Institute of Mathematics and Computational Sciences

ISSN - 0103-2569

Current State on Representation of Reference Architectures

Milena Guessi
Lucas Bueno Ruas de Oliveira
Elisa Yumi Nakagawa

№ 363

ICMC TECHNICAL REPORT

São Carlos, SP, Brazil
May/2011

Abstract

Software architectures play a major role in determining system quality, since they form the backbone to any successful software-intensive system. In this context, reference architecture is a generic architecture for a class of software systems used as a foundation for the design of concrete architectures from this class. Because of their fundamental role, reference architectures need to be properly represented in order to be effectively used, and understood by all stakeholders. In this perspective, we have found initiatives proposing different approaches to represent them. However, there is a lack of detailed overview about all these approaches. In this work, we identify possibly all approaches used or proposed to represent reference architectures. For this, we have used a technique proposed by Evidence-Based Software Engineering (EBSE): the Systematic Review. As achieved results, we can observe different approaches adopted to represent reference architectures; however, there is no consensus about which are more adequate or their effectiveness. Based on this overview, interesting and important perspectives for future research can also be found.

Contents

List of Figures

List of Tables

Chapter

1

Introduction

Reference architectures have emerged as elements that aim at facilitating and systematizing the development of software systems. However, there is no commonly accepted definition for reference architectures (Angelov et al., 2009). In this work, we considered that a reference architecture aggregates knowledge of a domain, identifying abstract solutions of a problem and promoting reuse of design expertise by achieving solid, well-recognized understanding of a specific domain. The purpose of a reference architecture is to serve as a blueprint for future architectures, aiming at preventing the re-invention or re-validation of solutions for solved problems (Muller, 2008). However, the proposition of reference architectures is not a trivial task, since it requires a comprehensive knowledge of the domain for which it is proposed. In this scenario, considering the relevance of reference architectures, they have been proposed and successfully used in different domains, such as for embedded systems (Eklund et al., 2005) and even for software engineering (Nakagawa et al., 2007).

Particularly, representing the reference architectures in order to make them understandable for a wide variety of stakeholders (such as customers, product managers, project managers and engineers) consists of an essential activity for their success. In this sense, an architectural description comprises the set of documents used to define and document the architecture, and its use is of assistance many times during the software life cycle, for instance, in the evaluation and evolution phases (IEEE, 2000a). Despite the relevance of reference architectures, it has been observed that their representation, in practice, is often conducted informally. Therefore, we have observed a concern in representing reference architectures. However, there is a lack of work that summarizes all lines of work conducted in this perspective or provides direction about which approaches for the representation of these architectures are more adequate.

The main objective of this work is to identify, summarize and analyze possibly all approaches that have been proposed or used to represent reference architectures. For this, we have adopted and conducted a systematic review, an technique used to summarize, assess and interpret all relevant studies related to a specific question, topic area, or phenomenon of interest. Results have pointed out that this is a novelty topic of research and, considering the relevance of reference architectures in the software development, more effort must be focused on this research area. Besides that, we could identify future and important research lines based on our achieved results.

This work is organized as follows. In Chapter 2 we present a brief overview on reference architectures, their description, and on the systematic review technique. In Chapter 3 we present the conducted systematic review. In Chapter 4 we discuss achieved results. In Chapter 5 we summarize our contributions and discuss perspectives for further work.

Chapter

2

A Brief Overview

Software architectures capture and preserve designers' intentions about system structure and behavior (Kruchten et al., 2006). In this context, reference architectures have been proposed, playing a dual role with regard to specific target software architectures (Angelov et al., 2009; Muller, 2008): it generalizes and extracts common functions and configurations; and it provides a base for instantiating target systems. In other words, they can be seen as a knowledge repository of a given domain. In this perspective, the use of reference architectures as artifacts to be reused seems to be relevant.

Particularly, the representation of software architecture is a very important task, since artifacts that facilitate the communication of the architecture to humans or machines are in fact desirable and even essential. In this scenario, we can find the proposition of formal notations in order to describe them, such as the Architectural Descriptions Languages (ADLs) ACME (Garlan, 2000) and WRIGHT (Allen, 1997). Furthermore, the use of multiple architectural views to represent software architectures is widely accepted. An architectural view consists of the view obtained by focusing on a particular perspective of the architecture, for instance, logical, implementation or physical (Clements et al., 2003). Additionally, UML has been recently adopted for the representation of architectural views (Ivers et al., 2004). In this context, we can find approaches to represent reference architectures; however, this is a challenging task, since it is necessary to define the amount of detail required and represent them properly.

In another perspective, during the study of a new knowledge area, researchers usually conduct a bibliographic review (almost always an informal review) to identify publications related to a specific subject. However, this kind of review does not use a systematic approach and does not offer any support to avoid bias during the selection of the publications that will be analyzed. Thus, it is important to have mechanisms to summarize and provide an overview about an area or topic of interest (Petersen et al., 2008). In this context, Evidence-Based Software Engineering (EBSE) (Dybå et al., 2005) has recently arisen, inspired in the medical area, and has given considerable contribution to the Software Engineering area. In this sense, EBSE has investigated and proposed the use of the systematic review technique which provides a comprehensive and systematic evaluation of research using a predefined strategy of search aiming at minimizing bias (Kitchenham e Charters, 2007). In this context, an individual evidence (for instance, a case study or an experimental study divulged in a publication/paper) which contributes to a systematic review is called *primary study*, while the result of a systematic review is a *secondary study*.

Chapter 3

Systematic Review Application

Our systematic review was conducted in the reference architecture context, aiming at identifying all relevant primary studies related to the representation of this architecture. It was conducted from September 1st, 2010 to October 15th, 2010 and involved three people (one software engineering researcher, one systematic review specialist, and one undergraduate student). In order to conduct our systematic review, we followed the process proposed by Kitchenham (Kitchenham, 2004) and presented in Figure3.1. In short, this process is composed by three main phases: (i) planning; (ii) conduction; and (iii) reporting. These phases are explained in more details during presentation of our systematic review.

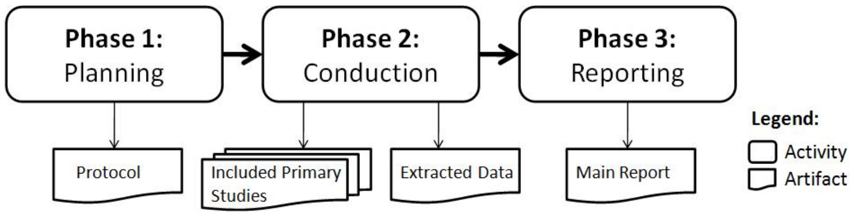

Figure 3.1: Systematic review process (Adapted from (Kitchenham, 2004))

3.1 Phase 1: Planning

In this phase, the objectives and the systematic review protocol are defined. The protocol consists of a predetermined plan that describes the research questions and how the systematic review will be conducted, i.e., the search strategy. The protocol establishes also which are the selection criteria, and the data extraction and synthesis method.

(i) **Research Questions**: Aiming to find all primary studies to understand and summarize evidences about proposed or used approaches to represent reference architectures, the following research question (RQ) was established:

- RQ1: Which are the approaches proposed or used to represent reference architectures?

Additionally, we defined four secondary research questions to specify the objectives of our search:

- RQ2: Which are the UML techniques proposed or used to represent reference architectures?
- RQ3: Which are the ADLs proposed or used to represent reference architectures?
- RQ4: What is the maturity level of the found approaches, considering their application to describe reference architectures?

(ii) **Search Strategy**: In order to establish the search strategy, considering the research questions, we initially identified the main keywords "Reference Architecture" and "Architectural Description". Following, we identified synonyms for these keywords: "Architectural View", "Architecture View", "Architecture Description", "Architectural Model", and "Architecture Model". In particular, we also considered "Reference Model" in our search string, since some authors use this referring to reference architectures. However, it is important to highlight that there are also authors who define reference architectures and reference models differently (OASIS, 2006). The acronym "ADL" was not considered because the term "Architectural Description" was already included. Also, only papers written in English were considered in our systematic review, since it is the most common language in scientific papers. We used the boolean OR operator to link the main terms and their synonyms. Finally, all these terms were combined using the boolean AND operator. Thus, the final search string was:

```
(("Reference Architecture" OR "Reference Model")
AND ("Architectural View" OR "Architecture View" OR
"Architectural Model" OR "Architecture Model" OR
"Architectural Description" OR "Architecture Description"))
```

In addition to the search strategy, we established which search sources (i.e., publication databases) would be used to find the primary studies. In order to select the most adequate publication databases for our search, we considered the criteria discussed in (Dieste et al., 2009). Thus, the criteria used to select the sources were: (i) content update (regularity with which the articles are published); (ii) availability (evaluate if the full text of the works are available); (iii) quality of results (accuracy of the results returned by the search); and (iv) versatility export (existence of a mechanism to export the results). Thus, the selected databases chosen for our systematic review are shown in Table**??**. Furthermore, Scopus was added, since it is considered the largest database of abstracts and citations (Kitchenham e Charters, 2007).

Table 3.1: Selected Databases

Source	Location
ACM Digital Library	www.portal.acm.org
IEEE Xplore	www.ieeexplore.ieee.org
ScienceDirect	www.sciencedirect.com
Scopus	www.scopus.com
Springer	www.springer.com/lncs
Web of Science	www.isiknowledge.com

(iii) **Inclusion and Exclusion Criteria:** The selection criteria are used to evaluate each study recovered from the publication databases. In this sense, these criteria make it possible to include only studies that are relevant to the research topic. Thus, the inclusion criteria (IC) for our systematic review were:

- IC1: The primary study proposes or uses an approach to represent reference architectures; and
- IC2: The primary study proposes or uses an ADL to describe reference architectures.

Otherwise, the exclusion criteria are used to exclude the studies that do not contribute to answer the research questions. Thus, the exclusion criteria (EC) established were:

- EC1: The primary study does not address reference architectures;
- EC2: The primary study does not propose or use any approach to represent reference architectures;
- EC3: The primary study does not present an abstract or its full text is not available;

- EC4: The primary study is written in a different language than English;
- EC5: The primary study is directly related to another primary study of the same author. In this case, only the most recent primary study will be considered; and
- EC6: The primary study consists of a compilation of work, for instance, from a conference or workshop.

(iv) **Data Extraction and Synthesis Method:** In order to extract data, we plan to build data extraction tables related to each research question. These tables must synthesize results aiming at facilitating achieving conclusions. During the extraction process, the data of each primary study will be independently extracted by two reviewers. If disagreement occurs, discussion will be conducted. To summarize and describe the set of data, the statistical synthesis method will be used.

3.2 Phase 2: Conduction

In this phase, the primary studies are identified, selected and evaluated according to the previously established selection criteria. For each selected study, data is extracted and synthesized. The identification is done by looking for all primary studies that match the search string in the search sources. This match is automatically done by the search engines of the selected sources. It is worth highlighting that only primary studies published until September 14, 2010 are considered in this work.

To support the organization of the primary studies, we used JabRef[1], an open source reference management system. It made it possible to store information of the primary studies (including, for instance, title, authors, and abstract), as well as the inclusion/exclusion criteria applied to select each primary study.

As a result of our search in the publication databases using the search string, 110 primary studies were recovered. Our first measure was to remove repeated primary studies, since some databases, such as Scopus, may index studies of other databases, such as IEEEXplore and Springer. Thus, we found 59 unique studies. Then, the title and abstract sections of these studies were read and the selection criteria were applied. Thus, a total of 13 studies were selected for further reading. Next, these studies were read in full and the selection criteria were again applied. In this step, one study was not available fully, one study does not propose or use an approach to represent reference architectures, and one was directly related to another primary study of the same author. Finally, 10 studies were considered relevant for our systematic review.

[1]`http://jabref.sourceforge.net`

Table 3.2 summarizes the total primary studies recovered in each database, the number of included studies, the rate index[2] and the search date. For instance, from ACM Digital Library, we obtained a total of 19 primary studies; applying the selection criteria, six studies were included; the index rate is 60% (i.e., 6/10). It is important to observe that ACM Digital Library and Scopus were the most efficient sources, since each of them had six studies included. Particularly, if the search was limited to these two databases (ACM Digital Library and Scopus), we would find nine of the 10 primary studies included. Otherwise, Science Direct contributed with only one paper.

Table 3.2: Search sources, obtained and included primary studies

Database	Obtained	Included	Rate Index	Date
ACM Digital Libray	19	6	60%	Sept 14,2010
IEEEXplore	21	3	30%	Sept 14,2010
ISI Web of Science	18	3	30%	Sept 14,2010
Science Direct	5	1	10%	Sept 14,2010
Scopus	38	6	50%	Sept 14,2010
SpringerLink	9	2	20%	Sept 14,2010

Table 3.3 presents the 10 selected primary studies. Column "Type" indicates if the study was published in a Journal Article (JA), Conference Paper (CP), Technical Report (TR) or a Book Chapter (BC). Moreover, most studies were included by IC 1 (i.e., the primary study proposes or uses an approach to represent reference architectures). Following, a more detailed analysis was conducted on the 10 primary studies included in our systematic review.

3.3 Phase 3: Reporting

In this last phase, we present analytical results of our systematic review. The obtained data extraction and synthesis of knowledge considering each research question is discussed next.

RQ1: Regarding RQ1 (i.e., approaches proposed or used to represent reference architectures), we have identified 10 different adopted approaches to the representation of reference architectures. Table 3.4 summarizes each of the found approaches. Particularly, studies S4, S5, S9, and S10 represented the reference architecture using a graphic approach other than UML. Also, study S8 proposed an architectural view, the Conceptual View, which describes by means of ontologies each of the terms that are present in the

[2]Ratio between the included studies of a database and the total of primary studies included in the systematic review.

Table 3.3: Included primary studies

Study	**Authors**	**Publication Year**	**Inclusion Criteria**	**Type**
S1	Bashroush et al. (Bashroush et al., 2005)	2005	IC2	CP
S2	Chen et al. (Chen et al., 2010)	2010	IC2	JA
S3	Fritschi and Gatziu (Fritschi e Gatziu, 1999)	1999	IC2	TR
S4	Hofmann et al. (Hofmann et al., 2009)	2009	IC1	CP
S5	Lin et al. (Lin et al., 2008)	2008	IC1	CP
S6	Lopez and Blobel (Lopez e Blobel, 2009)	2009	IC1	JA
S7	Meland et al. (Meland et al., 2009)	2009	IC1	CP
S8	Nakagawa and Maldonado (Nakagawa e Maldonado, 2008)	2008	IC1	CP
S9	Sarathy et al. (Sarathy et al., 2010)	2010	IC1	CP
S10	Schroth (Schroth, 2008)	2008	IC1	BC

reference architectures. It is worth noting that this diversity points out that there is no consensus about how to represent reference architectures. Further detailed analysis of each approach is presented next.

Table 3.4: Short description of the approach proposed or used in each primary study

Study	**Approach proposed or used in the primary study**
S1	Architectural Description Language ADLARS
S2	Architectural Description Language π-ADL
S3	Architectural Description Language WRIGHT
S4	Model-View-Controller and Moderator Architectural Styles
S5	Layers Architectural Style
S6	Business, Information and Computational Architectural Views
S7	Component Architectural View and Information Model
S8	Module, Runtime, Deployment and Conceptual Architectural Views
S9	Logic and Physical Architectural Views
S10	Community, Process, Service and Infrastructure Architectural Views

RQ2: This research question addresses UML techniques proposed or used to represent reference architectures. Table 3.5 summarizes the UML techniques found in the primary studies. For instance, Lopez and Blobel (Lopez e Blobel, 2009) (study S6) adopted three architectural views: enterprise, information and computational. Then, the authors used, respectively, the use case diagram, the class diagram and the component diagram of UML to represent each of these views.

Table 3.5: UML techniques proposed or used in the representation

Study	Architectural View	UML Technique
S6	Enterprise View	Use Case Diagram
S6	Computational View	Component Diagram
S6	Information View	Class Diagram
S7	Component and Connector View	Component Diagram
S7	Information Model	Class Diagram
S8	Deployment View	Deployment Diagram
S8	Module View	Class Diagram
S8	Runtime View	Component Diagram

The enterprise view offers a perspective of the system's architecture and environment, describing the system's purpose, scope and policies. The computational view shows the logical components and their interactions through interface. The information view, also referenced by study S7 as an information model, offers a perspective on the information's structure and its semantics. The runtime view proposed by Nakagawa and Maldonado (Nakagawa e Maldonado, 2008) is also known as Component and Connector View (C&C) by the "Views and Beyond" method (Clements et al., 2003). The C&C view shows the system as a set of cooperating units of runtime behavior and was also proposed in study S7. The deployment view describes the machines, software that is installed on those machines and network connections that are used by the software systems. Finally, the module view shows how a software system is structured as a set of implementation units.

It is important to notice that most of the UML techniques were not applied for the representation of these architectures, since only the use case diagram, component diagram, class diagram and deployment diagram were proposed or used.

RQ3: This research question addresses the ADLs proposed or used to represent reference architectures. Table 3.6 summarizes the ADLs found in the primary studies. In spite of the diversity of ADL available in the literature, there are few initiatives of using ADL to represent reference architectures. Bashroush et al. (study S1) developed the ADLARS (Architecture Description Language for Real-time Software Product Lines), an ADL to be used in the software product line context. This ADL aims at supporting the relationship between the system's feature model and the architectural structures. The feature model encompasses the variability and commonality among the different products within the scope of a family. Thus, the link between the feature model and the architectural structures allows for automatic generation of product architectures from the family reference architecture, making the task of deriving product-specific architectures much more straightforward.

Table 3.6: ADLs proposed or used to represent reference architectures

Study	Architectural Description Language
S1	ADLARS
S2	π-ADL
S3	WRIGHT

Chen et al. (study S2) applied π-ADL to describe the High-Level Architecture (HLA), which is a reference architecture and a common infrastructure for large scale distributed interactive simulation systems and was also accepted as an IEEE standard in 2000 (IEEE, 2000b). In π-ADL, architectures are composite elements representing systems. In this sense, a composite consists of external interfaces and internal behavior. Thus, π-ADL supports the description of composite dynamic behaviors. Additionally, the virtual machine for the system dynamic analysis, called π-ADLVM, can be used to test the architectural models written in π-ADL.

Fritschi and Gatziu (study S3) presented the representation of reference architecture by means of WRIGHT (Allen, 1997). This language presents the basic architectural abstractions of components, connectors and configurations, providing explicit structural notations for each of these elements. Particularly, the configuration is a collection of component instances combined via connectors.

RQ4: This research question addresses the maturity level of the approaches, considering their application to describe reference architectures. In order to answer this question, we classified each study in one of the following categories: (i) proposal, if the primary study proposes an approach to represent reference architectures, but does not present any case study; (ii) case study, if the primary study presents at least one case study; and (iii) application, if the approach is already been used in a project or in the development industry. Table 3.7 shows the maturity level of each approach. It is observed that only two studies presented a case study. In this sense, more studies proposing or applying the found approaches were obtained, even though part of them was adopted from software architectures. In this sense, it seems interesting to concentrate efforts to increase the maturity level of these approaches and to verify if they are adequate for the representation of reference architectures.

Table 3.7: Maturity level of the found approaches

Maturity Level	Total	Percentage	Primary Studies
Proposal	4	40%	S1, S4, S5, S10
Case study	2	20%	S2, S8
Application	4	40%	S3, S6, S7, S9

Chapter 4

Discussion

Results of our systematic review point out that, in spite of relevance of reference architectures to the software development, there are few recent works that show concern with their adequate representation. Therefore, more studies are necessary that investigate which are and how other, if any, architectural views can be used to properly describe reference architectures. Also, investigation of well-known graphic notations, such as UML, would be relevant, aiming at standardizing the representation of such architectures. Additionally, the use of ADLs in the context of reference architectures should be further investigated concerning their application and suitability.

Regarding the research questions established for our systematic review, it is noticeable that all of them were successfully answered. Besides that, we believe that the results presented in this work are representative of the whole universe of existing approaches to represent reference architectures, since systematic reviews provide mechanisms to achieve it.

Considering knowledge arisen from this work, it is possible to identify interesting and important research lines that can be investigated in future work. For instance, investigation or proposition of software tools to support the representation and modeling of reference architectures; conduction of case studies in order to increase the maturity level of the found approaches; investigation of how well-known graphic notation could be applied in order to represent different detail levels of reference architectures; and, establishment of a consensus of the best approaches to the representation of reference architectures.

Regarding limitation of this work, other research sources could be selected, such as Google Scholar, and other keywords could be added to the search string. Moreover, it is worth highlighting that systematic review conduction is not a trivial task because of the amount of papers that need to be manipulated. Besides that, relevant primary studies written in other languages than English were overlooked in this instance.

Chapter 5

Conclusion

The main contribution of this work was to present a detailed panorama about proposal and use of approaches for the representation of reference architectures. Based on this panorama, another important contribution is to bring forth new research lines on this topic. As main result, we found that the representation of reference architectures has not been sufficiently explored and there are still different perspectives that could be investigated in order to deal adequately with reference architectures, aiming at promoting their dissemination and effective use.

Acknowledgments:

This work is supported by Brazilian funding agencies: FAPESP, CAPES, and CNPq.

References

Allen, R. J. *A Formal Approach to Software Architecture.* Tech. Report CMU-CS-9-144, Carnegie Mellon University, 1997.

Angelov, S.; Grefen, P.; Greefhorst, D. A Classification of Software Reference Architectures: Analyzing Their Success and Effectiveness . In: *ECSA'09 at WICSA'09*, Cambridge, UK, 2009, p. 141–150.

Bashroush, R.; Brown, T. J.; Spence, I.; Kilpatrick, P. ADLARS: An Architecture Description Language for Software Product Lines. In: *SEW'05*, Maryland, USA, 2005, p. 163–173.

Chen, J.; Wu, D.; Zhang, J.; Oquendo, F. Formal modelling and analysis of HLA architectural style. *Int. Journal of Modelling, Identification and Control*, v. 9, n. 1-2, p. 71–82, 2010.

Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; Stafford, J. *Documenting Software Architectures: Views and Beyond.* Addison-Wesley, 560 p., 2003.

Dieste, O.; Grimán, A.; Juristo, N. Developing Search Strategies for Detecting Relevant Experiments. In: *Empirical Software Engineering*, v. 14, Springer Netherlands, p. 513–539, 2009.

Dybå, T.; Kitchenham, B.; Jorgensen, M. Evidence-based software engineering for practitioners. *IEEE Software*, v. 22, n. 1, p. 58–65, 2005.

Eklund, U.; Örjan Askerdal; Granholm, J.; Alminger, A.; Axelsson, J. Experience of introducing reference architectures in the development of automotive electronic systems. *SIGSOFT*, v. 30, n. 4, p. 1–6, 2005.

Fritschi, H.; Gatziu, S. *A Reusable Architecture to Construct Active Database Systems.* Tech. Report ifi-99.02, University of Zurich, 1999.

Garlan, D. Software architecture: a roadmap. In: *ICSE'00*, Limerick, Ireland, 2000, p. 91–101.

Hofmann, C.; Hollender, N.; Fellner, D. W. Workflow-Based Architecture for Collaborative Video Annotation. In: *OCSC'09 at HCII'09*, San Diego, USA, 2009, p. 33–42 (LNCS v.5621).

IEEE IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. 2000a.

IEEE IEEE 1516-2000 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA). 2000b.

Ivers, J.; Clements, P.; Garlan, D.; Nord, R.; Schmerl, B.; Silva, J. R. O. *Documenting Component and Connector Views with UML 2.0.* Tech. Report CMU/SEI-2004-TR-008, 2004.

Kitchenham, B.; Charters, S. *Guidelines for performing systematic literature reviews in software engineering.* Tech. Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

Kitchenham, B. A. *Procedures for performing systematic reviews.* Tech. Report TR/SE-0401 and NICTA TR 0400011T.1, Keele University and National ICT Australia Ltd, 2004.

Kruchten, P.; Obbink, H.; Stafford, J. The Past, Present and Future of Software Architecture. *IEEE Software*, v. 23, n. 2, p. 22–30, 2006.

Lin, C.; Lu, S.; Lai, Z.; Chebotko, A.; Fei, X.; Hua, J.; Fotouhi, F. Service-Oriented Architecture for VIEW: A Visual Scientific Workflow Management System. In: *IEEE SSC'08*, Honolulu, USA, 2008, p. 335–342.

Lopez, D. M.; Blobel, B. G. A development framework for semantically interoperable health information systems. *Int. Journal of Medical Informatics*, v. 78, n. 2, p. 83–103, 2009.

Meland, P. H.; Ardi, S.; Jensen, J.; Rios, E.; Sanchez, T.; Shahmehri, N.; Tondel, I. A. An architectural foundation for security model sharing and reuse. In: *ARES'09*, Fukuoka, Japan, 2009, p. 823–828.

Muller, G. *Reference Architecture Primer.* Tech. Report version 0.6, Embedded Systems Institute, 2008.

Nakagawa, E. Y.; Maldonado, J. C. Reference architecture knowledge representation: an experience. In: *SHARK'08 at ICSE'08*, Leipzig, Germany, 2008, p. 51–54.

Nakagawa, E. Y.; Simão, A. S.; Ferrari, F.; Maldonado, J. C. Towards a reference architecture for software testing tools. In: *SEKE'07*, Boston, USA, 2007, p. 1–6.

OASIS *Reference model for service oriented architecture 1.0.* Tech. Report SOA-RM, OASIS Standard, 2006.

Petersen, K.; Feldt, R.; Mujtaba, S.; Mattson, M. Systematic Mapping Studies in Software Engineering. In: *EASE'08*, Bari, Italy, 2008, p. 71–80.

Sarathy, V.; Narayan, P.; Mikkilineni, R. Next Generation Cloud Computing Architecture: Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure. In: *WETICE'10*, Larissa, Greece, 2010, p. 48–53.

Schroth, C. A Service-oriented Reference Architecture for Organizing Cross-Company Collaboration. In: *Enterprise Interoperability III*, Springer, p. 71–83, 2008.