Research Article / Systems and Computer Engineering https://doi.org/10.15446/ing.investig.102248

Towards a Theory of Interoperability of Software Systems

Hacia una teoría de interoperabilidad de los sistemas de software

Diana M. Torres[1], David Chen[2], Mónica K. Villavicencio[2], and Carlos M. Zapata[3]

ABSTRACT

Interoperability is a property of software quality that is related to the cooperation between software systems for exchanging information. However, this concept is not well explained or understood. A theory would be useful to explain interoperability in terms of its essential elements and propositions. Theoretical contributions of interoperability are intended to formalize this concept by using common frameworks, models, and meta-models. However, tentative contributions developed in the past have failed to propose a theory of interoperability due to four reasons: (i) a disunified vocabulary is used, (ii) the essential elements for describing interoperability are not well identified, (iii) only a single level of interoperability is assessed, and (iv) interoperability principles are not well formalized. This paper tentatively proposes an axiomatic theory of interoperability as a complementary approach to the existing knowledge. The proposed theory seeks to better formalize the concepts of interoperability and suggest actions aimed at establishing interoperability. After a brief review of related works and the state of the art, a set of axioms and propositions is presented. This theory is evaluated by a group of experts, and an example is presented to illustrate its use. Conclusions and future works are outlined at the end of the paper.

Keywords: interoperability, axiomatic theory, software system, axiom, proposition, expert consultation

RESUMEN

La interoperabilidad es una propiedad de calidad del software que tiene que ver con la cooperación entre sistemas de software para intercambiar información. Sin embargo, este concepto carece de una explicación o un completo entendimiento. Una teoría permitiría explicar la interoperabilidad en términos de sus elementos esenciales y proposiciones. Las contribuciones teóricas acerca de la interoperabilidad proponen formalizar este concepto utilizando marcos comunes, modelos y metamodelos. No obstante, las contribuciones tentativas desarrolladas en el pasado no logran proponer una teoría. Esto, debido a cuatro razones: (i) usan un vocabulario desunificado, (ii) omiten los elementos esenciales para describir la interoperabilidad, (iii) se enfocan en un nivel particular de interoperabilidad y (iv) presentan una formalización incompleta con respecto a los principios de interoperabilidad. En este artículo se propone una teoría axiomática de interoperabilidad como un enfoque complementario al conocimiento existente. Con la teoría propuesta se pretende formalizar los conceptos de interoperabilidad y sugerir acciones para establecer la interoperabilidad. Con base en una revisión de los trabajos relacionados y el estado del arte, se define un conjunto de axiomas y proposiciones. Un conjunto de expertos valida la teoría, y se expone un ejemplo para ilustrar su uso. Las conclusiones y los trabajos futuros se describen al final del artículo.

Palabras clave: interoperabilidad, teoría axiomática, sistema de software, axioma, proposición, consulta de expertos

Received: April 21st, 2022
Accepted: March 29th, 2023

Introduction

Interoperability is referred to as a software quality property (International Organization for Standardization, 2008), an ability of entities for working together (Liu *et al.*, 2020), and a feature of information systems (Turk *et al.*, 2020), among other definitions. A characteristic of interoperability is the possibility of using exchanged information, *i.e.*, understanding and interpreting the exchanged information without additional effort.

Seven essential elements have been identified and defined for describing interoperability (Torres *et al.*, 2018): software system (source and target), information, language, symbol, context, and interface (Table 1). Uniformity in the terminology of interoperability is a requisite for building a common theory.

A software phenomenon like interoperability can be explained by using a theory. The elements of a theory are constructs (called essential elements) and the relationships between them. Building a theory comprises five steps: (i) defining essential elements, (ii) defining propositions, (iii) providing explanations about the propositions, (iv)

[1] PhD in Engineering – Systems and Informatics, Universidad Nacional de Colombia, Colombia. Affiliation: Universidad Nacional de Colombia, Colombia. Email: dimtorresri@unal.edu.co

[2] PhD in Physics and Engineering, University of Bordeaux, France. Affiliation: Full professor, University of Bordeaux, IMS Laboratory, France. Email: david.chen@ims-bordeaux.fr

[3] PhD in Engineering, Université du Québec in Montreal, Canada. Affiliation: Full professor, ESPOL University, Ecuador. Email: mvillavi@espol.edu.ec

[4] PhD in Engineering – Systems, Universidad Nacional de Colombia, Colombia. Affiliation: Full professor, Universidad Nacional de Colombia, Colombia. Email: cmzapata@unal.edu.co

determining the scope, and (v) testing the theory (Sjøberg *et al.*, 2012). A theory can also be conceived as axiomatic when it introduces new concepts and deduces their properties (Tall, 2004).

Theoretical contributions regarding interoperability (Table 2) attempt to formalize, characterize, and describe it according to different perspectives. To this effect, they use common frameworks defining concepts, practices, and criteria related to interoperability; common models representing interoperability solutions; meta-models including models of problems involving interoperability; and, finally, systematic literature reviews useful for collecting proposals focused on the challenges of interoperability.

Table 1. Essential elements of interoperability

Concept	Definition
Software system (i.e., source and target software system)	*"A system consisting solely of software and possibly the computer equipment on which the software operates".* *"A system made up of software, hardware, and data that provides its primary value by the execution of the software".*
Information	*"The meaning assigned to data by known conventions. The meaning that humans assign to data by means of known conventions that are applied to the data".*
Symbol	*"A representation of something by reason of relationship, association, or convention".*
Language	*"A system consisting of: 1) a well-defined, usually finite, set of characters 2) rules for combining characters with one another to form words or other expressions 3) a specific assignment of meaning to some of the words or expressions, usually for communicating information or data among a group of people, machines, etc.".*
Context	*"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".*
Interface	*"(i) A common boundary between a considered system and another system, or between parts of a system, through which information is conveyed. (ii) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other".*

Source: Torres Ricaurte *et al.* (2018)

The aforementioned contributions have at least one of the following problems:

1. The authors refer to interoperability by using a disunified terminology. The causes are a lack of consensus about what interoperability is, the lack of a unified vocabulary, and an incomplete characterization of interoperability.
2. The essential elements describing interoperability are not explicitly identified. The causes include the following: a complete view of interoperability is missing; comparing the approaches to interoperability is difficult; and some interoperability solutions which use current approaches are missing.
3. Interoperability principles are left aside because interoperability problems are difficult to identify and the source of the problems is disregarded.
4. A general view of interoperability is not reached because the proposals are focused on a single level of interoperability, *i.e.*, technical, syntactical, semantic, or organizational. A general and complete representation of interoperability is still missing.

This paper presents an axiomatic theory of interoperability based on a set of axioms and propositions. Such an axiomatic theory involves the seven essential elements of this concept. Axioms and propositions are stated in order to explain the role of each essential element during interoperability, as well as the relationships between such elements. Moreover, a discussion about the stated propositions is carried out which involves a group of independent interoperability experts.

An illustrative example is proposed with the aim to demonstrate the applicability of the axioms and propositions in a real situation. The example is taken from a local software development company.

The proposed axiomatic theory seeks to explain the interoperability that occurs between two software systems when information is transmitted by using an interface. Such information (containing symbols) is written in a language and interpreted according to a common context. Additionally, the axiomatic theory is used for describing how the essential elements are related, characterizing related problems based on the essential elements and the rules of interoperability, as well as comparing approaches by using a unified characterization.

Theoretical contributions analysis

To gather theoretical contributions regarding interoperability, a systematic literature review methodology was applied (Software Engineering Group, 2007). This method involved the following phases:

1. *Planning*

The main research question is *What is the set of essential elements for describing interoperability?* Information sources include scientific papers from 2000 to 2022 in databases such as Science Direct, IEEE Xplore, Springer, Scopus, and Google Scholar. The query was defined as follows: (Interoperability OR Interoperate OR Interoperation OR Interoperable) AND (Systems OR Software OR "Software systems") AND (Theory OR Formal OR Axiomatic OR Formalization OR Theoretical OR General).

2) *Execution*

The search was performed once in 2022 and again in 2023 for recent studies. The following selection criteria were applied after reading the abstracts and keywords: a) papers written in English from peer-reviewed conference proceedings, journal articles, book sections, and doctoral dissertations; b) papers with title and keywords related to formalizations regarding the concept, with interoperability as a main topic, and empirical studies; and c) articles including formalizations and models of interoperability in different kinds of software systems.

3) *Results and analysis*

106 (93 + 13) studies were collected for analysis. Three kinds of study were identified: general models, meta-models, and models of interoperability. Given our interest in generality, 38 papers presenting general models and meta-models were analyzed. Table 2 reports the 19 most recent papers (2018-2023).

As a result, it was noted that theoretical contributions regarding interoperability embrace approaches proposing common frameworks, common models, meta-models, and systematic literature reviews.

Common frameworks

Standards are used for reaching a common terminology about some aspects of interoperability. However, such terminology is specific to some issues regarding the concept (*e.g.*, interoperating heterogeneous manufacturing software units) and disunified concerning other proposals.

In other interoperability frameworks, an analysis of some interoperability levels (technical, syntactic, semantic, and organizational) is performed. Interoperability is evaluated within a specific context (*e.g.*, cloud, construction, enterprise, *etc.*). Therefore, a general view of this phenomenon has not been reached. Some proposals discuss the necessary information and conditions for accomplishing systems interoperability. However, such proposals are based on a disunified terminology, and their scope is limited to a given context.

Common models

Some common models aim to implement solutions to specific interoperability problems. Thus, an abstraction of this phenomenon is difficult, and some elements are overlooked. However, the formalizations of interoperability presented in said models are useful for identifying relationships and patterns of interoperability.

Table 2. Findings of the literature review

Reference	Proposal contribution	Interoperability level				Criteria		
		Semantic level	Technical level	Syntactic level	Organizational level	Unified terminology	Identification of some essential elements of interoperability	Statement of principles (rules, propositions, etc.) of interoperability
(Mistry *et al.*, 2022)	Common framework		X		X		X	
(Sana *et al.*, 2021)	Common framework	X	X	X			X	
(Turk, 2020)	Common framework	X						X
(Lafourcade and Lombard-Platet, 2020)	Common framework		X					X
(Liu *et al.*, 2020)	Common framework				X		X	
(Brilhault *et al.*, 2022)	Common model	X						
(Jepsen *et al.*, 2021)	Common model	X	X	X				
(Haile and Altmann, 2018)	Common model	X			X			
(Ribeiro *et al.*, 2018)	Common model		X	X				
(Challita *et al.*, 2018)	Common model	X						
(Horcas *et al.*, 2022)	Meta-model	X		X			X	
(Torres *et al.*, 2022)	Meta-model	X	X	X		X	X	
(Davies *et al.*, 2020)	Meta-model	X					X	
(Delgado *et al.*, 2020)	Meta-model	X					X	X
(Serapio *et al.*, 2019)	Meta-model				X	X		
(Torab-Miandoab *et al.*, 2023)	Systematic literature review	X	X	X				
(Fraga *et al.*, 2020)	Systematic literature review	X			X			
(Burzlaff *et al.*, 2019)	Systematic literature review				X		X	
(Burns *et al.*, 2019)	Systematic literature review		X				X	

Source: Authors

Meta-models

It is possible to identify two types of interoperability meta-models: (i) meta-models focused on data representations by using formalisms (*e.g.*, categorical theory, semantic models, and mathematical foundation, among others) which have partial views of interoperability; for this reason, they are limited for addressing some interoperability problems; (ii) meta-models with a high level of abstraction about interoperability and specific perspectives of the systems (*e.g.*, business processes, software systems, and general systems theory), which are limited by their areas of interest. In both types of meta-models, the principles of interoperability are left aside, and a disunified terminology is addressed.

Systematic literature reviews

There are reviews of current research on interoperability and its future needs. These works are useful for understanding the need to formalize interoperability and for analyzing how some elements of interoperability are used in the descriptions of this concept.

The findings related to the gaps in the previous works are summarized in Table 2.

Methodology

This section proposes an axiomatic theory for explaining what interoperability is and how to achieve it. In this theory, axioms seek to thoroughly describe what interoperability is, while propositions are statements about what is needed for achieving interoperability (Figure 1). Such an axiomatic theory is based on the identified essential elements of interoperability, *i.e.*, source software system, target software system, information, context, language, symbol, interface, and their relationships.

In addition, it is necessary to understand how the seven essential elements relate to each other, what role they play in establishing interoperability (exhibited relationship), and what dependencies and attributes they have. To this effect, the literature was examined by using an adapted methodology comprising three stages: structural and functional analysis, semantic processing, and discourse mapping. As a result, a pre-conceptual schema (PCS) of the essential elements and their structural relationship was obtained (Torres *et al*., 2022).

The relationships observed between the essential elements indicate that interoperability happens at least between two software systems: a source and a target software system. By instantiating the PC of interoperability (Torres *et al*., 2022), it is possible to realize that, even though a two-way relationship is expected (*e.g.*, an answer derived from the exchange), our model allows representing such a situation while changing the roles of the software systems involved. The core element of interoperability is information, which is created by the source software system and used in the target software system. The information is written by using some language and exchanged via an interface. Symbols are part of its content. Each software system has its own context, which describes a set of software system elements used for matching and transforming the exchanged information. Furthermore, a common context is necessary for interpreting the information.

With the results of the mapping and unification process and the interoperability PC about the relationships between essential elements, a new perspective was established, aimed at reaching underlying and intuitive reasoning regarding how the essential elements should be arranged for achieving interoperability. All this, while considering the definitions of axioms as well accepted statements and the propositions as facts accepted by the expert community (Tall, 2004). This work focused on recognizing and analyzing such statements (*i.e.*, well accepted statements and facts) with regard to each essential element (including its mapped concepts), looking at statements like:

"Interoperability is characterized" AND "interoperability means" AND "interoperability requirements are" AND "the means to achieve interoperability" AND "interoperability is provided by means" AND "to ensure interoperability is necessary" AND "an interoperable system meets" AND "to establish interoperability" AND "interoperability depends/ requires".

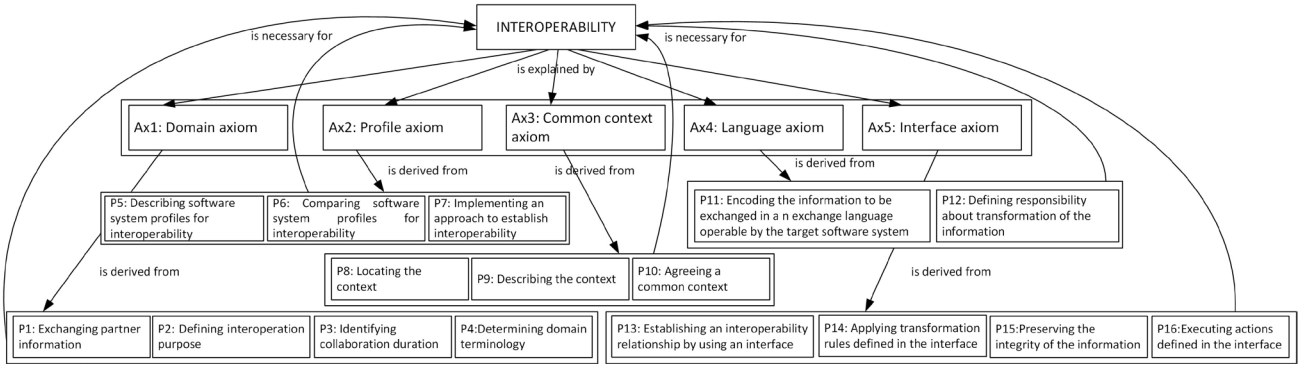

Figure 1. Representation and summary of the axiomatic theory
Source: Authors

As a result, an axiomatic theory of interoperability can be proposed which is supported by the findings of the literature (Figure 2).

Axiomatic theory of interoperability

AXIOM 1. INTEROPERABILITY DOMAIN – Exchanging domain information to prepare interoperability.

> When one needs to establish interoperability between the software systems of two organizations, the domain information of both software systems should first be exchanged for a preliminary mutual understanding of their backgrounds.

For example, if one needs to establish interoperability between two software systems in different domains such as e-Government (encompassing entities as citizens, business, administration, *etc*.) and e-health (encompassing entities as citizens, administration, e-surveillance, *etc*.), this mutual understanding (respect towards partner information, the purpose and duration of interoperability, and the domain terminology) is very important as a starting point.

Proposition 1: Exchanging partner information

> Partners should exchange information to allow their collaboration. This includes organization names, location, economic sector, people in charge of a possible collaboration, and any potential constraints.

Derived axioms and propositions	Reference
Ax1. Domain axiom	(Fraga *et al*., 2020), (Mistry *et al*., 2022)
P1. Exchanging partner information	(Delgado *et al*., 2020), (Liu *et al*., 2020), (Mistry *et al*., 2022), (Torres *et al*., 2022)
P2. Defining the purpose of interoperation	(Delgado *et al*., 2020), (Liu *et al*., 2020), (Haile and Altmann, 2018), (Mistry *et al*., 2022), (Serapio *et al*., 2019), (Torres *et al*., 2022), (Turk, 2020)
P3. Identifying the duration of collaboration	(Liu *et al*., 2020), (Torres *et al*., 2022)
P4. Determining the domain terminology	(Delgado *et al*., 2020), (Fraga *et al*., 2020), (Liu *et al*., 2020)
Ax2. Profile axiom	(Fraga *et al*., 2020), (Haile and Altmann, 2018), (Mistry *et al*., 2022), (Torab-Miandoab *et al*., 2023)
P5. Describing software system profiles for interoperability	(Haile and Altmann, 2018)
P6. Comparing software system profiles for interoperability	(Mistry *et al*., 2022), (Haile and Altmann, 2018)
P7. Implementing an approach to establish interoperability	(Mistry *et al*., 2022), (Turk, 2020)
Ax3. Common context axiom	(Davies *et al*., 2020), (Fraga *et al*., 2020), (Jepsen *et al*., 2020), (Liu *et al*., 2020), (Mistry *et al*., 2022), (Sana *et al*., 2021), (Torres *et al*., 2022), (Turk, 2020)
P8. Locating the context	(Delgado *et al*., 2020), (Liu *et al*., 2020), (Torres *et al*., 2022)
P9. Describing the context	(Brilhault *et al*., 2022), (Delgado *et al*., 2020), (Torab-Miandoab *et al*., 2023), (Torres *et al*., 2022)
P10. Agreeing on a common context	(Brilhault *et al*., 2022), (Davies *et al*., 2020), (Lafourcade and Lombard-Platet, 2020), (Liu *et al*., 2020), (Torres *et al*., 2022)
Ax4. Language axiom	(Burns *et al*., 2019), (Burzlaff *et al*., 2019), (Davies *et al*., 2020), (Fraga *et al*., 2020), (Jepsen *et al*., 2021), (Liu *et al*., 2020), (Sana *et al*., 2021), (Torab-Miandoab *et al*., 2023), (Torres *et al*., 2022)
P11. Encoding the information to be exchanged in an exchange language operable by the target software system	(Horcas *et al*., 2023), (Liu *et al*., 2020), (Torres *et al*., 2022)
P12. Defining responsibility regarding the transformation of the information	(Burzlaff *et al*., 2019), (Davies *et al*., 2020), (Liu *et al*., 2020)
Ax5. Interface axiom	(Brilhault *et al*., 2022), (Haile and Altmann, 2018), (Liu *et al*., 2020), (Sana *et al*., 2021), (Torres *et al*., 2022), (Turk, 2020)
P13. Establishing relationship of interoperability by using an interface	(Brilhault *et al*., 2022), (Challita *et al*., 2018), (Torab-Miandoab *et al*., 2023), (Torres *et al*., 2022)
P14. Applying transformation rules defined in the interface	(Brilhault *et al*., 2022), (Challita *et al*., 2018), (Jepsen *et al*., 2020), (Torres *et al*., 2022)
P15. Preserving the integrity of the information	(Fraga *et al*., 2020), (Horcas *et al*., 2023), (Torres *et al*., 2022), (Turk, 2020)
P16. Executing actions defined in the interface	(Haile and Altmann, 2018), (Jepsen *et al*., 2020), (Torres *et al*., 2022)

Figure 2. Supports of the axiomatic theory
Source: Authors

Proposition 2: Defining the purpose of interoperation

It is important to identify the purpose and what will be exchanged between the two software systems in order to be prepared for establishing interoperability.

Proposition 3: Identifying the duration of collaboration

The duration of the collaboration between two software systems must be defined, as well as the frequency of interoperations.

Proposition 4: Determining the domain terminology

In order to establish semantic interoperability between two software systems in different domains, it is important to identify the differences between their terminologies. This can be done by using an ontology, a glossary, software documentation, *etc.*

Figure 3 presents a template to provide domain information for interoperability purposes.

Organization name	
Area	(e-government, commerce, manufacturing, *etc.*)
Entity name	(name of the organization: company, firm, *etc.*)
Location	(city and country)
Activity	(*e.g.*, software provider, manufacturing company, agriculture fame...)
Purpose	(information to be exchanged: ex: billing document)
System involved	(software or information system involved at both sides)
Duration	(precise the duration: long term (years), short term (weeks) or just one session
Responsibility	(name of the person at both sides responsible for this interoperability)
Constraint	(any particular constraint(s) for this interoperability)

Figure 3. Interoperability domain template
Source: Authors

AXIOM 2. INTEROPERABILITY PROFILE – Exchanging software profile for interoperability

When two software systems seek to interoperate, they should exchange information about their software profiles in order to predict the possibility of interoperation.

Proposition 5: Describing software system profiles for interoperability

Software profiles for interoperability must be first identified and described. These are sets of characteristics describing some particular aspects of a software system relevant for interoperability, such as the coding language, the operating system, the protocol for communication, *etc*.

A template for documenting such characteristics is proposed in Figure 4.

Software system profile	
Name:	(*Name for identifying the software system—e.g., MS Excel, WhatsApp, etc.*)
Version:	(*Number for identifying the current operational software product—e.g., V 1.0.2*)
Programming language:	(*List of software system development languages—e.g., HTML, JavaScript, Python, etc.*)
Communication protocols:	(*List of the communication protocols used by the software system—i.e., http, TCP/IP, etc.*)
Operating system:	(*List of the operating systems supporting the software system—e.g., Windows 10, Mac OS X, etc.*)
Platform:	(*List of devices supporting the software system—e.g., pc, tablet, etc.*)
DBMS:	(*List of DBMSs used by the software system—e.g., PostgreSQL, MySQL, etc.*)
Information exchange format:	(*List of languages used for exchanging information—e.g., XML, Json, Atlas, etc.*)
Communication interfaces available:	(*List of interfaces for exchanging information—e.g., Export/Import Data from*)

Figure 4. Interoperability profile of a software system
Source: Authors

Proposition 6: Comparing software system profiles for interoperability

Exchanging and comparing profiles of two software systems allows knowing whether they are interoperable (according to the differences between their profiles) and consequently searching for an appropriate solution to establish interoperability.

Proposition 7: Implementing an approach to establish interoperability

Incompatibility between two software system profiles leads to failures of interoperability. Therefore, an appropriate approach (*i.e.*, integrated, unified, and federated) (International Organization for Standardization, 2011) should be implemented to establish interoperability.

An integrated approach means using a common format and language in each system. A unified approach implies building a neutral format at the meta level for mapping two systems. A federated approach involves dynamically negotiating and mapping two systems 'on the fly' (*i.e.*, without a pre-defined format/language). The choice depends on the context and the requirements of interoperability.

AXIOM 3. INTEROPERABILITY CONTEXT – Interpreting exchanged information using a common context

Exchanged information can be correctly understood when both software systems have a common context.

Proposition 8: Locating the context

> Before interoperating, both software systems should define and agree on the minimum level of detail regarding each context necessary to understand the information they are going to exchange.

Proposition 9: Describing the context

A prerequisite for establishing interoperability is to describe the context of each software system as a list of software system elements, their attributes, and their rules.

Proposition 10: Agreeing on a common context

Agreeing on a common context consists of identifying relationships between the elements of the source and the target software systems.

A common context should include the relationships stating the direct matching of the elements in both contexts, the necessary rules for transforming source elements into target elements, and rules describing how to interpret particular elements of the source software system.

Without a common context between source and target software systems, the understanding and use of information is uncertain, and it could be misinterpreted. Such lack of agreement on the exchanged information can generate conflicts in the target software system, *i.e.*, business rules violations, loss of consistency in the information, and lack of reliability with regard to the information.

AXIOM 4. INTEROPERABILITY LANGUAGE – Encoding information for interoperability

> An exchange language is used for encoding information to be sent to another software system. The exchange language should be recognized by the two software systems that are going to interoperate.

Proposition 11: Encoding the information to be exchanged in an exchange language operable by the target software system

> Information is written in the source software system by using an exchange language recognized by the target software system. The source software system has one or several exchange languages used for transmitting information.

Proposition 12: Defining responsibility regarding the transformation of the information

> During the interoperation between two heterogeneous software systems, the exchanged information should be transformed into a language/format understandable by the target software system. The responsibility of this transformation should be defined before interoperation, and it may be assigned to the source or to the target software system.

AXIOM 5. INTEROPERABILITY INTERFACE– Establishing interface for interoperability

> An interface is needed for establishing a relationship between the source and the target software systems during interoperation. A relationship between said systems implies linking their elements. When an interface is used, a clear identification of the actions to be executed in the transmission of information is required.

Proposition 13: Establishing relationship of interoperability by using an interface

> An interface is an established agreement between two software systems. This agreement includes specifications, *e.g.*, what language will be used during the exchange, what languages/translators are available for understanding the information, what are the contexts of both software systems, how the common context for the two software systems should be established, what transformation rules are necessary, what conditions are imposed in the elements of each context, and what actions are needed to perform the exchange.

Proposition 14: Applying transformation rules defined in the interface

> Transformation rules are descriptions about the way in which source software system elements are related to target software system elements. Transformations rules seek to describe, in a formal language, the following equivalences: (i) element to element, when an element of source software system is a representation of the same entity in the contexts of the two systems; (ii) attribute to attribute, when attributes of elements representing the same entity or different entities refer to the same property; and (iii) element to attribute, when an attribute is represented as an element in the other context.

Proposition 15: Preserving the integrity of the information

> Conditions in the source software context are constraints regarding the way in which the elements interact within the software system. Conditions are conveyed by using an interface to preserve the integrity of the information. A decision about the need to respect the conditions should be made in the target software system.

Proposition 16: Executing actions defined in the interface

> The interface contains actions seeking to establish interoperability between the source and the target software systems. These actions include (a) establishing communication with the target software system; (b) agreeing on the common context of both software systems; (c) transforming information from a language (chosen by the source software system) to a language recognized by the target software system; and (d) conveying the information to the target software system.

In addition, adequate data must be used for performing these actions (required data), and the results must be delivered as a dataset (provided data).

Results

Eleven experts were invited to review the proposed axiomatic theory and answer a previously approved questionnaire. The participants met the following criteria: experience in software interoperability (*i.e.*, university professors, researchers, consultants, and professionals), work related to interoperability (*i.e.*, interoperability research, software quality, interoperability projects for industry, and formal methods for software engineering), years of experience (two or more), and academic degree (*i.e.*, Master and PhD). The results regarding the level of agreement on a five-point Likert scale (strongly agree, agree, neutral, disagree, and strongly disagree) are presented in Figure 5. Table 3 also summarizes the results concerning some indices used for declaring consensus.

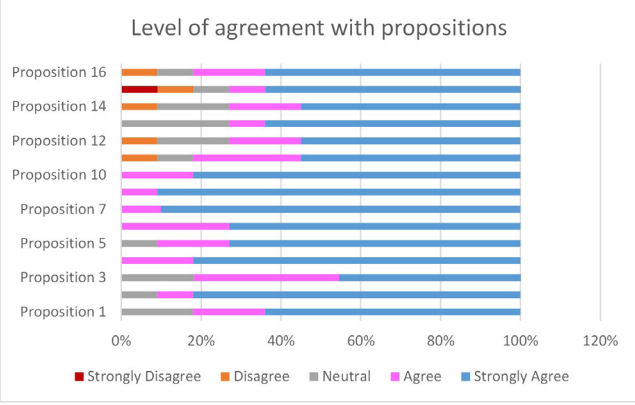

Figure 5. Expert consultation results
Source: Authors

Consensus with all propositions can be declared because there are positive results in at least three of the evaluated indices (Table 2). Nevertheless, the concerns of the experts with propositions 12-15 can be discussed based on the results in the interquartile range (IQR) measure. The expert concerns on proposition 12 are related to the transformation of the information. The experts considered that such a responsibility should be addressed in the interface or in a third system. However, we believe that such a decision should be assigned within an explicit or implicit negotiation between the two systems.

Regarding propositions 13 and 14, some experts referred to the need for a means to exchange information (corresponding to the proposed definition of interface) in the form of APIs, database links, cloud servers, web servers, *etc.* The conception of an interface as a pre-established agreement was also the main concern of other experts. Such an agreement can be unilateral, as in the case of the aforementioned mechanisms (an analysis from a practical point of view could be necessary). In addition, some experts hesitated on the need for transformation rules, as systems sometimes have equivalent profiles and contexts, and interoperability can be directly established.

As for proposition 15, the expert concerns mainly revolve around the time when the preservation of the information should happen (before or after transformation).

Table 3. Evaluated indices for consensus (von der Gracht, 2012)

Variable	LQ	UQ	IQR	Median positive	Mean	4-5%	Standard deviation (SD)	Coefficient of variation (CV)	APMO
P1	*4*	*5*	*1*	*100%*	*4,45*	*82%*	*0,82*	*0,16*	*81,82*
P2	*5*	*5*	*0*	*100%*	*4,73*	*91%*	*0,65*	*0,13*	*90,91*
P3	*4*	*5*	*1*	*100%*	*4,27*	*81%*	*0,79*	*0,20*	*81,82*
P4	*5*	*5*	*0*	*100%*	*4,82*	*100%*	*0,40*	*0,08*	*100,00*
P5	*4*	*5*	*1*	*100%*	*4,64*	*91%*	*0,67*	*0,13*	*90,91*
P6	*4*	*5*	*1*	*100%*	*4,73*	*100%*	*0,47*	*0,09*	*100,00*
P7	*5*	*5*	*0*	*100%*	*4,90*	*100%*	*0,32*	*0,06*	*100,00*
P9	*5*	*5*	*0*	*100%*	*4,91*	*100%*	*0,30*	*0,06*	*100,00*
P10	*5*	*5*	*0*	*100%*	*4,82*	*100%*	*0,40*	*0,08*	*100,00*
P11	*4*	*5*	*1*	*91%*	*4,27*	*82%*	*0,10*	*0,02*	*90,91*
P12	*3*	*5*	*2*	*91%*	*4,18*	*73%*	*1,08*	*0,22*	*81,82*
P13	*3*	*5*	*2*	*100%*	*4,36*	*73%*	*0,92*	*0,18*	*72,73*
P14	*3*	*5*	*2*	*91%*	*4,18*	*73%*	*1,08*	*0,22*	*81,82*
P15	*3*	*5*	*2*	*82%*	*4,09*	*73%*	*1,45*	*0,29*	*90,91*
P16	*4*	*5*	*1*	*91%*	*4,36*	*82%*	*1,03*	*0,21*	*90,91*
Reference values for consensus: LQ>=3; UQ=5; IQR<=1; Median Positive >= 80%; Mean ±5; 4-5% > 50%; SD ± 1,64; CV<=0,5; APMO >=69,7									

Source: Authors

Regarding propositions 13 and 14, some experts referred to the need for a means to exchange information (corresponding to the proposed definition of interface) in the form of APIs, database links, cloud servers, web servers, *etc.* The conception of an interface as a pre-established agreement was also the main concern of other experts. Such an agreement can be unilateral, as in the case of the aforementioned mechanisms (an analysis from a practical point of view could be necessary). In addition, some experts hesitated on the need for transformation rules, as systems sometimes have equivalent profiles and contexts, and interoperability can be directly established.

As for proposition 15, the expert concerns mainly revolve around the time when the preservation of the information should happen (before or after transformation).

Illustrative example

To provide an example, the case of a software development company dedicated to supporting the needs related to the technology and processes of SMEs was considered. This company seeks to develop software based on Microsoft

technologies (*i.e.*, Web, desktop, and mobile applications and service-oriented software). Their main customers are in the judicial, real estate, and CRM sectors.

Specifically, the example corresponds to the software systems of two companies in the commerce and financial sectors. The purpose of interoperability has to do with the target software system's need (the commerce company) for obtaining customer financial information. The source software system (the financial company) only provides such information to authorized partners. The target software system should commit to respecting the regulatory policies regarding information management, as well as the rules pertaining to the permanence of the information, among others.

AXIOM 1. INTEROPERABILITY DOMAIN

According to propositions 1 and 2, the preliminary information of the two organizations is summarized in Figure 6. Based on proposition 3, the duration of the collaboration is limited to the duration of the agreement between the two organizations. The frequency of interoperability is expected to be at least once a day. Additionally, the necessary domain terminology is provided in the form of a regulation document, as well as the API documentation of the source software system, according to proposition 4. This terminology is related to terms such as information holder, information source, and information operator, among others.

Enterprise 1		**Enterprise 2**	
Area	Commerce union	**Area**	Financial
Entity name	Reserved	**Entity name**	Reserved
Location	Medellín, Colombia	**Location**	Colombia
Activity	Commercial activity support	**Activity**	Managing the credit and financial information of individuals
Purpose	Consulting the credit and financial information of potential customers	**Purpose**	Exchanging credit and financial information
System involved	Credit assignment Software	**System involved**	ENTERPRISE 2 Web Application
Duration	Long term	**Duration**	long term
Responsibility	Not provided	**Responsibility**	Not provided
Constraint	---	**Constraint**	No public information available

Figure 6. Interoperability domain, case study
Source: Authors

AXIOM 2. INTEROPERABILITY PROFILE

According to proposition 5, the interoperability profile is filled with the available information, which is shown in Figure 7. However, the main requirements for interoperability are determined in a unilateral agreement using a web service provided by the source software system. Hence, it is not necessary to exchange some information.

Software system profile	**Name**: Credit Assignment Software	**Name**: ENTERPRISE 2 Web Application
Version	--	--
Programming language	Netbeans 8.2 JSP HTML, CSS	C# .Net framework®
Communication protocols	HTTPS	HTTPS
Operating system	Windows Server®	Windows Server®
DBMS	SQL Anywhere® SQL Server®	SQL Server®
Information exchange format	JSON	JSON
Platform	Linux, windows	Windows
Communication interfaces available	Apache Axis 2	REST APIs SOAP

Figure 7: Software interoperability profile, case study
Source: Authors

According to proposition 6, the software system profiles allow identifying a popular standard description for applications intended to consume and deploy web services. This kind of standardization facilitates communication between heterogeneous software systems. Based on the two software system profiles, interoperability between them might be possible. The method employed is more similar to the *unified* approach, according to proposition 7, as there is a neutral format known by both software systems, albeit not jointly agreed.

AXIOM 3. INTEROPERABILITY CONTEXT

According to proposition 8, the necessary details of the context for interoperating both software systems is conditioned by the web service provided, *i.e.*, the source software system only delivers the pre-determined information about the customers, as requested by the target software system, and it should be inferred according to the format of the exchanged JSON file. The source software context (Figure 8) is hidden from the target software system. According to proposition 10, the common context is defined on the target side during the integration of the exchanged information (*i.e.*, for creating and updating information of the customers).

AXIOM 4. INTEROPERABILITY LANGUAGE

According to proposition 11, the exchange language is JSON, which is recognized by both software systems. The source software system is programmed for exchanging a JSON file. The standard structure of this language is commonly recognized and involves context. Transforming the exchanged information is the responsibility of the target software system, according to proposition 12. This responsibility is implicitly defined in the interoperability agreement.

AXIOM 5. INTEROPERABILITY INTERFACE

According to proposition 13, the two organizations have agreed to establish an interface via the web service provided by the source software system. This system seeks to produce an identity certificate and a public key that provides access to partner companies. Settings should be accepted by the target software system in order to achieve communication, exchange, common understanding, and accessibility to the information, according to proposition 16. As stated in proposition 14, the transformations are applied according to the JSON structure, which maps the information in objects of the target software system's model.

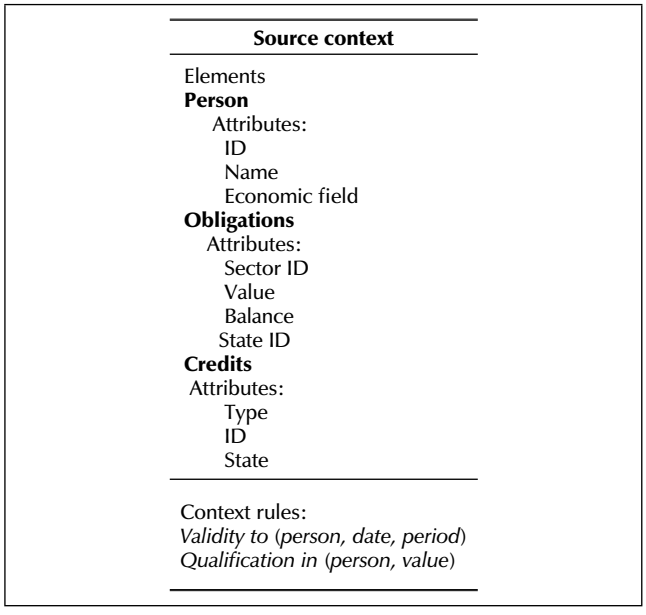

Figure 8. Source software context, case example.
Source: Authors

According to proposition 15, the target software system should respect the conditions contained in the source software system's documentation, given that financial information is sensitive to the precise moment in which the exchange happens, as well as to the interpretation given by the source software system. For example, the expiration period of the report is only seven days. In addition, a positive rating for a company or individual means that all obligations are up to date.

Discussion

The proposed explanation of what software interoperability is and how it works leads to recognizing the interaction between its seven essential elements. The lack of a unified terminology leads to different definitions of interoperability, depending on the approaches and type of interoperability addressed. A definition using the seven essential elements allows characterizing issues with a complete view of all the elements and actors involved. Moreover, a common characterization is useful to compare, discuss, and evaluate different approaches of interoperability.

The above-presented axioms are descriptions of what is needed for achieving interoperability. These descriptions can constitute a roadmap for addressing an interoperability project between organizations. This, with regard to (i) determining the necessities of interoperability (*i.e.*, intention, purpose, duration, and frequency); (ii) collecting information about software systems' interoperability features and profiles as a feasibility study; and (iii) implementing an approach to establish interoperability. The propositions are a reference for understanding the necessary arrangements made between software systems, which should be solved in order to achieve interoperability. The proposed axiomatic theory seeks to analyze and discuss approaches of interoperability and is useful for identifying issues in this regard, as well as their causes.

Conclusions and perspectives

This paper proposes an axiomatic theory (five axioms and sixteen propositions) of interoperability between heterogeneous software systems. This axiomatic theory is based on the seven essential elements of interoperability (and their relationships). An interoperability domain template and software system profiles are also proposed. Additionally, the notions of context and language, regarded as essential elements but lacking in the state of the art, are developed in their corresponding axioms and propositions.

As a result of this research, the proposed axiomatic theory allows explaining how interoperability happens and how it can be achieved. Consultation with some experts allowed validating the pertinence and consistency of the proposal. Finally, the application of the theory in a case study regarding a software development company showed the relevance and feasibility of the approach.

By using this example, it was determined that the relationships between essential elements are appropriately stated for identifying and describing interoperability issues and their solutions. Thus, the compliance of the propositions and the completeness and expressiveness of the axiomatic theory can be verified.

To conclude, the axiomatic theory presented herein allows (i) understanding what interoperability is and how it can be established, (ii) describing interoperability issues, and (iii) recognizing the source of said issues and addressing their solutions.

As future work, our axiomatic theory could be used for characterizing open issues of interoperability and identifying practices and activities intended to achieve it. Moreover, using this theory for creating measures and measurement methods of interoperability can provide the state of the art with added value. Finally, the axiomatic theory could be applied for collecting data and creating methods in different stages of software system development, such as planning,

analysis, implementation, deployment, and maintenance, seeking to predict, detect, and solve interoperability issues. All this, with the aim to advance the maturity of software system interoperability.

CRediT author statement

All authors: conceptualization, methodology, software, validation, formal analysis, investigation, writing (original draft, writing, review, and editing), data curation.

References

Burns, T., Cosgrove, J., and Doyle, F. (2019). A review of interoperability standards for industry 4.0. *Procedia Manufacturing*, *38*(1), 646-653. https://doi.org/10.1016/J.PROMFG.2020.01.083

Brilhault, Q., Yahia, E., and Roucoules, L. (2022). Qualitative analyses of semantic interoperability approaches: toward learning based model transformations. *IFAC-PapersOnLine*, *55*(10), 2348-2353. https://doi.org/10.1016/J.IFACOL.2022.10.059

Burzlaff, F., Wilken, N., Bartelt, C., and Stuckenschmidt, H. (2019). Semantic interoperability methods for smart service systems: A survey. *IEEE Transactions on Engineering Management*, *69*(6), 4052-4066. https://doi.org/10.1109/TEM.2019.2922103

Challita, S., Zalila, F., and Merle, P. (2018). *Specifying semantic Interoperability between heterogeneous cloud resources with the FCLOUDS formal language* [Conference presentation]. IEEE International Conference on Cloud Computing, San Francisco, CA, USA. https://doi.org/10.1109/CLOUD.2018.00053

Davies, J., Welch, J., Milward, D., and Harris, S. (2020). A formal, scalable approach to semantic interoperability. *Science of Computer Programming*, *192*(1), 102426. https://doi.org/10.1016/J.SCICO.2020.102426

Delgado, A., Calegari, D., González, L., Montarnal, A., and Benaben, F. (2020, January 7-10). *Towards a metamodel supporting e-government collaborative business processes management within a service-based interoperability platform* [Conference presentation]. 53rd Hawaii International Conference on System Sciences, Maui, HI, USA. https://doi.org/10.24251/HICSS.2020.246

Fraga, A. L., Vegetti, M., and Leone, H. P. (2020). Ontology-based solutions for interoperability among product lifecycle management systems: A systematic literature review. *Journal of Industrial Information Integration*, *20*(1), 100176. https://doi.org/10.1016/J.JII.2020.100176

Haile, N., and Altmann, J. (2018). Evaluating investments in portability and interoperability between software service platforms. *Future Generation Computer Systems*, *78*(1), 224-241. http://dx.doi.org/10.1016/j.future.2017.04.040

Horcas, J.M., Pinto, M., and Fuentes, L. (2023). A modular metamodel and refactoring rules to achieve software product line interoperability. *Journal of Systems and Software, 197*(1). https://doi.org/10.1016/J.JSS.2022.111579

International Organization for Standardization (2008). *Software engineering – Software product quality requirements and evaluation (SQuaRE) quality model (ISO/IEC 25010:2008)*. ISO.

International Organization for Standardization (2011). *Advanced automation technologies and their applications -- Requirements for establishing manufacturing enterprise process interoperability -- Part 1: Framework for enterprise interoperability (ISO 11354:2011)*. ISO.

Jepsen, S.C., Worm, T., Mork, T. I., and Hviid, J. (2021, Jun 3-3). *Industry 4.0 middleware software architecture interoperability analysis* [Conference presentation]. 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT), Madrid, Spain. https://doi.org/10.1109/SERP4IoT52556.2021.00012

Lafourcade, P., and Lombard-Platet, M. (2020). About blockchain interoperability. *Information Processing Letters*, *161*(1), 105976. https://doi.org/10.1016/J.IPL.2020.105976

Liu, L., Li, W., Aljohani, N. R., Lytras, M. D., Hassan, S. U., and Nawaz, R. (2020). A framework to evaluate the interoperability of information systems – Measuring the maturity of the business process alignment. *International Journal of Information Management*, *54*, 102153. https://doi.org/10.1016/J.IJINFOMGT.2020.102153

Mistry, P., Maguire, D., Chikwira, L., and Lindsay, T. (2022). *Interoperability is more than technology*. The King's Fund. https://www.kingsfund.org.uk/sites/default/files/2022-09/Interoperability%20is%20more%20than%20technology%20report.pdf

Ribeiro, E. L. F., Vieira, M. A., Claro, D. B., and Silva, N. (2018, March 19-21). *Transparent interoperability middleware between data and service cloud layers* [Conference presentation]. CLOSER, Funchal, Portugal. https://doi.org/10.5220/0006704101480157

Sana, K., Hassina, N., and Kadda, B. B. (2021, April 9-11). *Towards a reference architecture for interoperable clouds* [Conference presentation]. 2021 8th International Conference on Electrical and Electronics Engineering, Antalya, Turkey. https://doi.org/10.1109/ICEEE52452.2021.9415944

Serapião, G. Guédria, W., and Panetto, H. (2019). An ontology for interoperability assessment: A systemic approach. *Computer in Industry*, *16*(1), 100100. https://doi.org/10.1016/j.jii.2019.07.001

Sjøberg, I. K., Dybå T., Anda B. C. D., and Hannay, J. E. (2008). Building theories in software engineering. In F. Shull, J. Singer, and D. I. K. Sjøberg (Eds.), *Guide to Advanced Empirical Software Engineering* (pp. 312-336). Springer. https://doi.org/10.1007/978-1-84800-044-5_12

Software Engineering Group (2007). *Guidelines for performing systematic literature reviews in software engineering (EBSE-2007-001)*. School of Computer Science and Mathematics, Keele University. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Tall, D. (2004). Building theories: The three worlds of mathematics. *For the Learning of Mathematics*, *24*(1), 29-32.

Torab-Miandoab, A., Samad-Soltani, T., Jodati, A., and Rezaei-Hachesu, P. (2023). Interoperability of heterogeneous health information systems: a systematic literature review. *BMC Medical Informatics and Decision Making*, *23*, 18. https://doi.org/10.1186/s12911-023-02115-5

Torres, D. M., Villavicencio, M. K., and Zapata, C. M. (2022). Representing interoperability between software systems by using pre-conceptual schemas. *International Journal on Electrical Engineering and Informatics, 14*(1), 101-127. https://doi.org/10.15676/ijeei.2022.14.1.7

Torres, D. M., Villavicencio, M. K., and Zapata, C. M. (2018, August 29-31). *Towards a terminology unification in software interoperability* [Conference presentation]. 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Prague, Czech Republic. https://doi.org/10.1109/SEAA.2018.00083

Turk, Z. (2020). Interoperability in construction – Mission impossible? *Developments in the Built Environment*, *4*(1), 100018. https://doi.org/10.1016/j.dibe.2020.100018

von der Gracht, H. A. (2012). Consensus measurement in Delphi studies: Review and implications for future quality assurance. *Technological Forecasting and Social Change*, *79*(8),1525-1536. https://doi.org/10.1016/j.techfore.2012.04.013