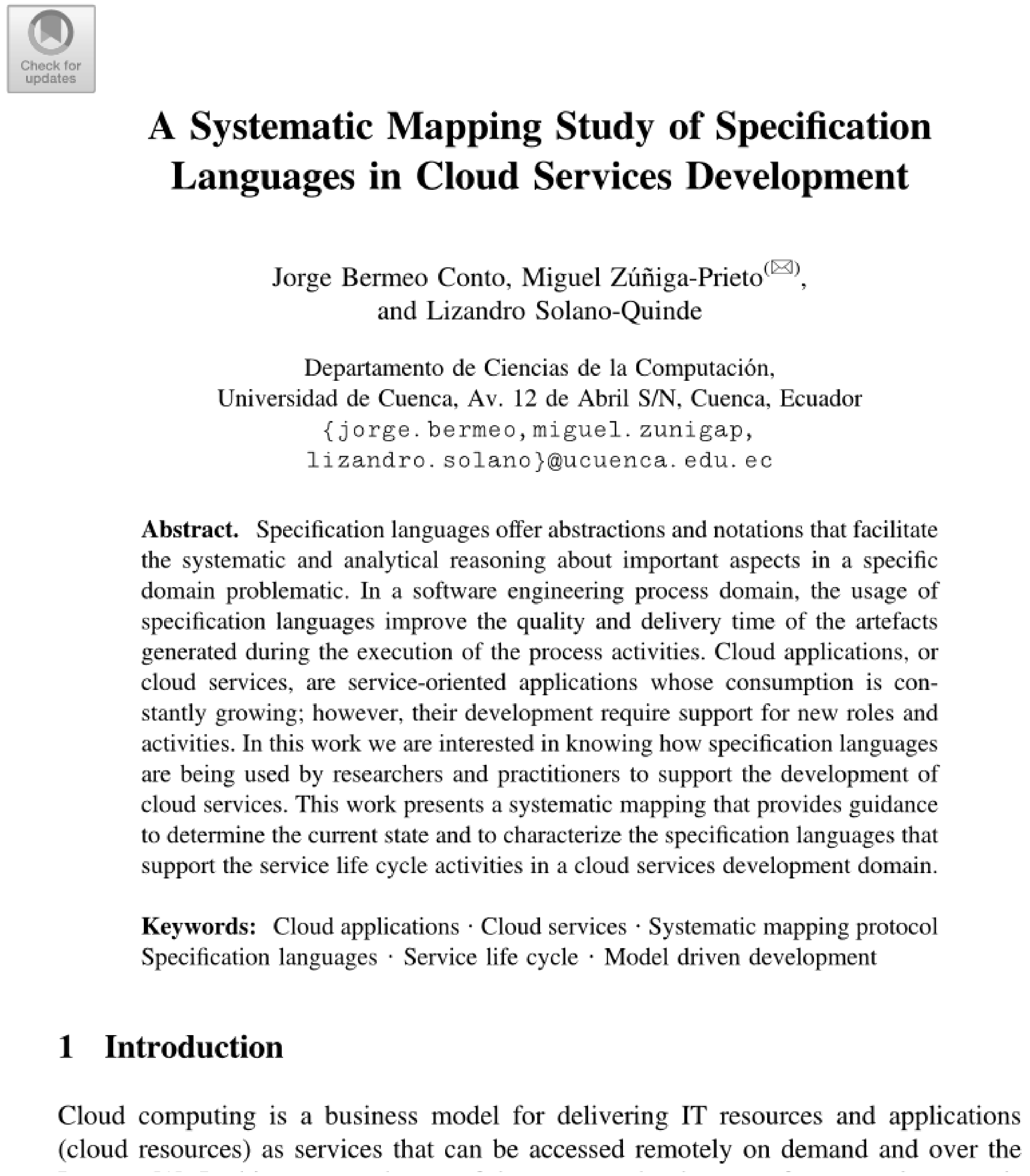
Series ISSN
1865-0929
Edition Number
1
Number of Pages
XXII, 709
Number of Illustrations
83 b/w illustrations, 235 illustrations in colour
Topics
Software Engineering / Programming and Operating Systems

CLOSE

PAGE 1



A Systematic Mapping Study of Specification Languages in Cloud Services Development

Jorge Bermeo Conto, Miguel Zúñiga-Prieto(✉), and Lizandro Solano-Quinde

Departamento de Ciencias de la Computación,
Universidad de Cuenca, Av. 12 de Abril S/N, Cuenca, Ecuador
{jorge.bermeo,miguel.zunigap,
lizandro.solano}@ucuenca.edu.ec

Abstract. Specification languages offer abstractions and notations that facilitate the systematic and analytical reasoning about important aspects in a specific domain problematic. In a software engineering process domain, the usage of specification languages improve the quality and delivery time of the artefacts generated during the execution of the process activities. Cloud applications, or cloud services, are service-oriented applications whose consumption is constantly growing; however, their development require support for new roles and activities. In this work we are interested in knowing how specification languages are being used by researchers and practitioners to support the development of cloud services. This work presents a systematic mapping that provides guidance to determine the current state and to characterize the specification languages that support the service life cycle activities in a cloud services development domain.

Keywords: Cloud applications · Cloud services · Systematic mapping protocol Specification languages · Service life cycle · Model driven development

1 Introduction

Cloud computing is a business model for delivering IT resources and applications (cloud resources) as services that can be accessed remotely on demand and over the

Internet [1]. In this context, the set of data centers, hardware, software, and storage is known as cloud. 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; whereas in cloud computing users purchase remote access to cloud resources.

Cloud applications are service-oriented applications that from a software engineering perspective are software provided as services [2]. These are distributed applications, usually composed of web services, which consume resources obtained from cloud providers during their execution. 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, run-time 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.

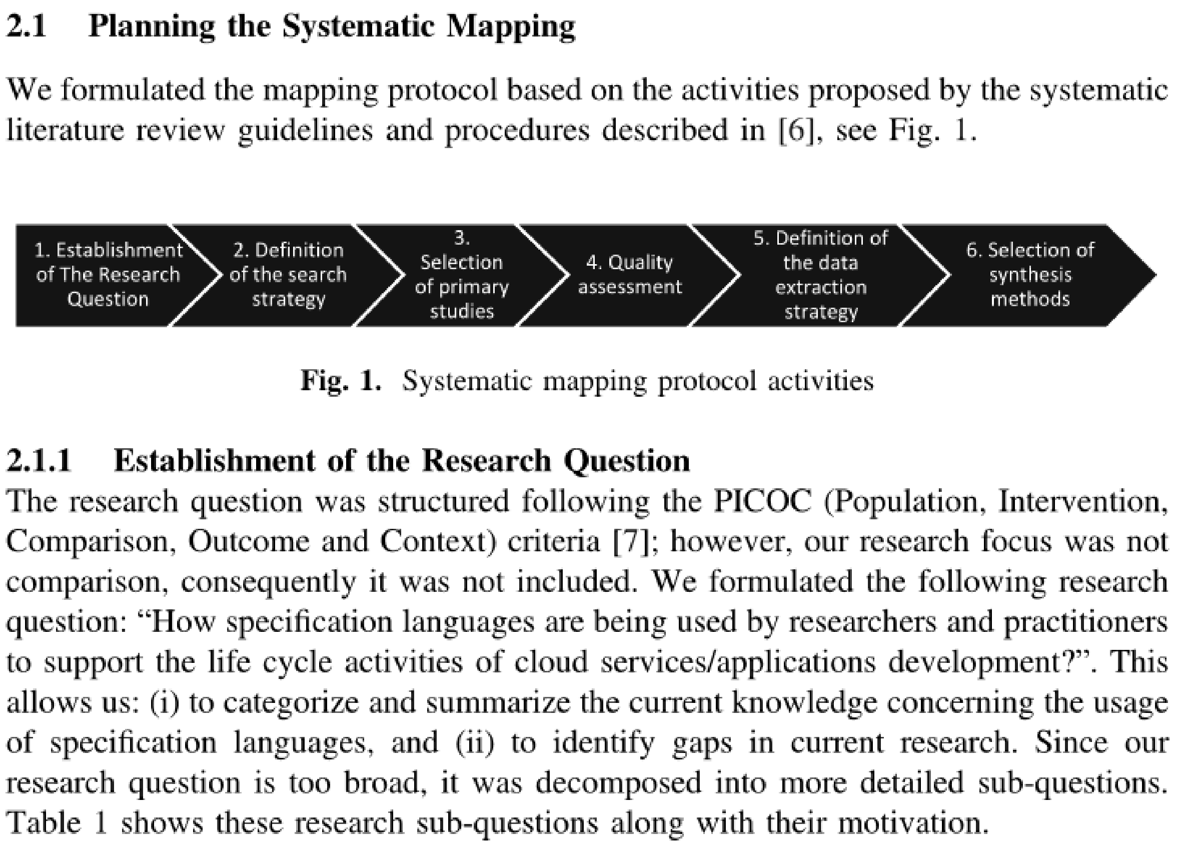
M. Botto-Tobar et al. (Eds.): CITT 2018, CCIS 895, pp. 72–88, 2019.
https://doi.org/10.1007/978-3-030-05532-5_6

The different activities of the service life cycle require developers to systematically reason 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.

In this paper we conduct a systematic mapping in order to gather evidence and characterize specification languages that support the service life cycle activities in a cloud application domain, to identify the issues languages addressed by those languages, and to identify gaps in the existing research. This systematic mapping was based on a mapping protocol that we have already designed in [20], which was extended to improve its accuracy. To the best of our knowledge, this is the first systematic mapping with this objective.

2 Research Method

A systematic mapping study is a formalized and repeatable process that provides guidance for categorizing and summarizing existing information about a research question in an unbiased way. A systematic mapping study has three stages [3]: Planning, Conducting, and Reporting. We plan to perform a systematic mapping study by taking into account guidelines provided by [4–6].



2.1 Planning the Systematic Mapping

We formulated the mapping protocol based on the activities proposed by the systematic literature review guidelines and procedures described in [6], see Fig. 1.

1. Establishment of The Research Question
2. Definition of the search strategy
3. Selection of primary studies
4. Quality assessment
5. Definition of the data extraction strategy
6. Selection of synthesis methods

Fig. 1. Systematic mapping protocol activities

2.1.1 Establishment of the Research Question

The research question was structured following the PICOC (Population, Intervention, Comparison, Outcome and Context) criteria [7]; however, our research focus was not comparison, consequently it was not included. We formulated the following research question: "How specification languages are being used by researchers and practitioners to support the life cycle activities of cloud services/applications development?". This allows us: (i) to categorize and summarize the current knowledge concerning the usage of specification languages, and (ii) to identify gaps in current research. Since our research question is too broad, it was decomposed into more detailed sub-questions. Table 1 shows these research sub-questions along with their motivation.