]. We used these levels to enhance collaboration among scientists that collaboratively develop scientific experiments in a software ecosystem that has a SPL in the core of its infrastructure. We obtained positive results as witnessed in [Neiva et al. 2015], however we aim to explorer higher interoperability levels. As revealed by this mapping study, to explore high interoperability levels is a good contribution to literature and offer a lot of research opportunities and questions to be answered. This work can be considered as a necessary step to help researchers that intend to contribute with this research field.

References

- Barbosa, E. A., Batista, T.: Garcia, A. and Silva, E. (2011) “PL-AspectualACME: an Aspect-Oriented Architectural Description Language for Software Product Lines”. In: Proceedings of the 5th European Conference on Software Architecture, Springer-Verlag, pages 139—146.
- Bastida, L. and Nieto, F. J. (2007) “Context-Aware Service Compositions: A Way to Facilitate Interoperability”. In: Enterprise Interoperability II, Springer London, pages 427—430.
- Bernstein P. A. (1996) “Middleware: A Model for Distributed System Services.” In: Communications of the ACM. New York. vol. 39, n. 2, pages 86—98.
- Bosch, J. (2009) “From software product lines to software ecosystems.” In: Software Product Line Conference (SPLC), San Francisco, CA, USA: ACM, Carnegie Mellon University, vol. 446, pages 111—119.

- C4ISR Architecture Working Group. (1998), “Levels of information systems interoperability (LISI)”. March, 30.
- Clements, P.C. and Northrop, L. (2001) *Software Product Lines: Practices and Patterns*. Addison-Wesley.
- Dobrev P., Kalaydjiev O., Angelova G. (2007) “From Conceptual Structures to Semantic Interoperability of Content”. Bookchapter in *Conceptual Structures: Knowledge Architectures for Smart Applications*, Springer Verlag, pages. 192—205.
- Dhungana D, Seichter D, Botterweck G, Rabiser R, Grunbacher P, Benavides D, Galindo JA. (2011) “Configuration of multi product lines by bridging heterogeneous variability modeling approaches”. In: *Software Product Line Conference (SPLC)*, pages 120-129.
- Ellis, C.A., Gibbs, S.J. and Rein, G. (1991) “Groupware: some issues and experiences”. In: *Communications of the ACM*, vol. 34, n. 1, pages 39—58.
- GridWise Architecture Council Interoperability Framework Team. (2007) “Interoperability Context-Setting Framework”, V1.0, July.
- Kitchenham, B. A., Budgen, D. and Brereton, O. P. (2011) “Using Mapping Studies as the Basis for Further Research - A Participant-Observer Case Study”. In: *Information & Software Technology*, vol. 53, n. 6, pages 638—651.
- Kitchenham, B. and Charters, S. (2007) “Guidelines for Performing Systematic Literature Reviews in Software Engineering” Technical Report, Keele University and Durham University Joint Report.
- Kitchenham, B. (2010) “What’s up with software metrics? – A preliminary mapping study”. In: *Journal of systems and software*, vol. 83, n. 1, pages 37—51.
- Kitchenham, B.A., Dyba, T., Jorgensen, M. (2004) “Evidence-Based Software Engineering”. In: *Proceedings of the 26th International Conference on Software Engineering, (ICSE '04)*, IEEE Computer Society, Washington DC, USA, pages 273—281.
- Morris, C. W. (1938), “Foundations of the Theory of Signs”.
- Morris, E., Levine, L., Meyers, C., Place, P., Plakosh, D. (2004) “SOSI: System of Systems Interoperability”, vol. 5, n. 573.
- NATO Allied Data Publication 34 (ADatP-34). (2003) “NATO C3 Technical Architecture (NC3TA)”, Version 4.0.
- Neiva, F.W., David, J.M.N., Braga, R. and Campos, F. (2016) “Towards Pragmatic Interoperability to Support Collaboration: A Systematic Review and Mapping of the Literature”. In: *Information and Software Technology*, vol. 72, pages 137—150.
- Neiva, F.W., David, J.M.N., Braga, R., Campos, F. and Freitas, V. (2015) “PRIME: Pragmatic Interoperability Architecture to Support Collaborative Development of Scientific Workflows”. In: *Proceedings of the 2015 IX Brazilian Symposium on Components, Architectures and Reuse Software (SBCARS)*, pages 50—59.

- Papakonstantinou, Y. and Vassalos, V. (2001) “The Enosys Markets Data Integration Platform: Lessons from the Trenches”. In: Proceedings of the International Conference on Information and Knowledge Management, pages 538—540.
- Papazoglou, M.P., Traverso, P., Dustdar, S. and Leymann, F. (2007) “Service-oriented computing: state of the art and research challenges”. In: Computer, n. 11, pages 38—45.
- Petersen, K., Feldt, R., Shahid, M. and Mattsson, M. (2008) “Systematic Mapping Studies in Software Engineering”. In: Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering (EASE'08), pages 68—77.
- Petticrew, M. and Roberts, H. (2008) “Systematic reviews in the social sciences: A practical guide”. John Wiley and Sons.
- Pohl, K., Bockle, G., Linden, F.V.D. (2005), Software Product Line Engineering, Springer.
- Prikladnicki, R. and Audy, J.L.N. (2010) “Process Models in the Practice of Distributed Software Development: A Systematic Review of the Literature”. In: Information & Software Technology, vol. 52, n. 8, pages 779—791.
- Siegmund, N., Pukall, M., Soffner, M., Koppen, V. and Saake, G. (2009) “Using Software Product Lines for Runtime Interoperability”. In: Proceedings of the Workshop on AOP and Meta-Data for Software Evolution, RAM-SE'09, n. 4.
- Silva, A. P. D., Costa, E., Bittencourt, I. I., Brito, P. H. S., Holanda, O. and Melo, J. (2010) “Ontology-Based Software Product Line for Building Semantic Web Applications”. In: Proceedings of the 2010 Workshop on Knowledge-Oriented Product Line Engineering, KOPLE'10, ACM Special Interest Group on Programming Languages (SIGPLAN), n. 1, pages 1—6.
- Steinmacher, I., Chaves, A.P. and Gerosa, M.A. (2012) “Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature”. In: Computer Supported Cooperative Work (CSCW), vol. 22, n. 2-3, pages 113—158.
- Sugumaran, V., Sooyong, P., and Kyo, C. K. (2006) “Software Product Line Engineering”. In: Communications of the ACM, vol. 49, n. 12, pages 28—32.
- Tolk, A. and Muguira J.A. (2003) “The Levels of Conceptual Interoperability Model”. In: Proceedings of the Fall Simulation Interoperability Workshop. Orlando, Florida: Simulation Interoperability Standards Organization, pages 1—10.
- Tolk, A., Diallo, S.Y., Turnitsa, C.D. (2007) “Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering” In: Journal of Systemics, Cybernetics and Informatics, vol. 5, n. 5, pages 65—74.
- Zeigler, B.P., Hammonds, P.E. (2007), Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange, Academic Press.