

Different Notions Lead Difficulties of Making Information System Modernization Decisions

Miia-Maarit Saarelainen and Virpi Hotti
School of Computing
University of Eastern Finland
Finland
miia.saarelainen@uef.fi, virpi.hotti@uef.fi

ABSTRACT

If organizational purposes or functions changes then usually we are concerned with information system modernization. However, there are difficulties of making information system modernization decisions because of suppliers and clients may have different notions about modernization. In this paper, we present a review and qualitative analysis of statements for information system modernization. The objective of this review is to describe current reported knowledge in terms of what kind of modernization is defined, what is the domain of modernization, is modernization a synonym for reengineering, and does modernization include replacement situations. We found 42 statements for modernization from 36 papers. These findings show that the concept of modernization is unestablished. However, if we want to have similar notions about information system modernizations then we have to be transparent where the changes have effect. Therefore, one solution based on the analysis of relationships between the entities of enterprise architecture is presented to help for making information system modernization decisions.

KEYWORDS

software modernization, system modernization, architecture-driven modernization, literature review, content analysis

1 INTRODUCTION

When something is becoming modern or something is made modern, it might mean that something is more suitable for present styles or needs, or the newer information, methods, or technology is used [1]. Nowadays, we are dependent on various information systems. The

information systems can be seen as a collection of people, processes, data, models, technology and partly formalized language, which serves an organizational purpose or function [2]. If the organizational purpose or function changes, then we have to make something modern and quite often we are concerned with information system modernization.

Many information systems can be called legacy systems. Legacy systems are typically described as mission critical systems, systems that are in an important role in an organization, and systems which are vital but unmanageable [3, 4]. Bennett [4] has further described that legacy systems may contain years of accumulated experience and knowledge and, in some cases, the system may be the only place where an organization's business rules exist. Wallace et al. [5] cited that system "modernization is a specialized application of system reengineering, which is the disciplined evolution of a system from its current state to a new one". Weiderman et al. [6] present five activities which can be used with legacy systems: assessment, maintenance, transformation (can be seen as reengineering), replacement and combined strategies. Comella-Dorda et al. [7] divided the system evolution activities into the following categories: maintenance, modernization and replacement. Furthermore, they define that modernization involves more extensive changes than maintenance and the system must be replaced when it can no longer be evolved.

In this paper, we are interested in information system modernization. Our earlier studies have discussed decision-making in modernizations [8-

10], but during our research we have been doubtful about the concept of modernization. Suppliers and clients may have different notions about modernization[11], which causes difficulties of making modernization decisions.

Although modernization has been studied from different perspectives, the concept of modernization is vague. Our motivation is to find what modernization is. A definition can be described as “*a statement expressing the essential nature of something*” [12]. Suonuuti [13] states: “*Definitions are statements describing a concept*”. In this study, we review and analyze the statements of information system modernization. We collect the statements by using a literature review. In our review, a statement is something which tells us what modernization is. Our search returned altogether 1056 hits of which 36 papers included some kind of a statement or statements of modernization. Statements were further analyzed with qualitative methods. The key findings present the variations of the modernization concept and the need for defining concepts consistently.

This paper is organized as follows: Section 2 introduces our research methods, explains how literature review was conducted and the limitations of the study; Section 3 presents the data extraction and synthesis, as well as, explains further analysis and finally Section 4 discusses our results, and presents one solution based on the analysis of relationships between the entities of enterprise architecture to help making information system modernization decisions.

2 METHODS

We are interested in the statements of information system modernization. Therefore, we need to find out how the concept of modernization is defined in scientific papers. The data collection method in this study is literature review. Qualitative methods (content analysis and creating themes) are used for analyzing the collected material.

Research literature can be reviewed for different purposes: to provide a theoretical background for research, to learn the breadth of the research field

or to answer practical questions by finding out what is said in existing research literature [14]. In our case, the purpose for the literature review is to learn what is said in the existing literature. Our review has been conducted by adapting two systematic review guidelines [14, 15]. Okoli and Schabram [14] describe that a literature review is systematic, when it 1) follows methodological approach systematically, 2) explains explicitly how review has been conducted, 3) includes all relevant material comprehensively, and 4) can be reproduced by other researchers. Kitchenham and Charters [15] define systematic literature review as “a form of secondary study that uses a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable”. In our review, we have decided not to assess the quality of the studies and thus, our review is not systematic in the means of Kitchenham and Charters [15]. Moreover, in our case, we look for statements of modernization and we use all the scientific articles, not only primary studies, from the computing field which can be retrieved from the electronic databases we have chosen. However, we have defined our review process as follows:

1. Specifying the search terms
2. Selecting the databases
3. Creating the selection criteria
4. Searching the papers
5. Appraising the hits and selecting the papers
6. Citing the definitions from the papers

Our research strategy is a mix of exploratory and descriptive strategies. Exploratory strategy tries to find out what is happening, seeks new insights and generates new ideas and hypotheses for future research and descriptive strategy describes the current status of a phenomenon or a situation [16]. A systematic literature review aims at identifying, evaluating and interpreting all available research material to answer a research question [15]. In our study, we collected all scientific papers from the field of computing using the databases we have access to. We did not use books or grey material or manual searches. However, we used all the databases, which were available to us.

Typically, the quality of the material accepted to the review should be assessed [15]. However, we did not assess the quality of the accepted papers. We were interested in the statements (if they exist or not), not the quality of the papers. Therefore, we decided to include all statements which tried to explain what modernization is. In our opinion, our accepted statements represent well the statements and their quality in the field of computing.

After going through the literature, we started the further analysis. First, we used content analysis and second, created themes. Content analysis is the quantification of qualitative text material [17]. Content analysis starts when the researcher creates codes. The codes are sets of words and phrases found from the text. After that, the researcher searches for occurrences of those codes from the text. Themes are created from the text by paying attention to the interesting attitudes, words and phrases which describe the research subject well [18].

We wanted to find out how the concept of *modernization* has been defined. Our main research question is: How is software/system modernization defined? To answer the question, we conducted pilot searches with different search terms to find out the best terms. We used ACM Digital Library as a pilot database and we tried different search terms e.g. (software AND modernization), (system AND modernization), (software AND modernization) OR (system AND modernization), “software modernization” OR “system modernization”. We had decided to search all fields, not only the abstract or the title, because a paper can include a statement although the terms are not mentioned in the title or the abstract. After we piloted our search, we decided that the best way to search for the answer to our question is to use the search term “software modernization” OR “system modernization”. We came up this search string, because we were interested in what is modernized: the software, the information system, the software system, the legacy system or the legacy software. Our search string finds all of these terms.

We used the following databases to conduct our search:

- ACM Digital Library¹
- IEEE Xplore²
- SpringerLink³
- Elsevier Science Direct⁴
- EBSCOhost Academic Search Elite⁵
- Wiley Online Library⁶
- Emerald⁷

We did not use manual searches. We included the papers which

- are scientific conference or journal papers
- come from the field of computing
- are available in full versions
- are written in English
- include a statement of (software/system) modernization

We excluded papers which are written by the authors of this paper or are found an earlier search from other databases (Figure 1).

Altogether, we received 1056 hits from the selected databases. Finally, after going through all the databases, we had 36 articles where modernization was defined (Table 1).

Table 1. Searches

Database	Number of hits	Selected articles
ACM Digital Library	108	23
IEEE Xplore	296	9
SpringerLink	228	-
Elsevier Science Direct	220	2
Wiley Online Library	122	2
Emerald	18	-
EBSCOhost	64	-
Total	1056	36

¹ <http://dl.acm.org/>

² <http://ieeexplore.ieee.org/Xplore/home.jsp>

³ <http://link.springer.com/>

⁴ <http://www.sciencedirect.com/>

⁵ <http://www.ebscohost.com/academic/academic-search-elite>

⁶ <http://onlinelibrary.wiley.com/>

⁷ <http://www.emeraldinsight.com/>

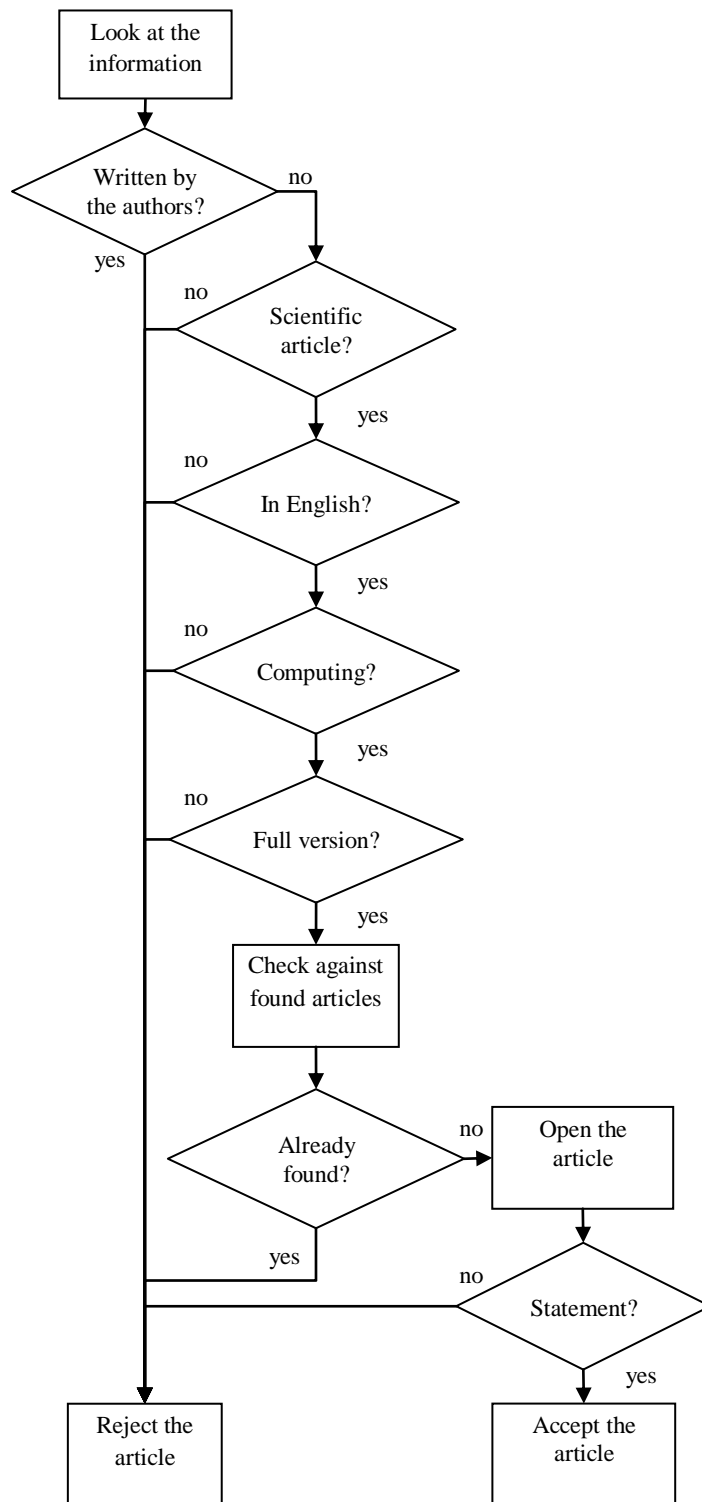


Figure 1. The paper acceptance process

3 EXTRACTING AND SYNTHESIZING THE DATA

Modernization is named software modernization in 13 statements, ADM in 9 statements, modernization in 7 statements, and system modernization in 5 statements. In eight cases modernization is named divergently: LIS modernization, modernization of enterprise systems (in two cases), modernizing a legacy system (in two cases), model-driven modernization (in two cases) and software system modernization. We compared the modernization domains to the defined modernizations (Table 2).

Table 2. Comparison of modernization domains and defined modernizations

	Defined modernization								
Domain	ADM	LIS modernization	model-driven modernization	modernization	modernization of enterprise systems	modernizing a legacy system	software modernization	software system modernization	system modernization
aged legacy system							1		
aging software system							1		
enterprise system									1
existing applications	1								
existing IT assets					2				
existing software	1								
existing software artifacts							1		
existing software assets	4						3		
existing system	2			1			1		
legacy information system	1						1		
legacy software			1						
legacy system			1	1			2		
LIS		1					3		
software system				1				1	
system				4		2			4

In system modernization and modernization, the domain is typically some kind of system. In

software modernization the domain can be some kind of system and the software artifacts or assets. In ADM the possible domains are some kind of system, the applications, the software and the software assets.

The use of the statements is not straightforward. In three papers [19-21], the writers use the term software modernization, but they define it with the statement of Architecture-Driven Modernization [22]: *“ADM is the concept of modernizing existing systems with a focus on all aspects of the current systems architecture and the ability to transform current architectures to target architectures.”* Instead of “ADM” the writers use “software modernization” [20, 21] and “the software modernization paradigm” [19]. The writers have replaced “existing systems” with “LISs” [20, 21] and “legacy information systems” [19]. Also, in one paper [23], the writers use different terms synonymously: first software modernization, then modernization of legacy software systems.

Comella-Dorda et al. cite Weiderman et al. [6]. Weiderman et al. do not use the concept of modernization, but transformation. Weiderman et al. further explain that transformation can be seen as reengineering [6]. Comella-Dorda et al. divide modernization into white-box and black-box modernization. This idea was also first presented in Weiderman et al.’s report [6]. Seacord et al. [24] define modernization similarly to Comella-Dorda et al. Comella-Dorda et al. use the concept “system modernization” whereas Seacord et al. use the concept “software modernization”. Seacord is one of the writers in Comella-Dorda et al.’s paper. The statement of modernization seems to have been developed in Software Engineering Institute during the research project of reengineering.

Recent articles [19, 25-29] seem to cite OMG’s statements of Architecture-Driven Modernization (ADM). In Architecture-Driven Modernization Task Force Glossary of Definitions and Terms [22] the following definition is used *“ADM is the concept of modernizing existing systems with a focus on all aspects of the current systems architecture and the ability to transform current*

architectures to target architectures”. However, in ISO/IEC 19506 -standard [30] Architecture-Driven Modernization is defined as follows: *“ADM is the process of understanding and evolving existing software assets of a system of interest. ADM focuses at collecting, sharing, utilizing, transforming, presenting, maintaining and storing models of the architectural aspects of existing systems. ADM does not preclude source-to-source migrations (where appropriate), but encourages user organizations to consider modernization from an analysis and design perspective. In doing so, project teams ensure that obsolete concepts or designs are not propagated into modern languages and platforms”*.

In two papers, modernization and reengineering are discussed as synonyms [31, 32]. Chung et al. [32] write that the “term, “modernization,” was used for software reengineering of a legacy system”. Cho et al. [31] present the reengineering method and tools, but the paper is named “The MARMI-RE methodology: a method and tools for legacy system modernization”.

In two papers [7, 33], however, reengineering is presented as a subtopic of modernization. Canfora et al. [33] summarize: *“the problem of modernizing a legacy system [...] was addressed by several authors [...] who concentrated their efforts on three main research topics: the reengineering of a traditional (i.e. not Web-based) system into a Web-based one”*. Comella-Dorda et al. [7] state: *“There are different approaches to the modernization of legacy assets including reengineering”*.

Further, five papers describe that modernization is based on reengineering, but model-driven development principles have been added to traditional reengineering [19, 20, 28, 29, 34]. In these papers, it is said: *“Software modernization does not replace traditional reengineering, but it combine it together with new model-driven development principles”* [20], *“ADM is based on reengineering, but it considers different models as input and output artifacts of the process”* [28], *“ADM [...] advocates carrying out the reengineering process but considering model-*

driven development principles” [29], “Architecture-driven modernization advocates carrying out reengineering processes by following the MDA standard” [19], and “ADM [...] advocates carrying out reengineering processes by considering model-driven development principles” [34].

Three papers [7, 35, 36] distinguish modernization and replacement. Modernization is typically described as a solution to lengthen the life cycle of a business critical system or software and to avoid their replacement. However, in two papers [37, 38] modernization is described as the replacement of a system.

When we went through the statements we noticed that there were similarities in the statements. We wanted to analyze the statements to find out their origins because it seemed that most of the used statements came from the following sources: Comella-Dorda et al.’s paper [7] and from the sources of OMG’s Architecture Driven Modernization Task Force. We also noticed that some of the statements brought up reengineering. Therefore, we wanted to find out if our observations were trustworthy and we started with a content analysis. Our codes were phrases from the original statements of Comella-Dorda et al. and OMG. We also looked for the term “reengineering”. The following themes characterized the following phrases which were used as codes:

- *Comella-Dordaish*. “more [...] changes than maintenance” or “business value that must be preserved” or “changes” or “represents greater effort than [...] maintenance” [7]
- *ADMish*. “understanding and evolving existing software” [30] or “all aspects of the current system’s architecture and the ability to transform current architectures into target architectures” [22]
- *Reengineering*. “reengineering”
- *Different*.

We have cited the statements as they were in the original text. We underlined the defined

modernization. Furthermore, we bolded and italicized the domain of the modernization.

Statements, which described well the theme Comella-Dordaish were similar as following:

1. *Modernization involves major changes to a **system**, but which preserve a significant amount of the old system [39].*
2. *These system evolution activities can be divided into three categories: maintenance, modernization, and replacement [...]. Repeated system maintenance supports the business needs sufficiently for a time, but as the **system** becomes increasingly outdated, maintenance falls behind the business needs. A modernization effort is then required that represents a greater effort, both in time and functionality, than the maintenance activity. Finally, when the old system can no longer be evolved it must be replaced [31].*
3. *Software system evolution activities can be divided into three categories: maintenance, modernization and replacement [...]. Modernization involves extensive and pervasive changes, requiring a significantly greater effort than during maintenance activities [40].*

The theme ADMish was created from similar statements as following:

1. *ADM is a process for understanding and evolving **existing software assets** in order to restore the value of existing applications [27].*
2. *Software modernization is the concept of evolving **LIS** with a focus on all aspects of the current system’s architecture and the ability to transform current architectures into target architectures [21].*
3. *We refer to the software modernization as the process of understanding and evolving **existing software assets** in order to: maintain, integrate, refactor, reuse, migrate, or replace [41].*

Statements which described the theme Reengineering were:

1. *System modernization is a specialized application of system reengineering, which is the disciplined evolution of a **system** from its current state to a new one* [5].
2. *The term, "modernization," was used for software reengineering of a **legacy system**. Software reengineering in this case consists of reverse engineering and forward engineering* [32].
3. *Software modernization, and particularly Architecture-Driven Modernization, has become the best solution in the **legacy systems'** evolution [...]. Software modernization is a new specific kind of evolutionary maintenance paradigm to solve reengineering problems* [29].
4. *Reengineering has normally been used to obtain new improved versions of **aged legacy system** [...]. Software modernization helps to solve the formalism and standardization problems of reengineering* [34].

Following statements described the theme Different:

1. *A key area in software modernization is renovating **aging software systems** to take advantage of today's parallel and distributed computing environments* [42].
2. *Software modernization converts **legacy systems** into component-based systems. The process involves program understanding, business rules extraction, and software transformation* [43].
3. *Modernization of enterprise systems creates new opportunities for corporations to transform their **existing IT assets** into the service-oriented architectures (SOA)* [44].
4. *The immediate effect of software ageing is that information systems become Legacy Information Systems (**LIS**) and must therefore be replaced with a new, improved system. This activity is known as software modernization* [37].

Ten statements in [7, 11, 31, 33, 35, 36, 39, 40, 45] belong to the theme "Comella-Dordaish", also Comella-Dorda et al.'s own statements, sixteen

statements in [19-21, 23, 25-28, 38, 41, 46, 47] to the theme "ADMish", four [5, 29, 34, 48] to the theme "Reengineering", and twelve in [36, 37, 42-44, 49-55] to the theme "Different". These themes highlight that 16 statements resemble OMG's statements of Architecture-Driven Modernization, 10 statements are similar to Comella-Dorda et al.'s statements, 4 statements bring up the relations between modernization and reengineering, and 12 statements differ from the others.

4 DISCUSSION

We noticed that the defined modernization (system, software, architecture-driven) does not reveal what the domain of modernization is. In the statements of system modernization or modernization, the domain of modernization was some kind of system. However, in the statements of software or architecture-driven modernization, the domain of modernization varied from system to software assets and system architecture. Furthermore, we noticed that contradictory statements were used. Sometimes it seems that it does not matter what word comes before modernization. System and software modernization can be defined similarly as well as software and architecture-driven modernization. Our findings reveal the vagueness of the term modernization

This study has revealed that typical sources of modernization statements are Comella-Dorda et al. and OMG's Architecture-Driven Modernization statements. If we look at how the statements have evolved it seems that the term modernization was at the beginning *system* modernization [5, 7, 56], then came software modernization and finally OMG started to talk about *Architecture-Driven* Modernization [57].

If we want to have similar notions about information system modernizations, then the changes have to be transparent. Therefore, we have studied enterprise architecture frameworks, especially TOGAF Content Metamodel [58], and how to make information system modernization decisions based on the analysis of relationships between the entities. We have identified at least

four entities (Figure 2), which should be taken into account when making the decision.

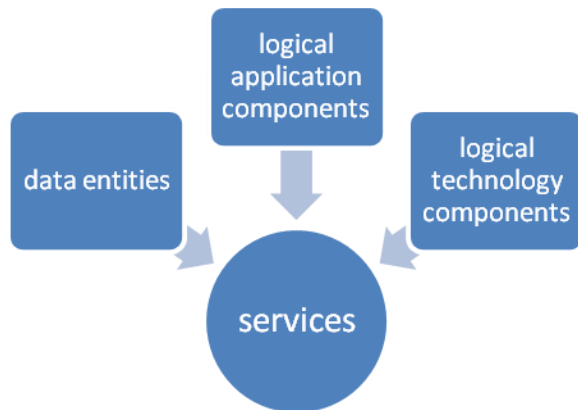


Figure 2. Entities which the changes in the operation affect

Services provide governed interfaces to access organizational functions, as well as, they provide and consume data entities. Therefore, if organizational purposes or functions are going to be changed, it is reasonable to list all data entities which are provided and consumed by services to be changed. Furthermore, we have to be familiar with the logical components of the information system because the services are realized through logical application components and implemented on logical technology components.

5 REFERENCES

- [1] Merriam-Webster.com., "modernize", 2014, <http://www.merriam-webster.com/dictionary/modernize>.
- [2] R. Hirschheim, H.K. Klein and K. Lyytinen, Information systems development and data modeling : conceptual and philosophical foundations, Cambridge: Cambridge University Press, 1995, pp. 289.
- [3] J. Bisbal, D. Lawless, Bing Wu and J. Grimson, "Legacy information systems: issues and directions," *Software, IEEE*, vol. 16, pp. 103-111, 1999.
- [4] K. Bennett, "Legacy systems: coping with stress," *Software, IEEE*, vol. 12, pp. 19-23, 1995.
- [5] E. Wallace, P.C. Clements and K.C. Wallnau, "Discovering a system modernization decision framework: a case study in migrating to distributed object technology," in *Proceedings of the 1996 International Conference on Software Maintenance*, pp. 185, 1996.
- [6] N. Weiderman, L. Northrop, D. Smith, S. Tilley and K. Wallnau, "Implications of Distributed Object Technology for Reengineering," Software Engineering Institute, Carnegie Mellon University., Tech. Rep. CMU/SEI-97-TR-005, 1997.
- [7] S. Comella-Dorda, K. Wallnau, R. Seacord and J. Robert, "A Survey of Black-Box Modernization Approaches for Information Systems," in *Proceedings of the International Conference on Software Maintenance (ICSM'00)*, pp. 173, 2000.
- [8] M. Saarelainen, J.J. Ahonen, H. Lintinen, J. Koskinen, I. Kankaanpää, H. Sivula, P. Juutilainen and T. Tilus, "Software modernization and replacement decision making in industry: a qualitative study," in *Proceedings of the 10th international conference on Evaluation and Assessment in Software Engineering*, pp. 12-21, 2006.
- [9] M. Saarelainen, J. Koskinen, J.J. Ahonen, I. Kankaanpää, H. Sivula, H. Lintinen, P. Juutilainen and T. Tilus, "Group Decision-Making Processes in Industrial Software Evolution," in *International Conference on Software Engineering Advances*, 2007. ICSEA 2007. pp. 78, 2007.
- [10] M. Saarelainen, "Why Groups are Used in Software System Modernization Decisions? Comparing Group Decision-Making in Private and Public Sector," in *11th International Conference on Computer and Information Technology (CIT)*, 2011 IEEE, pp. 350-355, 2011.
- [11] J. Koskinen, H. Lintinen, J.J. Ahonen, T. Tilus and H. Sivula, "Empirical study of industrial decision making for software modernizations," in *International Symposium on Empirical Software Engineering*, 2005. pp. 10, 2005.
- [12] Merriam-Webster.com., "definition", 2012, <http://www.merriam-webster.com/dictionary/definition>.
- [13] H. Suonuuti, Guide to Terminology, Helsinki: Tekniikan sanastokeskus ry, 2001, .
- [14] C. Okoli and K. Schabram, "A Guide to Conducting a Systematic Literature Review of Information Systems Research," *Sprouts: Working Papers on Information Systems*, vol. 10, 2010.
- [15] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Tech. Rep. EBSE Technical Report EBSE-2007-01, 2007.
- [16] P. Runeson, M. Host, A. Rainer and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2012, .

- [17] M.D. Myers, *Qualitative Research in Business & Management*, SAGE Publications Ltd, 2009, pp. 296.
- [18] J. Eskola and J. Suoranta, *Johdatus laadulliseen tutkimukseen*, Tampere: Vastapaino, 2000, pp. 266.
- [19] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, I. Caballero and M. Piattini, "Software modernization by recovering Web services from legacy databases," *Journal of Software: Evolution and Process*, vol. 25, pp. 507-533, 2013.
- [20] R. Pérez-Castillo, I.G. de Guzmán, M. Piattini and B. Weber, "Integrating event logs into KDM repositories," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 1095-1102, 2012.
- [21] M. Fernández-Ropero, R. Pérez-Castillo, B. Weber and M. Piattini, "Empirical assessment of business model transformations based on model simulation," in *Proceedings of the 5th international conference on Theory and Practice of Model Transformations*, pp. 137-151, 2012.
- [22] OMG. , "ADM Glossary of Definitions and Terms", 2009,
http://adm.omg.org/ADM_Glossary_Spreadsheet_pdf.pdf.
- [23] Y. Dajsuren, M. Goldstein and D. Moshkovich, "Modernizing legacy software using a System Grokking technology," in *IEEE International Conference on Software Maintenance (ICSM)*, 2010, pp. 1-7, 2010.
- [24] R. Seacord, D. Plakosh and G. Lewis, *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*, Addison-Wesley, Pearson Education, 2003.
- [25] R. Pérez-Castillo, I.G. de Guzmán, O. Avila-García and M. Piattini, "On the Use of ADM to Contextualize Data on Legacy Source Code for Software Modernization," in *Proceedings of the 2009 16th Working Conference on Reverse Engineering*, pp. 128-132, 2009.
- [26] J. Canovas and J. Molina, "An Architecture-Driven Modernization Tool for Calculating Metrics," *IEEE Softw.*, vol. 27, pp. 37-43, 2010.
- [27] R. Pérez-Castillo, I.G. de Guzmán and M. Piattini, "Implementing business process recovery patterns through QVT transformations," in *Proceedings of the Third international conference on Theory and practice of model transformations*, pp. 168-183, 2010.
- [28] R. Pérez-Castillo, I.G. de Guzmán and M. Piattini, "Business process archeology using MARBLE," *Information and Software Technology*, vol. 53, pp. 1023-1044, 2011.
- [29] R. Pérez-Castillo, I.G. de Guzmán and M. Piattini, "Knowledge Discovery Metamodel-ISO/IEC 19506: A standard to modernize legacy systems," *Computer Standards & Interfaces*, vol. 33, pp. 519-532, 2011.
- [30] ISO/IEC 19506, "Information technology - Object Management Group Architecture-Driven Modernization (ADM) - Knowledge Discovery Meta-Model (KDM)," 2012.
- [31] E.S. Cho, J.E. Cha and Y.J. Yang, "MARMI-RE: a method and tools for legacy system modernization," in *Proceedings of the Second international conference on Software Engineering Research, Management and Applications*, pp. 42-57, 2006.
- [32] S. Chung, J.B.C. An and S. Davalos, "Service-Oriented Software Reengineering: SoSR," in *40th Annual Hawaii International Conference on System Sciences*, 2007. HICSS 2007. pp. 172c-172c, 2007.
- [33] G. Canfora, A.R. Fasolino, G. Frattolillo and P. Tramontana, "A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures," *J.Syst.Software*, vol. 81, pp. 463-480, 4. 2008.
- [34] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, M. Piattini and ÁS. Places, "A case study on business process recovery using an e-government system," *Software: Practice and Experience*, vol. 42, pp. 159-189, 2012.
- [35] J. Koskinen, J.J. Ahonen, H. Sivula, T. Tilus, H. Lintinen and I. Kankaanpää, "Software Modernization Decision Criteria: An Empirical Study," in *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering*, pp. 324-331, 2005.
- [36] N. Ilk, J.L. Zhao, P. Goes and P. Hofmann, "Semantic enrichment process: An approach to software component reuse in modernizing enterprise systems," *Inf.Syst.Front.*, vol. 13, pp. 359-370, jul. 2011.
- [37] R. Pérez-Castillo, J.A. Cruz-Lemus, I.G. de Guzmán and M. Piattini, "A family of case studies on business process mining using MARBLE," *J.Syst.Software*, vol. 85, pp. 1370-1385, 2012.
- [38] E. Putrycz and A.W. Kark, "Recovering business rules from legacy source code for system modernization," in *Proceedings of the 2007 international conference on Advances in rule interchange and applications*, pp. 107-118, 2007.
- [39] D.J. Meacham, J.B. Michael, M. Shing and J.M. Voas, "Standards interoperability: Applying software safety

assurance standards to the evolution of legacy software," in IEEE International Conference on System of Systems Engineering, 2009. SoSE 2009. pp. 1-8, 2009.

[40] G. Menkhaus and U. Frei, "Legacy system integration using a grammar-based transformation system," in 26th International Conference on Information Technology Interfaces, 2004. pp. 119-124, 2004.

[41] K. Normantas, S. Sosunovas and O. Vasilecas, "An overview of the knowledge discovery meta-model," in Proceedings of the 13th International Conference on Computer Systems and Technologies, pp. 52-57, 2012.

[42] C.T.H. Everaars, F. Arbab and B. Koren, "Using coordination to restructure sequential source code into a concurrent program," in IEEE International Conference on Software Maintenance, 2001. Proceedings. pp. 342-351, 2001.

[43] C. Chiang and C. Bayrak, "Legacy Software Modernization," in IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06. pp. 1304-1309, 2006.

[44] N. Ilk, J.L. Zhao and P. Hofmann, "On Reuse of Source Code Components in Modernizing Enterprise Systems," in IEEE Symposium on Advanced Management of Information for Globalized Enterprises, 2008. AMIGE 2008. pp. 1-5, 2008.

[45] Z. Li, X. Anming, Z. Naiyue, H. Jianbin and C. Zhong, "A SOA Modernization Method Based on Tollgate Model," in Proceedings of the 2009 International Symposium on Information Engineering and Electronic Commerce, pp. 285-289, 2009.

[46] M. Wimmer and G. Kramler, "Bridging grammarware and modelware," in Proceedings of the 2005 international conference on Satellite Events at the MoDELS, pp. 159-168, 2006.

[47] J. Mazón and J. Trujillo, "A model driven modernization approach for automatically deriving multidimensional models in data warehouses," in Proceedings of the 26th international conference on Conceptual modeling, pp. 56-71, 2007.

[48] S. Chung, J.B.C. An and S. Davalos, "Service-Oriented Software Reengineering: SoSR," in 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007. pp. 172c-172c, 2007.

[49] A. Rodríguez, A. Caro and E. Fernández-Medina, "Towards framework definition to obtain secure business process from legacy information systems," in Proceedings of

the first international workshop on Model driven service engineering and data quality and security, pp. 17-24, 2009.

[50] R. Pérez-Castillo, I.G. de Guzmán, M. Piattini, B. Weber and Á.S. Places, "An empirical comparison of static and dynamic business process mining," in Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 272-279, 2011.

[51] H. Bagheri and K. Sullivan, "Bottom-up model-driven development," in 35th International Conference on Software Engineering (ICSE), 2013, pp. 1221-1224, 2013.

[52] P. Jamshidi, A. Ahmad and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE Transactions on Cloud Computing, vol. 1, pp. 142-157, 2013.

[53] A. Bergmayr, H. Bruneliere, J.L. Canovas Izquierdo, J. Gorronogoitia, G. Kousiouris, D. Kyriazis, P. Langer, A. Menychtas, L. Orue-Echevarria, C. Pezuela and M. Wimmer, "Migrating Legacy Software to the Cloud with ARTIST," in 17th European Conference on Software Maintenance and Reengineering (CSMR), 2013, pp. 465-468, 2013.

[54] G.E. Boussaidi, A.B. Belle, S. Vaucher and H. Mili, "Reconstructing Architectural Views from Legacy Systems," in 19th Working Conference on Reverse Engineering (WCRE), 2012, pp. 345-354, 2012.

[55] G. Chénard, I. Khriss and A. Salah, "Towards the Automatic Discovery of Platform Transformation Templates of Legacy Object-oriented Systems," in Proceedings of the 6th International Workshop on Models and Evolution, pp. 51-56, 2012.

[56] S. Comella-Dorda, K. Wallnau, R.C. Seacord and J. Robert, "A Survey of Legacy System Modernization Approaches," Carnegie Mellon University., Tech. Rep. CMU/SEI-2000-TN-003, 2000.

[57] OMG, "Why do we need standards for the modernization of existing systems?" OMG ADM Task Force, 2003.

[58] The Open Group, "TOGAF® Version 9.1, an Open Group Standard", 2014,
<http://pubs.opengroup.org/architecture/togaf91-doc/arch/index.html>.