Interoperability-Related Architectural Problems and Solutions in Information Systems: A Scoping Study

Conference Paper *in* Lecture Notes in Computer Science · August 2014

DOI: 10.1007/978-3-319-09970-5_27

3 authors:

Hadil Abukwaik
ABB

Davide Taibi
Tampere University

Dieter Rombach
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau

Interoperability-Related Architectural Problems and Solutions in Information Systems: A Scoping Study

Hadil Abukwaik, Davide Taibi, and Dieter Rombach

University of Kaiserslautern
Gottlieb-Daimler-Straße 47
67663 Kaiserslautern, Germany
{abukwaik,taibi,rombach}@cs.uni-kl.de

Abstract. [*Context*] With the increasing industrial demands for seamless exchange of data and services among information systems, architectural solutions are a promising research direction which supports high levels of interoperability at early development stages. [*Objectives*] This research aims at identifying the architectural problems and before-release solutions of interoperability on its different levels in information systems, and exploring the interoperability metrics and research methods used to evaluate identified solutions. [*Methods*] We performed a scoping study in five digital libraries and descriptively analyzed the results of the selected studies. [*Results*] From the 22 studies included, we extracted a number of architectural interoperability problems on the technical, syntactical, semantic, and pragmatic levels. Many problems are caused by systems' heterogeneity on data representation, meaning or context. The identified solutions include standards, ontologies, wrappers, or mediators. Evaluation methods to validate solutions mostly included toy examples rather than empirical studies. [*Conclusions*] Progress has been made in the software architecture research area to solve interoperability problems. Nevertheless, more researches need to be spent on solutions for the higher levels of interoperability accompanied with proper empirical evaluation for their effectiveness and usefulness.

Keywords: Software interoperability, software architecture, information systems, scoping study.

1 Introduction

Interoperability among software systems endows them with the capability to meaningfully communicate and exchange information and services [1, 2]. However, interoperability faces many challenges, e.g., different communication protocols, incompatible architectures, heterogeneous data models, ambiguous meaning of information exchanged, and more. In response, several solution approaches have been proposed. On one hand, integration solutions that focus on solving interoperability problems after they happen are the most suggested ones. However, adopting any of these integration solutions to overcome systems' heterogeneity is expensive and requires significant effort [3]. On the other hand, before-release architectural solutions are proposed to build interoperability potentials in software systems with reduced cost. These architectural design

P. Avgeriou and U. Zdun (Eds.): ECSA 2014, LNCS 8627, pp. 308–323, 2014.

decisions have an immediate impact on systems' components and connectors that can be the main obstacle impeding interoperability [4]. Such a promise from the architectural solutions makes them a powerful base for interoperability and hence they are the main interest of this paper.

With the increasing complexity of information systems (ISs) and their interoperability requirements, software architects need to choose from existing solutions that support before-release interoperability. However, this task becomes a challenge with the proliferated architectural solutions that are scattered across research fields [5] (such as component-based software, open systems, enterprise application, etc.) with focus on multiple interoperability issues (such as syntax, structure, semantics, etc.). Also, having no evaluation results for proposed solutions is a significant issue which questions their effectiveness and real value gained when adopting these solutions [6].

In the light of the big magnitude and high business value of interoperability among ISs [7], it is important to alleviate the aforementioned task complexity and to support software architects in choosing appropriate interoperability solutions. Hence, in this research we performed a systematic scoping study in order to (1) identify the state-of-the-art of interoperability architectural problems and before-release solutions in ISs, and to (2) explore the state of evidence on the quality of the identified solutions. This study helps practitioners to understand the state of research on interoperability-related architectural approaches and to consider adopting them. Also, the findings provide researchers with insights regarding future research topics to cover the identified gaps.

The rest of this paper is structured as follows. Section 2 introduces a background, Section 3 overviews related work to our study and Section 4 outlines the design of the scoping study. Section 5 reports the results and Section 6 discusses their implications. While Section 7 presents the study limitations, Section 8 summarizes the conclusions.

2 Interoperability Levels - Background

Multiple classification models have been built for defining and organizing interoperability levels in software systems. These models help in defining the compatibility level between systems and the amount of effort required to enable them to work jointly. Examples of interoperability models include: (1) the Levels of ISs Interoperability (LISI) [8], (2) NC3TA Reference Model for Interoperability (NMI) [9] and (3) the Levels of Conceptual Interoperability Model (LCIM) [10]. Whereas the LISI and NMI focus on the technical level of interoperability, LCIM provides a more comprehensive classification from data sharing capabilities point of view which we see it as an essential goal of interoperation in ISs. Therefore, we select LCIM to be our reference model for interoperability levels in this study.

LCIM encompasses seven levels which increase from no interoperability level to conceptual interoperability level. Here we present a brief description for each of the seven levels of the LCIM model: (1) **No Interoperability**: no connection or data sharing with other systems. (2) **Technical Interoperability**: physical connection and data exchange with other systems. (3) **Syntactic Interoperability**; similar structure for information exchange and unambiguous data formats. (4) **Semantic Interoperability**:

shared reference model for information exchange and clear data meanings. (5) **Pragmatic interoperability**: methods and procedures used by participating systems are known by the others. Besides, understanding the context of the exchanged information and how they are used is unambiguous. (6) **Dynamic Interoperability**: changing data and operations over time in a participating component are comprehended by other components. The effect of exchanging information is explicitly announced. (7)**Conceptual Interoperability**: concepts and assumptions that components of the domain operate on are aligned. This requires documenting conceptual models by engineering methods allowing engineers to interpret and evaluate them.

3 Related Work

Sedek et al. [11] have systematically reviewed the current architecture-based approaches used for building interoperability in e-government portal until 2011. Sedek et al. reviewed previous works to identify a suitable approach for creating architectures with higher interoperability. They identified 17 studies and analyzed them with respect to: important characteristics of architectural aspect of e-government portal, the interoperability and reliability achievements of the current e-government portal architecture and the common limitations and strengths of the existing e-government architectures. Sedek's study concluded that current approaches lack improving the architecture towards a high level of interoperability and reliability. They stated that SOA and layered architectures are common in e-government portals. Also, they found that mediators are incorporated in architectures to resolve technical and semantic mismatches using approaches like Semantic Mediator Model and User Ontology.

To the best of our knowledge, the previously mentioned study is the only related work to this scoping study. Our research extends the work of Sedek et al. by: (1) reviewing both architectural problems and solutions of interoperability on different interoperability levels and (2) considering all types of ISs from different application domains rather than focusing only on enterprise systems from the e-government domain. These extensions broaden the scope of the research along with its collected data and strengthen the validity of the conclusions we build regarding the ISs interoperability. Moreover, we aim at exploring the evidence on the identified solutions' quality by looking for used evaluation method which supports or rejects their claims.

4 Research Methodology

In this research we systematically study the nature and extent of software architecture researches about interoperability problems and before-release solutions in ISs, to collate, summarize and disseminate research findings, and to identify research gaps. Therefore, we performed a scoping study following the process proposed by Petersen et al. [12] along with a data extraction form. Different than systematic literature review [13], we aimed at a broad analysis for literature rather than an in-depth analysis and quality assessment for selected papers. All materials of this study are available at the scoping study webpage[1].

4.1 Research Questions

The goal of this scoping study is to identify architectural problems and before-release solutions of interoperability in the context of ISs from the view point of researchers and software engineers. This goal is translated into the following research questions:

- *RQ1: Which levels of interoperability are handled in literature with architectural solutions?* This question intends to determine the extent to which architecture research addresses interoperability in terms of the levels of LCIM model.
- *RQ2: What are the architectural problems faced when building interoperability among ISs*? This question intends to identify the issues and key drivers that need to be considered while designing ISs to support the desired interoperability property.
- *RQ3: What are the architectural solutions for handling the identified problems?* This question intends to identify the architectural design decisions and activities proposed in literature to handle the identified interoperability issues.
- *RQ4: How are architectural solutions for interoperability evaluated?* This question intends to explore the evidence provided on the quality of identified solutions in terms of the used evaluation method.
- *RQ4.1: What interoperability measures are used to evaluate the architectural solutions*? This question intends to investigate interoperability metrics used as a part of the evaluation.

4.2 Data Sources and Search Strategy

According to the recommendations of Dybå et al. [6], we looked for published papers in journals and conference proceedings of the following databases: IEEE Xplore, ACM Digital Library, Springer Digital Library, Google Scholar, and Science Direct. Having the data sources selected, we performed trial searches using various combination of search terms derived from our research questions. Based on the results we defined our search terms as: (T1) Interoperability AND Architecture, (T2) Interoperation AND Architecture, (T3) Interoperability AND Architectural Design, and (T4) Interoperation AND Architectural Design. The search process was carried as follows:

- **Stage 1**: Pilot search the databases using the defined terms T1 to T4 separately and then combined with the "OR" operation to remove duplicates. The search was applied on the titles and abstracts (4128 studies).
- **Stage 2**: As abstracts from stage 1 showed irrelevance to the research questions, the database search was refined to be applied on titles only (246 studies).

[1] `http://wwwagse.informatik.uni-kl.de/staff/abukwaik/pub/ECSA14/scoping-study.htm`

- **Stage 3**: Inclusion/exclusion criteria, described in subsection 4.3, were applied on the 246 studies based on keywords, abstracts, and conclusions (22 studies).

We note that the primary studies included in Sedek's review were not a subset of our study. This was beacause the titles of these studies focused on eGoverment rather than interoperability and architecture. Consequently they were not retrieved in stage 2.

4.3 Inclusion and Exclusion Criteria

A study got included if it met all the inclusion criteria and none of the exclusion criteria, otherwise it got excluded. Inclusion criteria are:

- I1. Studies with a main focus on interoperability problems and architectural solutions in ISs.
- I2. Studies with architectural solutions supporting interoperability before release.

While exclusion criteria are:

- E1. Studies with writing language other than English.
- E2. Gray studies with unclear peer-review process (e.g., technical reports, short papers, keynotes, abstracts, etc.).
- E3. Secondary studies about interoperability problems and solutions (i.e., related works to this research).
- E4. Studies with minor interest in interoperability architectural aspects.
- E5. Studies proposing solutions for specific projects under restricted settings and conditions that cannot be generalized to ISs

Separately, two researchers applied the criteria on the studies and in discussion sessions, decisions about discrepant results were taken based on reached consensus. The search was conducted in November 2013, and had no timeframe limitations to get a broader coverage of studies related to our research questions. Note that we did not contact authors of included studies seeking unpublished evaluation or other related researches.

4.4 Data Extraction Strategy

Table 1 shows the fields that correspond to our predefined research questions. One researcher extracted the data from the 22 included studies and another checked it against the studies to ensure completeness and correctness of the extraction process.

4.5 Data Analysis

Qualitative data analysis was performed using an initial coding schema in a tabular form including interoperability problems, interoperability levels, architectural solutions, architectural components, and evaluation types. The coding schema provided definition of concepts, categories, and criteria that guided the translation of raw data into descriptions that answer the research questions.

Table 1. Data extraction form

Field		Description	RQ
F1	Title	Title of the paper	Documen-tation
F2	Author	Writer(s) of the paper	
F3	Year	Year of publishing the paper	
F4	Publication	Name of Journal / Proceeding	
F5	Keywords	Keywords of the paper	RQ1
F6	Objectives	Stated goals of the study by the authors- free text	RQ1
F7	IS type	Kind of IS application which the study focuses on	RQ2
F8	Interoperability problem(s)	Object of the study which the study tries to solve (i.e., problem of interest) - free text	RQ2
F9	Interoperability level	Level of LCIM that the study handles (see section 2)	RQ1
F10	Architectural solution(s)	Subject of the study that is proposed to solve the object (i.e., solution of problem) - free text	RQ3
F11	Solution elements	Concrete elements of the proposed subject (i.e., com-ponents of architectural solutions) - free text	RQ3
F12	Technology used	Technologies supporting implementation of proposed subjects (e.g., XML, Web Services … etc.)	RQ3
F13	Solution evidence	Evidence provided on the quality of proposed subjects (e.g., discussion, controlled experiment, case study, etc.)	RQ4
F14	Interoperability Metric	Quantitative measures used in the study evaluation to describe the interoperability property achieved	RQ4.1
F15	Comments	Additional notes provided in the study (i.e., claimed benefits, tradeoffs, limitations, or challenges) - free text	RQ2.1

5 Results

5.1 Overview

The identified primary studies were 22 that were performed in diverse application domains (e.g., eGovernment, eCommerce, eLearning, geographical, military, and biomedical systems). As seen in Fig. 1, there is a little increase in the number of studies on interoperability after 2004.

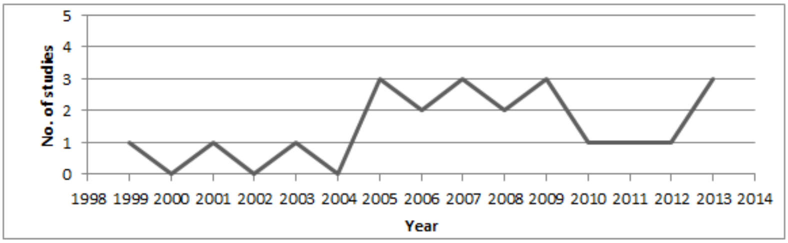


Fig. 1. Year-wise distribution of selected studies

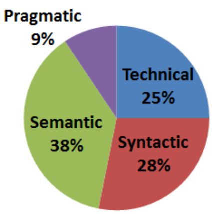
Studies were conducted in academic and industrial environments with 10 of 22, 45%, collaboration between the two. Almost all studies (21 of 22, 95%) were published in conferences, while one study appeared in a journal. Remarkably, there is no a dominating venue publishing many studies on interoperability architectural problems and solutions, i.e., each venue published one study except for one which published two studies. Also, one conference found dedicated to software interoperability named “Distributed Applications and Interoperable Systems”.

5.2 Interoperability Architectural Problems and Solutions

RQ1: Which levels of interoperability are handled in literature with architectural solutions?

To determine the interoperability concerns of each study, we analyzed its keywords F5, objectives F6, problem description F8, and solution advantages F15. Afterwards, we compared these concerns to the description of LCIM levels.

Figure 2 illustrates the distribution of the handled levels of interoperability over the included studies. Some studies addressed more than one level, e.g., S3 addressing both the semantic and pragmatic levels. Note that, semantic has the biggest share of the studies’ focus with a growing interest along the years while the pragmatic has a low share and disappeared after 2007. Syntactic and technical levels have convergent shares. In recent years, especially 2012 and 2013, the technical level grasps the attention of the inter-Cloud systems researchers (S18 and S22). Both the dynamic and conceptual levels got no share in the studies at all.



Interoperability level	Studies' ID
Technical	S4, S8, S9, S16, S17, S18, S20, S22
Syntactic	S1, S3, S4, S6, S9, S15, S16, S20, S21
Semantic	S2, S3, S5, S10, S11, S12, S13, S14, S16, S19, S20, S21
Pragmatic	S3, S6, S11

Fig. 2. Interoperability-level distribution over selected studies

RQ2: What are the architectural problems faced when building interoperability among ISs?

For each study, we examined the interoperability problem it addresses from the problem description F8. Then, we mapped each problem to the corresponding level of LCIM which shares and includes its concerns. Synthesizing the problems of all studies, we identified eight distinct architectural issues where seven of them related to LCIM levels as seen in Table 2.

P1: Semantic heterogeneity of data is the most common problem (occurrence number (N) = 11). It concerns architects about designing interoperable systems that correctly interpret the meaning of data elements being exchanged among them. For example, the authors of (S11) investigated designing interoperability among different

GIS systems, and stated that it was a challenge due to the growing number of heterogeneous spatial data sources with semantic differences.

P2: Syntactical heterogeneity of data has been reported frequently (N = 7). It requires architects to take into account the differences in data types, formats, and modeling languages of interoperating systems. For instance, in (S6), Carvalho et al. stated that exchanging geographic data among different layers on GIS required resolving its different representations first.

P3: Heterogeneity of communication protocols, platforms, and technical standards are considered as serious architectural problems (N = 7). It is essential for interoperability to make design decisions that enable the system to establish communication with systems of different technical properties. In (S9), Rabhi observed that developing cooperation among financial market systems required enormous effort due to their variant technologies, communication interfaces, and network protocols.

P4: Heterogeneity of data context has been reported as a problem in the context of financial and GIS systems (N=3). It is important for architects to reflect the context in which the designed system functionalities and data can be used to assure meaningful interoperability. For example, (S11) described possible context heterogeneity to happen in interpreting a domain value of a CropType attribute in the designed system. While in one land it could be "Wheat", in the other it could be "Corn".

Other stated problems include: **P5: Heterogeneity of method signatures; P6: Misunderstanding of the sematic interoperability meaning**; **P7: Redundancy of data**; and **P8: Inadequacy of architecture framework supporting interoperability**.

Table 2. Overview LCIM levels with their identified problems and solutions in the studies

Interoperability Level	Problem ID	Solution ID	Study ID
Technical	P3	Sol5	S4
		Sol7	S9, S16, S17, S18, S20, S22
		Sol10	S8
Syntactical	P2	Sol5	S6
		Sol7	S15, S16, S20
		Sol8	S3
		Sol9	S9
	P5	Sol6	S4
	P7	Sol13	S1
Semantic	P1	Sol1	S14, S21
		Sol2	S3, S5, S10, S12, S13
		Sol4	S11
		Sol3	S16, S19, S20
	P6	Sol11	S2
Pragmatic	P4	Sol2	S3, S11
		Sol5	S6
n/a	P8	Sol12	S7

RQ3: What are the architectural solutions for handling the identified problems?
For each study we studied the interoperability solution it proposed from the architectural solution F10, its components F11, and the used technology F12. Then we mapped the solutions to the identified problems in RQ2 (see Table 2).

Sol1: Standards address semantic interoperability problems, e.g., (S21) unambiguous semantic metadata is achieved through a standard-based metadata repository which provided formal description for the meaning of data types used in classes and attributes of data systems. Also, (S14) proposed standard-based modeling for processes and data between collaborating organizations.

Sol2: Ontologies solve semantic and context interoperability problems. For example, (S13) proposed ontology-based blackboard architecture to facilitate user retrieval for the correct service offered by eGovernment system based on his needs with less effort. This was by modeling the basic concepts of services from a user perspective.

Sol3: Semantic mediator aligns semantically related concepts. We identified three identified forms of mediators: *formal-methods-based mediator* aligns the behavior of systems using their LTS models (S16), *thesaurus-based mediator* mediates concepts using knowledge structures simpler than ontologies (S19), and *standard-based mediator* facilitates standardized information exchange and orchestration (S20).

Sol4: Wrapper encapsulates local data sources in export schema comprising the main concepts of the real world entities. As described in (S11), a wrapper receives queries from interoperating systems and translates them into a local form to enable processing them and to retrieve the required information from the local system.

Sol5: Adaptor The adaptor embeds the connection state and logic to one or more external systems, e.g., it can encapsulate a telnet-based connection to a remote Unix host (S4). Also, (S6) proposed using adaptor component to transform data among interfaces of different GIS devices.

Sol6: Facets provides different implementations for a standard interface of an action. Hence, the action can be invoked by different system types through its corresponding facet. In (S4), these facets are automatically generated by specialized tools.

Sol7: Middleware handles heterogeneities in communication protocols and data formats. In (S16), Bennaceur et al. presented how on-the-fly middleware component dynamically resolved heterogeneity of data formats in messages being exchanged between distributed systems.

Sol8: External data models are concerned with representing all sources of data that the system may exchange with other interoperating systems. In (S3), the authors gave examples on external data to include relational database sources, XML sources, HTML web wrapper sources, and computational procedures modeled as relations.

Sol9: Internet data formats are proposed to be used on the data level of distributed systems to ensure wide applicability of the associated components (S9), i.e., using XML and its variants like FIXML with CORBA for handling the communication.

Sol10: Technical reference model guides in expeditiously selecting technical standards using common vocabulary. According to (S8), this fosters interoperability by providing appropriate system standard profiles.

Sol11: Semantic reference model guides developing semantic interoperability capabilities in systems by fulfilling a set of semantic requirements. In (S2), these requirements are categorized as policy and governance, organization, and technology.

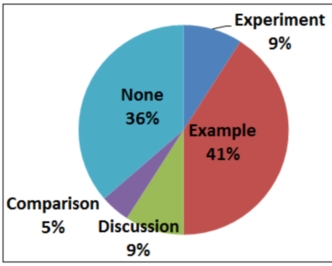
Sol12: Enterprise architecture framework provides a systematic blueprint to build interoperability among enterprise IS. In (S7), the identified framework resolves weakness es comparatively determined in legacy enterprise architecture frameworks.

Sol13: Central repository allows cooperative sharing of information among systems. For example, (S1) proposed using a central repository for installed applications on a phone device to enable sharing resources and context data among them.

A recurring theme we observe in the findings is basing the identified solutions on the service oriented architecture style (SOA), and implementing it with the web service technology. This theme was reported in nine studies (S5, S6, S10, S12, S13, S14, S17, S18, and S22). Also, we found that the different solutions are not associated with particular application domain or research field, i.e., they are applicable in general ISs.

5.3 Evidence on the Quality of the Identified Solutions

RQ4: How are architectural solutions for interoperability evaluated?
As seen in Fig. 3, 8 out of the 22 identified studies did not provide any evaluation of their proposed solutions. Because of the lack of empirical evidence regarding the quality of the identified solutions, it was not possible to determine their effectiveness.



Evaluation method	Studies' ID
Experiment	S1, S16
Example	S2, S3, S11, S13, S17, S18, S19, S20, S21
Discussion	S4, S22
Comparison	S7
None	S5, S6, S8, S9, S10, S12, S14, S15

Fig. 3. Evaluation-method distribution over selected studies

RQ4.1: What interoperability measures are used to evaluate the architectural solutions?
None of the studies included in this scoping study used interoperability metrics to appraise achieving it in the systems.

Studies with empirical evaluation focused only on assessing: *performance* in terms of query execution time (S1), *feasibility* in terms of concepts' understandability and development easiness (S7), and *validity* in terms of overcoming the interaction and application heterogeneity (S16). Noteworthy, neither (S7) nor (S16) was accompanied with quantitative data.

Studies with toy examples described their solutions' benefit against different interests: (S2) argued providing a good base for evaluating the maturity level of semantic interoperability capability of agencies; (S3) showed allowing context mediation without rigidity imposed by changing original context models; (S13) explained how end-users were provided with appropriate interfaces for published services; (S17) illustrated how groupware requirements diversity could be more easily fulfilled by controlling concurrency access to shared documents; (S19) clarified the feasibility of achieving semantic interoperability with simpler structures rather than ontologies; (S20) explained gained adaptivity, flexibility, and security; (S21) presented the feasibility to make semantically interoperable data using ontologies and standards.

Studies with no evidence claimed to achieve autonomy, flexibility, and extensibility (S11) and to allow optimized provisioning of computing, storage, and networking resources (S18). No reflection of such claims was found in the given examples.

6 Discussion

The study results reveal that software interoperability architectural problems and solutions have been studied especially on the syntactic and semantic levels over the last fifteen years. However, only a few studies proposing solutions to the higher LCIM levels have been published. Also, results demonstrate the low evidence level of the studies as the quality of their solutions was not properly evaluated in the included papers of our scoping study. Consequently, we want to draw the attention to the following issues that should be overcome to advance the research area:

Architectural basis for Higher Levels of Interoperability. This scoping study exhibits that research efforts have not addressed the dynamic and conceptual levels of interoperability yet. In fact, standalone architectural solutions are not adequate by themselves to comprehensively solve the aforementioned high levels. That is, a broader interdisciplinary view is needed, which involves organizational, managerial, and advanced technical decisions, e.g. using artificial intelligence methods and technologies. Accomplishing this interdisciplinary solution effectively needs the support of a mature architectural basis. For example, unaligned models of business processes would be better handled if constraints ambiguity of dynamically exchanged business data had already been handled using mature architectural solutions.

Accordingly, we emphasize on the importance to reach a reasonable degree of architectural maturity in backing interoperability on its higher levels. As indicated by [14] , achieving a clear interoperability maturity level determines systems' strengths and weaknesses in terms of their likelihood to interoperate; and hence defines the improvement priorities towards successful interoperability.

Prior Architectural Solutions to Support Interoperability Before Release. The results show that researchers tend to deal with interoperability problems after facing them, i.e., expensive posterior solutions [3]. Contrary, adopting prior architectural solutions can save time and effort, e.g., designing and implementing an interface adaptor for under construction system is less expensive than modifying a released system and integrating it with new components [3]. Therefore, we call for pushing the wheels of research in the direction of prior architectural solutions for interoperability.

Architectural Practices to Support Software Interoperability. In this study, only architectural design decisions have been found as architectural. However, software architecture includes other activities that affect systems characteristics like architectural analysis, synthesis, evaluation, and documentation [15]. It is thus of significant importance to direct such activities towards improving interoperability potentials of ISs and facilitating its tasks. For instance, it would be useful to have studies about best practices to evaluate design patterns with regards to interoperability. Also, studies about architecture documentation activities that introduce specialized interoperability views can help in analysis phases. Hence, researches on architectural activities supporting interoperability are required and should reserve a place in future studies.

Empirical Evidence on the Quality of Proposed Solutions. Based on our collected data, the majority of the identified architectural solutions have not been associated with reliable validation. This can lead to difficulties for practitioners to properly adopt interoperability solutions and to systematically enhance them in future works. Thus, it is important to provide trustworthy evidence like empirical evaluations to raise the reliability of a solution and encourage adopting it. Such evaluation should analyze a solution with respect to its achieved interoperability level, costs, and any other claimed benefits. The experience reported in the field of evidence-based software engineering explains the necessity of empirical evaluation to enable fast adoption of good practices, improve products' quality and minimize projects' failures [6].

Comparisons among Interoperability Architectural Solutions. The results show that the identified interoperability architectural solutions have not been compared to the already existing ones in literature. This is absolutely acceptable if solutions aim at solving interoperability problems that have not been addressed before. However, a proper justification on the preference of adopting a new solution over others addressing the same problem would be needed. Specifically, we call for comparing the experimental results of new solutions with results obtained from previous ones. Similar recommendation has been proposed by Aleti et al. [16] in the context of building new software architecture optimization methods. Moreover, it would be of additional help if trade-offs of solutions are declared too. In this sense, the community would benefit also from publicly sharing evaluation results to enable conducting comparisons.

Interoperability Metrics for Assessing Solutions. The included studies are inconsistent in estimating the benefits of their solutions, i.e., they differ in both the qualities they assessed and the metrics they used. This lack of consistency impedes comparing the solutions and thus we could not infer the architectural characteristics that influence the interoperability property of systems. Another issue is that, some studies measured interoperability using indirect metrics that have unclear relation to interoperability, e.g., autonomy, resource provisioning, security, and concurrency.

Hence, reporting bias represented in both inconsistency and indirectness should be overcome through using valid and reliable measures of interoperability, These measures include interoperability models like: the Levels of Information Systems Interoperability (LISI) model [8], the Operational Interoperability Model (OIM) [17], the LCIM [10], the System of Systems Interoperability (SOSI) model [18], and others. Using these interoperability models can be a good base for reporting the results of the previously discussed empirical evidence and comparisons on the quality of interoperability solutions. Though, it would be of greater benefit to come up with metrics that can precisely quantify systems' interoperability and clearly draw the lines between semantic, pragmatic, and conceptual levels.

Combining interoperability solutions' empirical evaluation, consistency in reporting results, and directness in assessing interoperability, we can definitely improve the strength of evidence of these solutions. Thus, estimating effectiveness and interoperability achieved when adopting these solutions can be more certain and trustworthy.

Reference Rules for Selecting Appropriate Interoperability Architectural Solutions. Currently, various interoperability architectural solutions have been identified and some are addressing similar problems. Therefore, it is important to provide guiding rules that define interoperability problems and assign them to their most suitable architectural solutions. For example, it would be a valuable assistance for junior interoperability architects facing a semantic data heterogeneity problem to have precise directions on how to choose from alternative solutions like ontology-based, standards-based and thesaurus-based mediations. Certainly, designers of such rules need to carefully take into account the different factors that may influence the effectiveness of adopting a specific solution. These factors include available resources, system components' modularity and dependency, targeted interoperability level, system domain, project size, developers' experience, etc.

Tool Support for Interoperability. Another useful support for practitioners designing and building interoperability would be to aid them with software tools that can automatically identify potential interoperability problems between two systems from their architectural models. More helpful these tools can be, if they can also suggest plausible architectural solutions for the detected problems using the aforesaid guidelines. For example, this can be implemented as a plug-in, to an existing software architecture modeling language (e.g., UML), which provides an interoperability view, reports architectural mismatches, and supports resolving these mismatches.

7 Limitation of This Study

Researcher Bias. (1) To produce unbiased conducting for the study, the selection criteria and data extraction protocol were derived from the research questions and reviewed by an independent researcher. For the same purpose, the study selection was performed by two researchers. (2) To ensure correct inference in extracting data from studies with poor or insufficient description, data extraction was performed by one researcher and reviewed by another with discussions as needed. (3) To increase the confidence about the outcome of interpreting the qualitative data, analysis results were reviewed and discussed until agreement among the researchers. This was important in cases where interoperability was described using different or no models. (4) For a transparent and replicable study, data and results of each step were documented.

Publication Bias. Although we did the search in large electronic databases, we did not contact authors to identify unpublished evaluation or other related researches. Also, even with deriving the search terms from the research questions, software engineering keywords are not standardized. Consequently, relevant studies might be missed due to our search terms choice. For these reasons, we do not claim generalizing the results for the whole research field. However, this research covered a significant part of the literature and provided valid results.

8 Conclusion

We have performed a scoping study to identify the architectural problems and solutions for interoperability in ISs. Also, we pursued evidence on the identified solutions' benefits in the selected 22 studies. The studies were published between 1999 and 2013.

Our study contributes by listing faced interoperability problems in IS and mapped them to the identified solutions. The study results reveal that while the technical, syntactical, semantic, and pragmatic interoperability problems are addressed, dynamic and conceptual ones still in need for research attention. The identified architectural solutions vary to include ontologies for semantic issues, adaptors for syntactical differences, middlewares for technical variations, and mediators for pragmatic problems.

Although most of the included studies justify their solutions using examples, many did not present any evaluation method. Besides, no direct interoperability metric is used to appraise the sought out interoperability.

In order to advance the software architecture research towards being a cornerstone in achieving interoperability, we conclude the necessity for further research to: (1) address interoperability on its higher levels, (2) provide empirical evidence for solutions using reliable interoperability metrics, and (3) support interoperability architects and developers with reference rules and tools. Findings also indicate a need to raise the recognition of the interoperability topic within software architecture venues.

Acknowledgments. This research was performed as part of the PhD research of Hadil Abukwaik under the supervision of Prof. Dieter Rombach. We thank Mohammed Abufouda for replicating the selection of studies and the data extraction. Our thanks also go for Liliana Guzmán, Dr. Matthias Galster, Dr. Matthias Naab, and the anonymous reviewers for their valuable comments and guidance.

A Appendix: Selected Studies

ID	Reference
S1	A. Brodt et al.: A mobile data management architecture for interoperability of resource and context data. In MDM (2011)
S2	A. Ojo et al.: Semantic interoperability architecture for electronic government. In dg.o (2009)
S3	A. Moulton et al.: Semantic Interoperability in the fixed income securities industry: A knowledge representation architecture for dynamic integration of web-based information. In HICSS (2003)
S4	G. Hatzisymeon et al.: An architecture for implementing application interoperation with heterogeneous systems. In DAIS (2005)

S5	L. Xianming et al.: Research on the Portlet Semantic Interoperability Architecture. In WCSE (2009)
S6	D. de Carvalho et al.: Functional and device interoperability in an architectural model of geographic information system. In SIGDOC (2007)
S7	J. Kim et al.: An enterprise architecture framework based on a common information technology domain (EAFIT) for improving interoperability among heterogeneous information systems. In SERA (2005)
S8	S. Zhu et al.: Army enterprise architecture technical reference model for system interoperability. In MILCOM (2009)
S9	F. Rabhi: Towards an open architecture for the integration and interoperability of distributed systems. In Ent-Net at SUPERCOMM (2001)
S10	B. Powers: A multi-agent architecture for NATO network enabled capabilities: enabling semantic interoperability in dynamic environments (NC3A RD-2376). In SOCASE (2008)
S11	E. Leclercq et al.: ISIS: a semantic mediation model and an agent based architecture for GIS interoperability. In IDEAS (1999)
S12	M. Paul: Enterprise geographic information system (E-GIS): A service-based architecture for geo-spatial data interoperability. In IGARSS (2006)
S13	G. Lepouras et al.: An active ontology-based blackboard architecture for web service interoperability. In ICSSSM (2005)
S14	C. Schroth et al.: UN/CEFACT Service-Oriented Architecture-Enabling Both Semantic And Application Interoperability. In KiVS (2007)
S15	P. Arapi et al.: ASIDE: An Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications. In ICALT (2006)
S16	A. Bennaceur et al.: Towards an architecture for runtime interoperability. In ISoLA (2010)
S17	R. Maciel et al.: WGWSOA: A service-oriented middleware architecture to support groupware interoperability. In CSCWD (2007)
S18	Y. Demchenko et al.: Intercloud Architecture for interoperability and integration. In CloudCom (2012)
S19	D. Arize et al.: ThesIS: A semantic interoperability service for a middleware service oriented architecture. In CSCWD (2013)
S20	R. Crichton et al.: An Architecture and Reference Implementation of an Open Health Information Mediator: Enabling Interoperability in the Rwandan Health Information Exchange. In FHIES (2013)
S21	G. Komatsoulis et al.: caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. In J-BHI (2008)
S22	A. Mohtasebi et al.: Analysis of Applying Enterprise Service Bus Architecture as a Cloud Interoperability and Resource Sharing Platform. In KMO (2013)

References

1. IEEE Standard Computer Dictionary, IEEE Std 610. A compilation of IEEE standard computer glossaries (1991)
2. ISO/IEC 2382-1: Information technology, vocabulary, Part 1: Fundamental terms (1993)
3. Gonçalves, R., Müller, J., Mertins, K.: Enterprise Interoperability III: New Challenges and Approaches. Springer (2007)
4. Davis, L., Payton, J., Gamble, R.: How system architectures impede interoperability. In: Proceedings of the 2nd International Workshop on Software and Performance, pp. 145–146 (2000)
5. Land, C.I.: Existing approaches to software integration–and a challenge for the future. Integration 40, 58–104 (2004)

6. Dybå, T., Kitchenham, B., Jørgensen, M.: Evidence-based software engineering for practitioners. IEEE Software 22(1), 58–65 (2005)
7. Loukis, , Charalabidis, : An empirical investigation of information systems interoperability business value in European firms. Computers in Industry 64(4), 412-420 (2013)
8. C4ISR Interoperability Workig Group: Levels of information systems interoperability (lISI). Technical Report, Department of Defense (1998)
9. Powers, B.J.: A multi-agent architecture for NATO network enabled capabilities: enabling semantic interoperability in dynamic environments (NC3A RD-2376). In: Kowalczyk, R., Huhns, M.N., Klusch, M., Maamar, Z., Vo, Q.B. (eds.) SOCASE 2008. LNCS, vol. 5006, pp. 93–103. Springer, Heidelberg (2008)
10. Turnitsa, C.: Extending the levels of conceptual interoperability model. In: Proceedings IEEE Summer Computer Simulation Conference. IEEE CS Press (2005)
11. Sedek, K., Sulaiman, S., Omar, M.: A systematic literature review of interoperable architecture for e-government portals. In: 2011 5th Malaysian Conference on Software Engineering (MySEC), pp. 82–87 (2011)
12. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, vol. 17 (2008)
13. Kitchenham, B.: Procedures for undertaking systematic reviews,Joint technical report. Computer Science Department, Keele University and National ICT Australia (2004)
14. Guédria, W., Chen, D., Naudet, Y.: A Maturity Model for Enterprise Interoperability. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 216–225. Springer, Heidelberg (2009)
15. Hofmeister, C., Kruchten, Nord, Obbink, Ran, America, P.: Generalizing a model of software architecture design from five industrial approaches. In : 5th Working IEEE/IFIP Conference on Software Architecture, pp.77–88 (2005)
16. Aleti, A., Buhnova, B., Grunske, L., Koziolek, A., Meedeniya, I.: Software architecture optimization methods: A systematic literature review. IEEE Transactions on Software Engineering 39(5), 658–683 (2013)
17. Clark, T., Jones, R.: Organisational interoperability maturity model for C2. In: Proceedings of the 1999 Command and Control Research and Technology Symposium (1999)
18. Edwin, M., Linda, L., Patrick, P., Daniel, P., Meyers, B.: System of systems interoperability (SOSI): Final report. Tech. rep. (2004)