Future Generation Computer Systems 69 (2017) 41–65

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A Reference Architecture for provisioning of Tools as a Service: Meta-model, Ontologies and Design Elements

Muhammad Aufeef Chauhan [a,*], Muhammad Ali Babar [a,b], Quan Z. Sheng [b]

[a] Software and Systems Section, IT University of Copenhagen, Copenhagen, Denmark
[b] The University of Adelaide, Adelaide, Australia

ARTICLE INFO

Article history:
Received 27 March 2016
Received in revised form
8 November 2016
Accepted 2 December 2016
Available online 6 December 2016

Keywords:
Cloud computing
Software reference architecture
Tools as a Service (TaaS)
Meta-model
Ontologies

ABSTRACT

Software Architecture (SA) plays a critical role in designing, developing and evolving cloud-based platforms that can be used to provision different types of services for consumers on demand. In this paper, we present a Reference Architecture (RA) for designing cloud-based Tools as a service SPACE (TSPACE), which can provision a bundled suite of tools following the Software as a Service (SaaS) model. The reference architecture has been designed by leveraging information structuring approaches and by using well-known architecture design principles and patterns. The RA has been documented using view-based approach and has been presented in terms of its context, goals, the RA meta-model, information structuring and relationship models using ontologies and components of the RA. We have demonstrated the feasibility and applicability of the RA with the help of a prototype and have used the prototype to provision software architecting tools. We have also evaluated the RA in terms of effectiveness of the design decisions and the RA's completeness and feasibility using scenario-based architecture evaluation method. The proposed TSPACE RA can provide valuable insights to information structure approaches and guidelines for designing and implementing TSPACE for various domains.

1. Introduction

Provisioning of tools in a Tools as a service SPACE (TSPACE) instance and providing support for the different activities and tasks during lifecycle of a TSPACE instance is not trivial. TSPACE can consist of a number of tools that can be used to perform various activities related to software architecting. To provision the tools for the end users, TSPACE not only requires facilitating the selection and provisioning of the tools but also needs to provide seamless operations of the tools in terms of distribution of the activities over various tools and integration among the artifacts that are generated and maintained by the tools. Multiple vendors using different technology paradigms and using different programming languages can provide the tools to be provisioned by TSPACE. For example, majority of the tools that are used for architecture modeling such as Microsoft Visio[1] and ArgoUML[2] are developed on top of desktop-based paradigm. The desktop and cloud-based word processing tools (e.g., Microsoft Office Suite[3] and Google Docs[4]) and specialized Web based applications (e.g., PakMe [1]) can be used for architecture documentation (architecture scenario description, architecture significant requirements elicitation and architecture design decisions documentation). Heterogeneous technological paradigms and involvement of multiple vendors highlight the importance of having a gluing mechanism that can facilitate the selection of appropriate tools from a pool of available tools and can support a seamless integration among the selected tools. Involvement of the heterogeneous tools requires a solution that is applicable and extendable for various types of the tools, irrespective of the technological paradigms and the tools' vendors.

We have leveraged semantic integration technologies for addressing the abovementioned challenges of hosting and provisioning tools as services. We have proposed ontologies for TSPACE. The use of ontologies in a specific domain can provide a powerful mechanism to semantically relate unstructured information [2].

* Corresponding author.
E-mail addresses: muac@itu.dk (M.A. Chauhan), ali.babar@adelaide.edu.au (M.A. Babar), michael.sheng@adelaide.edu.au (Q.Z. Sheng).

1 www.microsoftstore.com/Visio.
2 http://en.wikipedia.org/wiki/ArgoUML.
3 www.microsoft.com/Office.
4 docs.google.com.

http://dx.doi.org/10.1016/j.future.2016.12.002
0167-739X/

Ontologies also facilitate communication, integration and reasoning [2]. Our ontology-based solution enables the provisioning of Tools as a Service (TaaS) for performing different activities using appropriate tools hosted on clouds without requiring the individual tools to focus on how to relate the artifacts and data across multiple tools. The users (stakeholders) can choose a set of tools to perform specific activities using the selected tools. The selection of the relevant tools can be based on a number of reasons including but not limited to organizational policies, stakeholders' preferences for the tools, the tasks and the activities related to the projects that are to be performed using the tools, and process requirements of the projects. Restricting stakeholders to a specific set of tools is not a viable solution for performing complex activities. If the projects' stakeholders have the flexibility to choose from a set of tools, the provisioning mechanism needs to provide a flexible way to support tools selection from the set of tools according to the desired needs as well as to provide inter tool integration so that the artifacts that are produced or consumed in one tool can be related/integrated with the artifacts that are being maintained in other tools. The integration mechanism should also provide a support for additional collaboration and awareness activities among the users who perform the activities using the different tools.

Our proposed ontologies provide solution to three main lifecycle phases of the TSPACE. Firstly, the solution supports selection of the tools that are to be provisioned as part of the TSPACE. Once TSPACE is enacted, the solution provides support for semantic integration among the heterogeneous artifacts that are produced and maintained using different tools. Finally, the solution provides support for awareness of the activities that are performed by the stakeholders using the different tools. The awareness mechanism encompasses the activities that are performed on the semantically related artifacts and any conflicts that can occur as a result of the activities. However, as software architecting is a highly complex domain, our proposed approach can only partially automate the conflict identification mechanisms by identifying the potential areas of conflicts. The stakeholders working on the artifacts using different tools have to make the final decisions. The main contributions of the research reported in this article are:

- The TSPACE ontologies that can be used to capture concepts of TSPACE, including Capability Ontology, Tools and Artifacts Ontology, Change Ontology and Annotation Ontology.
- A meta-model to characterize TSPACE and to design concrete architecture for providing TSPACE, and the structure of a set of ontologies that formalizes the tools selection, tools provisioning and semantic integration among the artifacts consumed or generated by the hosted tools.
- A detailed description of the TSPACE Reference Architecture (RA) by using multiple levels of abstractions [3] and rationalizing the incorporation of different modules and components in the RA that are described in terms of development view, logical view, process view and deployment view as recommended by view based approaches [3].
- A detailed description of the use of well-known design principles and architectural patterns [4] for designing and reasoning architectures for TSPACE and a selected set of potential solutions to implement the RA.

The organization of the remainder of this paper is as follows. Section 2 describes the TSPACE RA requirements and documentation approach. Section 3 provides details on the TSPACE RA development approach and architecture meta-model. Section 4 elaborates the TSPACE ontologies and Section 5 provides details on the TSPACE RA design. Section 6 presents an overview of TSPACE prototype along with details on evaluation. Section 7 discusses related work and Section 8 concludes the paper.

2. TSPACE RA requirements and documentation approach

This section presents a brief background, functional and non-functional requirements of the TSPACE RA. Our research on TSPACE has been motivated by the need to provide a workspace where all the required tools can be bundled in a tools suite and provisioned as a service. The TSPACE purports to enable user(s) to have on demand provisioning of tools and semantically integrated artifacts in a Just-in-Time (JIT) fashion. The functional requirements are the functionalities that should be supported and the non-functional requirements are the quality attributes that should be achieved by the design of a TSPACE RA. The reported requirements are based on our previous work on a TaaS infrastructure [5] and a review of the literature on important quality characteristics of the cloud-based systems [6].

2.1. Functional requirements

We have identified the Functional Requirements (FRs) based on the key features required by the RA according to different lifecycle phases of a TSPACE, i.e., tools enactment and provisioning, semantic integration among the artifacts associated with tools after enactment and awareness of the stakeholders' activities during tools' lifecycle. Following are the functional requirements that have been enhanced based on our earlier work [5] in this line of research.

- *FR1—Enactment and provisioning of a TSPACE and associated tools according to the requirements of different activities of a project*: While provisioning tools, the architectural support should also consider the specific location and resource requirements' parameters of the tools.
- *FR2—Semantic integration among artifacts maintained by the tools constituting a TSPACE after enactment*: The TSPACE consists of multiple tools that may have their proprietary formats to store artifacts. The TSPACE architecture should support semantic integration among artifacts generated and maintained by the different tools.
- *FR3—Process centric integration to support collaboration among the tools*: The tools provisioned by TSPACE can also require alignment with organizational process and required support for process centric collaboration among the tools so that artifacts can be exchanged by the tools following specific processes.
- *FR4—Awareness of the operations that are performed on the artifacts during the lifecycle of a TSPACE instance using multiple tools*: Multiple artifacts are produced or consumed during the lifecycle of a specific project for which a TSPACE is instantiated. Hence, there is a need to raise awareness about users' activities associated with the operations that are performed on the artifacts.

2.2. Quality requirements

The quality (i.e., non-functional) requirements of a system are classified into two categories: (i) design time requirements that are discernable while a system is being designed and (ii) runtime requirements that are discernable once a system is operational [7,8]. The following are the design time and runtime quality requirements for a TSPACE:

- *QR1—Automated Provisioning*: A RA shall support automated provisioning of a TSPACE so that the required tools can be acquired automatically for a project based on the constraints on the location of the tools.
- *QR2—Flexibility*: As selection of the tools in a specific instance of a TSPACE depends upon the activities to be performed within a project, a RA shall be flexible enough to provide semantic integration, awareness and traceability support for different types of the tools.