Description languages for the lifecycle activities of services in the cloud domain: A systematic mapping protocol

Jorge Bermeo Conto*, *Miguel Zúñiga-Prieto*, *Lizandro Solano-Quinde

Departamento de Ciencias de la Computación, Universidad de Cuenca, Av. 12 de Abril s/n, Cuenca, Ecuador.

Autores para correspondencia: {jorge.bermeo, miguel.zunigap, lizandro.solano}@ucuenca.edu.ec

Fecha de recepción: 4 de junio 2017 - Fecha de aceptación: 25 de agosto 2017

ABSTRACT

Activities of a service life cycle require from developers the systematic reasoning about their related aspects. Specification languages are used in software engineering to improve the quality and delivery time of software systems by offering notations and abstractions that ease the reasoning about different aspects in a domain problem. Among various distinctive approaches, which propose specification languages, we are interested in those that support and enable the analytical reasoning about activities of the service life cycle in the cloud applications development domain. This work presents a protocol for the systematic mapping that provides guidance to gather evidence of specification languages that support the service life cycle activities in a cloud application domain, identify the issues that those specification languages have addressed and gaps in the existing research.

Keywords: Cloud computing, systematic mapping protocol, specification languages, service life cycle.

RESUMEN

Las actividades del ciclo de vida de servicios requieren de los desarrolladores el razonamiento sistemático sobre sus aspectos relacionados. Lenguajes de especificación se utilizan en la ingeniería de software con el fin de mejorar la calidad y el tiempo de entrega de los sistemas de software, ofreciendo notaciones y abstracciones que facilitan el razonamiento sobre los diferentes aspectos en un problema de dominio. Entre diversos enfoques, que proponen lenguajes de especificación, nos interesan los que soportan y permiten el razonamiento analítico sobre las actividades del ciclo de vida de servicios en el dominio de desarrollo de aplicaciones en la nube. Este trabajo presenta un protocolo para el mapeo sistemático que provee una guía para determinar el estado del arte de los lenguajes de especificación que soportan las actividades del ciclo de vida del servicio en el dominio de aplicación en la nube, e identificar los problemas que esos lenguajes de especificación no han abordado.

Palabras clave: Cloud computing, protocolo de mapeo sistemático, lenguajes de especificación, ciclo de vida de servicio.

1. INTRODUCTION

Cloud computing is a business model for delivering IT resources and applications (IT resources) as services that can be accessed remotely on demand and over the Internet (Leavitt, 2009). In this context, the set of data centers, hardware, software, and storage is known as cloud. In this business model, users purchase remote access to IT resources. The difference of this business model with the traditional resource delivery model is that in a traditional model resources are delivered in the form of products sold or licensed to users, and then used locally in their technological infrastructure.

Cloud applications are service-oriented applications that "from the point of view of software engineering is a software provided as a service" (Hamdaqa, Livogiannis, & Tahvildari, 2011). These are distributed applications, usually composed of web services, which consume resources, obtained from cloud providers during their execution. Owners of cloud applications buy cloud resources from cloud providers or compose services shared among organizations, then sell applications that are accessed either through graphical user interfaces or from other applications/services. Unlike traditional software engineering, service-oriented applications require new roles and new development tasks. The service life cycle includes different stages, i.e., design time, runtime and change time, as well as different stakeholders (service provider, application provider - service consumer - and service broker), where each stakeholder has different activities associated depending on the life cycle stage.

The different activities of the service life cycle require from the developers systematic reasoning about their related aspects. Specification languages are used in software engineering to improve the quality and delivery time of software systems by offering notations and abstractions that ease the reasoning about different aspects in a domain problem, helping to express system models. Among various distinctive approaches, which propose specification languages, we are interested in those that support and enable the analytical reasoning about service requirements at the service life cycle activities in the cloud applications development domain.

We are conducting a systematic mapping as to gather evidence of specification languages that support service life cycle activities in a cloud application domain, identify the issues those languages have address, and to identify gaps in the existing research. In this context, it is important to develop a protocol before conducting the mapping, which provides the guide for the systematic mapping to identify, assess and collate evidence (Mendes, 2005). In this paper, we present a protocol for the systematic mapping on specification languages in the service life cycle on the cloud applications domain and, hence, complete the first phase of the systematic mapping. This systematic mapping is aimed at finding all the existing evidence on specification languages proposed to support the life cycle activities for cloud applications and to find evidence of factors tar characterize these languages.

2. RESEARCH METHOD

A systematic mapping study is a formalized and repeatable process; it is a mean of categorizing and summarizing the existing information about a research question in an unbiased manner. A systematic mapping study has three stages (Kitchenham, Dybå, & Jørgensen, 2004; Kitchenham & Charters, 2007): planning, conducting, and reporting. We plan to perform a systematic mapping study considering the guidelines provided by Brereton, Kitchenham, Budgen, Turner, & Khalil (2007), Petersen, Feldt, Mujtaba, & Mattson (2008), and Kitchenham, Dybå, & Jørgensen (2004), where the protocol presented in this work is part of the planning stage. We formulated the mapping protocol based on the systematic literature review guidelines and procedures, proposed by Kitchenham, Dybå, & Jørgensen (2004) as depicted in Figure 1, and explained in the following sections.

Figure 1. Systematic mapping protocol activities.

2.1. Establishment of the research question

To examine the current usage of specification languages for supporting the life cycle activities associated with services in cloud environments, we formulated the following research question: "How description languages are being used by researchers and practitioners to support the life cycle activities of cloud services/applications?". This research question allows: i) to categorize and summarize the current knowledge concerning the usage of specification languages, and ii) to identify gaps in current

research as basis for the suggestion of areas for further investigation. Raising critical questions helps to identify and/or scope future research activities and is according to Kitchenham & Charters (2007) considered the most important aspect of the protocol for a systematic literature review. Therefore, the research question was structured following the PICOC (Petticrew & Roberts, 2006) criteria, as shown in Table 1. Our research focus is not comparison; consequently, it is not included.

Table 1. PICOC (Population, Intervention, Comparison, Outcome and Context) criteria to formulate the research question.

Criteria	Applied Criteria
Population	Cloud services / Cloud Applications
Intervention	Description languages / Specification languages
Outcome	identify improvements/supports in phases of a service life cycle
Context	Researchers, industry practitioners

Since our research question is too broad, to facilitate its addressing, it has been decomposed into more detailed sub-questions. Table 2 shows these research sub-questions along with their motivation.

Table 2. Research sub-questions.

Research sub-questions	Motivation
RQ1: How do specification languages support the life cycle activities of services?	To discover what activities of services' lifecycle are most frequently supported, which activity aspects are being specified, and which stakeholders are generally involved in the specification.
RQ2: Which are the characteristics of the offered specification languages?	To discover characteristics of specification languages frequently offered; language syntax and semantics.
RQ3: How do specification languages support the cloud paradigm?	To discover the delivery service model and the cloud environment used. Also, to find out the proposed infrastructure and if the approach is specific to a supplier.
RQ4: Which software development approaches are supported by the specification languages?	To discover which development approaches are being supported (e.g., agile, model-driven, incremental).

2.2. Definition of the search strategy

The search strategy includes digital libraries and manual search approaches of conference proceedings and journals. The main digital libraries used to search for primary studies are:

- ACM DL Digital Library,
- IEEE Explore Digital Library, and
- Scopus.

As shown in Figure 2, in order to define the query string, we identified the main concepts related to the research question (i.e., cloud, specification, language, service life cycle phases), then we determined alternative terms for those concepts (e.g., model or models for language), and finally we defined the query string to be used to extract primary studies from the selected libraries. Table 3 shows the resulting query string. Following, we planned to apply the query string to the title and abstract of each article for all the sources, and therefore we modified the query to fit the syntax and semantic of each digital library (see Table 3). As to perform a consistent search, these search terms will be also used in the manual search.

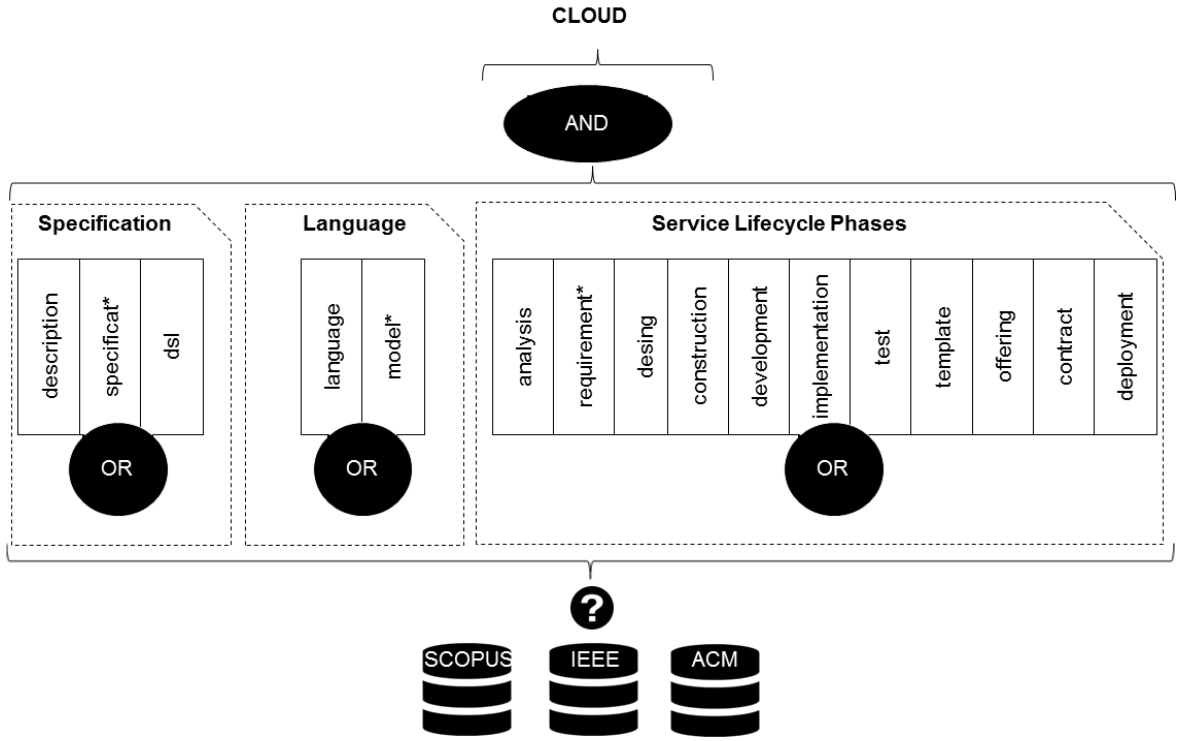

Figure 2. Summary of the query string definition process.

Table 3. Query string for each digital library.

Library	Query string
SCOPUS	*TITLE-ABS ((cloud* AND (description OR specificat* OR dsl) AND (language OR model*) AND (analysis OR requirement* OR design OR construction OR development OR implementation OR test OR template OR offering OR contract OR provisi* OR deployment))) AND PUBYEAR > 2005 AND (LIMIT-TO (SUBJAREA, "COMP"))*
IEEE	*(("Abstract":cloud* AND ("Abstract":description OR "Abstract":specificat* OR "Abstract":dsl) AND ("Abstract":language OR "Abstract":model*) AND (p_Abstract:analysis OR "Abstract":requirement* OR "Abstract":desing OR ...*
	("Document Title":cloud AND ("Document Title":description OR "Document Title":specificat* OR "Document Title":dsl OR)AND ("Document Title":language OR "Document Title":model*) AND ("Document Title":analysis OR "Document ...*
ACM	*recordAbstract:(+(cloud*) + (description specificat* dsl) + (language model*) + (analysis requirement* desing construction development implementation test template offering contract provisi* deployment))*
	acmdlTitle:(+(cloud) + (description specificat* dsl) + (language model*) + (analysis requirement* desing construction development implementation test template offering contract provisi* deployment))*

2.3. Selection of primary studies

The selection of studies will be performed through a multi-step process as shown in Figure3. The period to review included the studies published from 2006 to 2017 (inclusive). The starting year was selected since we wanted to know the influence of "Cloud Computing" on new approaches or proposals for description languages, and in 2006 Amazon Inc. officially launched Amazon Web Services (Amazon Web Services, n.d.).

Figure 3. Stages of selection of primary studies process.

At Stage 2, citations from the Stage 1 will be imported or copied to an Excel sheet. Citations will be sorted by study title, and Excel analysis tools will be used to search for and eliminate duplicates. For each subsequent stage, the Excel sheets will used.

At Stage 3 the articles that are outside the scope of this systematic mapping will be excluded. This is performed by reading the title and abstract of all studies that resulted from Stage 2, which helps to exclude articles with title or abstract that indicated clearly to be outside the scope. Only studies presenting specification languages to support the lifecycle activities associated with services in cloud environments will be selected.

Exclusion criteria: Studies that met at least one of the following criteria will be excluded:

- Papers that do not focus on the cloud domain or do not propose specification languages.
- Papers that are systematic reviews, mapping reviews or introductory papers for special issues, workshops, tutorials, and mini-tracks.
- Papers that are less than four pages or presenting only recommendations, guidelines, or design principles.
- Duplicate reports of the same study in different sources.

At Stage 4, if it is unclear from the title, abstract, and keywords whether a study is a relevant study, a full-text skim will be performed. Discrepancies in the selection will be solved by consensus among the authors of this paper. Studies will be discarded based on de inclusion/exclusion criteria defined in Stage 3.

Finally, in Stage 5, by using resulting primary studies of stage 4, we will proceed to do a reference-harvesting analysis (Littell, Corcoran, & Pillai, 2009) to find out whether we missed a relevant study. Because of this stage, studies for the quality assessment are obtained as detailed in the next section.

2.4. Quality assessment

In addition to general inclusion/exclusion criteria, it is critical to assess the "quality" of primary studies. Thereto, the quality items shown in Table 4 will be used to assess those studies.

Table 4. Description of quality assessment

1. Problem definition of the study (Dybå & Dingsøyr, 2008):
(2) The authors provide an explicit problem description for the study. (1) The authors provide a general problem description. (0) There is no problem description.
2. Context in which the study was carried out (Dybå & Dingsøyr, 2008):
(1) The authors provide an explicit description of the context in which this research was performed (e.g., lab setting, as part of a project, in collaboration with industry, etc.). (0.5) The authors provide some general words about the environment in which this research was performed. (0) There is no description of the environment.

3. Specification support used to deal whit the described problem: (2) The authors present a clear description about how the specification language supports the described problem. (1) The authors provide some general description about how the specification language supports the described problem. (0) There is no a clear description of how the problem is solved.
4. Type of validation conducted: (2) The authors show how their proposals were validated/evaluated, show its application, and presented the results of the validation/evaluation. (1) The authors describe the process of validation/evaluation, but do not show its application. (0) There is no validation.
5. Contributions of the study refer to the study results (Dybå & Dingsøyr, 2008): (2) The authors explicitly list the contributions/results of the study. (1) The authors provide some general words about the study results. (0) There is no description of the research results.
6. Insights derived from the study (Dybå & Dingsøyr, 2008): (2) The authors explicitly list insights/lessons learned from the study. (1) The authors provide some general words about insights/lessons learned from the study. (0) There is no description of the insights derived from the study.
7. Limitations of the study (Dybå & Dingsøyr, 2008). Options are: (2) The authors explicitly list the limitations/problems with the study. (1) The authors provide some general words about limitations/problems with the study. (0) There is no description of the limitations of the study.
8. The study has been published in a relevant journal or conference proceedings: (2) Very relevant (CORE conference ranking A, or Journal Citation Reports Tercile T1). (1) Relevant (CORE conference ranking B, or Journal Citation Reports Tercile T2). (0) Not so relevant (CORE conference ranking C, or Journal Citation Reports Tercile T3).

The addition of the eight items used to assess study quality (Table 4) will provide a final score (max 16) calculated by adding up the scores for all items (scores for the various options are given between brackets). These scores will not be used to exclude papers from the systematic mapping study, but rather to detect representative studies that discuss each research sub-question.

2.5. Definition of the data extraction strategy

The data extraction strategy will be based on providing the set of possible answers for each research sub-question that was defined. This strategy ensures the application of the same extraction data criteria to all selected papers and it facilitates their classification. The possible answers to each research sub-question are explained (see Table 5) in more detail in the following.

With regard to **RQ1,** a paper can be classified in one of the following answers:

- ***EC1.*** *Lifecycle phase for which the approach offers a solution* (Gu & Lago, 2007). The service life cycle model is presented with the separation of design time and runtime. The design time processes include (Wall, 2006a): Requirements engineering, Business modeling, Service Design, Service Development, Services testing and Service implementation. The runtime processes include (Wall, 2006b): Service publishing, Service provision, Service monitoring, Service discovery, Service orchestration /composition, Service Negotiation, Service invocation, Application testing and Service monitoring
- ***EC2.*** *Aspects of the service lifecycle activity specified*. The aspects of the service lifecycle are functional and non-functional. Clearly in the area of cloud computing, non-functional

requirements are of significant relevance for clients of cloud providers. In this aspect, modeling capabilities play an important role as they allow a structured representation of such requirements. While the spectrum of non-functional requirements is broad, pricing is one aspect that is addressed by several approaches from different perspectives (Bermayr, Grossniklaus, & Wimmer, 2013). In this sense, we have: Functional, Non-functional, Service level agreement, Architecture descriptions, Service descriptions, Implementation descriptions, Deployment descriptions, Execution environments, Node characteristics, Constrains and Pricing.

- ***EC3.*** *Role to whom the language is expected to provide support*. Service Lifecycle phase are associated with a stakeholder (Gu & Lago, 2007; INFAVRIO, n.d.): One of the design principles of service-oriented applications is to decouple the role of service provider and service consumer (Erl, 2008). In this context, when changes occur at one role, the less dependencies between these two roles, the less influence on another role. Service provider, service consumer, and service broker are three architectural roles (stakeholders) in service-oriented applications (Gu & Lago, 2007).

With regard to **RQ2**, a paper can be classified in one of the following answers:

- ***EC4.*** *Characteristics of a modeling language*. *Can be summarized as follows* (Bjørner, 2010; Bezivin, 2005): Abstract Syntax, Concrete Syntax, and Semantics. For Abstract Syntax we have: XML Shema, UML Library, UML Profiles, Ecore and Grammar.
- ***EC5.*** *Concrete Syntax*. We have according to Bjørner (2010) and Bezivin (2005): Graphical, Graphical (UML-based), Graphical (Cloud MIGXpress), Textual (JSON-based), Textual (XML-based), Textual (YAML-based), Graphical + Textual (OVFbased), Graphical (UMLbased), Textual (XMLbased) and Textual (XMLbased) + Graphical (VINO4Tosca).
- ***EC6.*** Semantics. Bjørner (2010) and Bezivin (2005) stated: English Prose, Mapping to TOSCA, Deployment Optimizer, Conformance Checker, Provisioning Engine, OpenStack, Deployment Optimizer, Deployment Engine, Open Nebula and Open Tosca.

With regard to **RQ3**, a paper can be classified in one of the following answers:

- ***EC7.*** *Service delivery model.* There are three basic service delivery models: i) *Infrastructure as a service* (IaaS), the basic computing capability, e.g., storage, processing, network, is delivered as the standardized services over the network. ii) *Platform as a service* (PaaS), services at this layer refer to a development environment where developers can build and run an application by using prebuilt components and interfaces that particular platform provides as a service. And iii) *Software as a service* (SaaS), software applications are delivered as services at this layer. Finally, Hybrid, that enables match the IaaS, PaaS or SaaS.
- ***EC8.*** Cloud environment (Bergmayr, Wimmer, Kappel, & Grossniklaus, 2014; Vaquero, Rodero-Merino, & Buyya, 2011). In cloud computing, resources, such as processing power and storage, platforms, and software, are viewed as commodities that are readily available from large data centers operated by cloud providers. Cloud computing leverages service-oriented architectures to unify elements of distributed, grid, utility, and autonomous computing. There are four elements to consider: i) *Dynamic provisioning* of resources offered by a cloud provider as a service; ii) *Pay-as-you-go* consumers can acquire and release such cloud resources on demand and pay only for what they actual consumed, it offers benefits both the cloud consumer and the cloud provider. iii) *Elastically scale* from the consumer perspective, the risk of under- or over-provisioning is avoided as the provisioned cloud resources can match with the consumer's demand (Vaquero, Rodero-Merino, & Buyya, 2011). In contrast, the cloud provider profits from an economy of scale and can offer cloud resources at a price that is lower than the one of an on-premise solution (Walker, 2009). and iv) *Quality of service* refers to the quality of a technical service, which can be expressed in terms of latency, availability and security (Venters & Whitley, 2012).
- ***EC9.*** Deployment Model: The computing infrastructure that delivers these services can be shared (Sun, Dong, & Ashraf, 2012). There are four major cloud deployment models: i) *Public cloud* - The cloud infrastructure is made available to the public or a large industry group. ii) *Private cloud* - The cloud infrastructure is operated solely for an organization. iii) *Hybrid*

cloud - The cloud infrastructure is a composition of two or more clouds that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability; and iv) *Community cloud* - The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns.

- ***EC10.*** Specific supplier. A proposal may or may not be linked to specific supplier such as: Google, Amazon and Azure.

With regard to **RQ4**, a paper can be classified in one of the following answers:

- ***EC11.*** Model driven development approach. A model-driven development approach helps in solving the problem of heterogeneity of technologies and integration; therefore, we are interested in knowing if the primary studies proposes solutions to support this approach.
- ***EC12.*** Incremental development approach. Service-oriented applications are usually developed in an incremental fashion by building reusable services that may interoperate with each other. In this context, we are interested in knowing if the primary studies propose solutions to support an incremental development approach.

Table 5. Data extraction strategy.

RQ1: How specification languages support the life cycle activities of services?			
EC1	Lifecycle phase for which the approach offers a solution.	Requirements engineering	☐
		Business modeling	☐
		Service design	☐
		Service development	☐
		Services testing	☐
		Service implementation	☐
		Service publishing	☐
		Service provision	☐
		Service monitoring	☐
		Service discovery	☐
		Service orchestration /composition	☐
		Service negotiation	☐
		Service invocation	☐
		Application testing	☐
		Service monitoring	☐
EC2	Aspects of the service lifecycle activity specified.	Functional	☐
		Non-functional	☐
		Service level agreement	☐
		Architecture descriptions	☐
		Service descriptions	☐
		Implementation descriptions	☐
		Deployment descriptions	☐
		Execution environments	☐
		Node characteristics	☐
		Constrains	☐
		Pricing	☐
		Others: _____	☐
EC3	Role to whom the language is expected to provide support.	Service provider	☐
		Service consumer	☐
		Service broker	☐
RQ2: Which are the characteristics of the offered specification languages?			
EC4	Abstract Syntax.	XML schema	☐
		UML library	☐
		UML profiles	☐
		Ecore	☐
		Grammar	☐
		Others: _____	☐

EC5	Concrete syntax.	Graphical	☐
		Graphical (UML-based)	☐
		Graphical (Cloud MIGXpress)	☐
		Textual (JSON-based)	☐
		Textual (XML-based)	☐
		Textual (YAML-based)	☐
		Graphical + Textual (OVFbased)	☐
		Graphical (UMLbased)	☐
		Textual (XMLbased)	☐
		Textual (XMLbased) + Graphical	☐
		Others: _____	☐
EC6	Semantics.	English prose	☐
		Mapping to TOSCA	☐
		Deployment optimizer	☐
		Conformance checker	☐
		Provisioning engine	☐
		OpenStack	☐
		Deployment optimizer	☐
		Deployment engine	☐
		Open Nebula	☐
		Open Tosca	☐
		Others: _____	☐
RQ3: How specification languages support the cloud paradigm?			
EC7	Service delivery model.	IaaS	O
		PaaS	O
		SaaS	O
		Hybrid: _____	O
EC8	Cloud environment feature supported.	Dynamic provisioning	☐
		Pay-as-you-go principle	☐
		Elastically scale	☐
		Quality of service.	☐
EC9	Deployment model.	Private	O
		Community	O
		Public	O
		Hybrid	O
EC10	Supplier.	Google	O
		Amazon	O
		Azure	O
		Supplier independent	O
		Other: _____	O
RQ4: Which software development approaches are supported by the specification languages?			
EC11	Software engineering paradigm is MDA.	Yes	O
		No	O
EC12	Incremental development approach.	Yes	O
		No	O

2.6. Selection of synthesis methods

We are going to apply both quantitative and qualitative synthesis methods. The quantitative synthesis is based on:

- Counting the primary studies that are classified in each answer from our research sub-questions.
- Defining bubble plots to report the frequencies of combining the results from different research sub-questions. A bubble plot is basically two x - y scatter plots with bubbles in category intersections. This synthesis method is useful to provide a map and giving a quick overview of a research field (Petersen, Feldt, Mujtaba, & Mattson, 2008).
- Counting the number of papers found in each bibliographic source per year.
- The qualitative synthesis is based on: Including several representative studies for each research sub-question by considering the results from the quality assessment.

ACKNOWLEDGEMENTS

This research was supported by the DIUC_XIV_2016_038 project.

REFERENCES

Amazon Web Services. (n.d.). *What is cloud computing?*. Available at https://aws.amazon.com/what-is-cloud-computing/

Bergmayr, A., Wimmer, M., Kappel, G., Grossniklaus, M. (2014). *Cloud modeling languages by example*. IEEE 7th International Conference on Service-Oriented Computing and Applications. https://doi.org/10.1109/SOCA.2014.56

Bermayr, A., Grossniklaus, M., Wimmer, M. (Eds.) (2013). *D9.1 State of the art in modelling languages and model transformation techniques*. ARTIST. Technische Universität Wien (FP7-317859), 61 p. Available at http://www.artist-project.eu/sites/default/files/D9.1%20SOTA%20in%20modeling%20languages%20and%20model%20transformationtechniques_M6_31032013.pdf

Bezivin, J. (2005). On the unification power of models. *Software and System Modeling*, *4*(2), 171-188. https://doi.org/10.1007/s10270-005-0079-0

Bjørner, D. (2010). *Software Engineering 1: Abstraction and Modeling*.Software Engineering, Springer.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *The Journal of Systems and Software*, *80*(4), 571-583. https://doi.org/10.1016/j.jss.2006.07.009

Dybå, T., Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859. https://doi.org/10.1016/j.infsof.2008.01.006

Erl, T. (2008). *SOA: Principles of service design*. Prentice Hall Service, 106 p.

Gu, Q., Lago, P. (2007). *A stakeholder-driven service life cycle model for SOA*. IW-SOSWE '07 2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting, 7 p.

Hamdaqa, M., Livogiannis, T., Tahvildari, L. (2011). *A eference odel for developing cloud applications.* Proceedings of the 1st International Conference on Cloud Computing and Services Science, 98-103. https://doi.org/10.5220/0003393800980103

Infravio (n.d.). *The definitive guide to SOA governance and lifecycle management*. 49 p. Infravio Resource Center. Available at http://embedded-computing.com/news/infravio-guide-soa-governance-lifecycle-management/

Kitchenham, B. A., Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering, Version 2.3*. EBSE Technical Report, Keele University, Germany.

Kitchenham, B. A., Dybå, T., Jørgensen, M. (2004). *Evidence-based software engineering*. Proceedings of 27th IEEE International Software Engineering Conference, IEEE Computer Society, pp. 273-281.

Leavitt, N. (2009). Is cloud computing really ready for prime time? *Computer*, *42*(1). https://doi.org/10.1109/MC.2009.20

Littell, J. H., Corcoran, J., Pillai, V. (2009). *Systematic reviews*. The handbook of social work research methods. Oxford University Press, Inc. 202 p.

Mendes, E. (2005). *A systematic review of web engineering research.* International Symposium on Empirical Software Engineering, pp. 498-507. https://doi.org/10.1109/ISESE.2005.1541857

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M. (2008). *Systematic mapping studies in software engineering*. 12th International Conference on Evaluation and Assessment in Software Engineering, 17, pp. 1.

Petticrew, M., Roberts, H. (2006). *Systematic reviews in the social sciences: A practical guide*. 336 pp. Oxford, UK: Blackwell Publ.

Sun, L., Dong, H., Ashraf, J. (2012). Survey of service description languages and their issues in cloud computing. *8th International Conference on Semantics, Knowledge and Grids*, 128-135. https://doi.org/10.1109/SKG.2012.49

Vaquero, L., Rodero-Merino, L., Buyya, R. (2011). Dynamically scaling applications in the cloud. *ACM SIGCOMM Computer Communication Review*, *41*(1), 45-52. https://doi.org/ 10.1145/1925861.1925869

Venters, W., Whitley, E. A. (2012). A critical review of cloud computing: researching desires and realities. *Journal of Information Technology*, *27*(3), 179-197.

Walker., E. (2009). The real cost of a CPU hour. *Computer*, *42*(4). https://doi.org/ 10.1109/MC.2009.135

Wall, Q. (2006a). *Understanding the service lifecycle within a SOA: Design time*. Dev2Dev at dev2dev.bea.com/pub/a/2006/08/

Wall, Q. (2006b). *Understanding the service lifecycle within a SOA: Run time*. Dev2Dev at dev2dev.bea.com/pub/a/2006/11/