Information and Software Technology 54 (2012) 417–431

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

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

A framework for analysis and design of software reference architectures

Samuil Angelov [a,*], Paul Grefen [a], Danny Greefhorst [b]

[a] School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[b] ArchiXL, Nijverheidsweg Noord 60-27, 3812 PM Amersfoort, The Netherlands

ARTICLE INFO

Article history:
Received 23 October 2010
Received in revised form 29 November 2011
Accepted 29 November 2011
Available online 7 December 2011

Keywords:
Software reference architecture
Software domain architecture
Software architecture design
Software product line architecture

ABSTRACT

Context: A software reference architecture is a generic architecture for a class of systems that is used as a foundation for the design of concrete architectures from this class. The generic nature of reference architectures leads to a less defined architecture design and application contexts, which makes the architecture goal definition and architecture design non-trivial steps, rooted in uncertainty.
Objective: The paper presents a structured and comprehensive study on the congruence between context, goals, and design of software reference architectures. It proposes a tool for the design of congruent reference architectures and for the analysis of the level of congruence of existing reference architectures.
Method: We define a framework for congruent reference architectures. The framework is based on state of the art results from literature and practice. We validate our framework and its quality as analytical tool by applying it for the analysis of 24 reference architectures. The conclusions from our analysis are compared to the opinions of experts on these reference architectures documented in literature and dedicated communication.
Results: Our framework consists of a multi-dimensional classification space and of five types of reference architectures that are formed by combining specific values from the multi-dimensional classification space. Reference architectures that can be classified in one of these types have better chances to become a success. The validation of our framework confirms its quality as a tool for the analysis of the congruence of software reference architectures.
Conclusion: This paper facilitates software architects and scientists in the inception, design, and application of congruent software reference architectures. The application of the tool improves the chance for success of a reference architecture.

1. Introduction

Software reference architectures have emerged as abstractions of concrete software architectures from a certain domain. A reference architecture (RA)[1] is used for the design of concrete architectures in multiple contexts serving as an inspiration or standardization tool [1,2]. Nowadays, the increasing complexity of software, the need for efficient and effective software design processes and for high levels of system interoperability lead to an increasing role of reference architectures in the software design process.

Concrete software architectures are designed in a specific context and reflect concrete business goals of the stakeholders. The types of possible goals, the identification of the stakeholders, the identification of the required system functionalities and qualities, and their effects on the architecture design have been well-studied and extensively published (see, e.g., [3,4]). Reference architectures, however, are designed to facilitate system design and development in multiple projects. Consequently, their design and application take place in a broader and, hence, less-defined context with a larger and less-defined stakeholder base. These less defined architecture design and application contexts and stakeholders, affect the ability of the architecture sponsors[2] and designers for clear judgment and decision making when articulating the architecture goals and elaborating the architecture design [3,4].

The reference architecture design and application contexts, goals, and design exist in a complex relation (see Fig. 1). The architecture goals set constraints on the context in which the architecture should be defined and on the architecture design. Vice versa, the context and design affect the achievement of the architecture goals. Furthermore, the design choices are affected by the context and vice versa, a design choice implies a certain context. In this

* Corresponding author. Address: Information Systems Group, School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Tel.: +31 40 247 2617; fax: +31 40 243 2612.
E-mail addresses: s.angelov@tue.nl (S. Angelov), p.w.p.j.grefen@tue.nl (P. Grefen), dgreefhorst@archixl.nl (D. Greefhorst).

[1] Throughout the paper, we use the term "reference architecture" to refer to the documented description of a software reference architecture.

[2] The stakeholders initiating the design of the architecture.

0950-5849/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2011.11.009

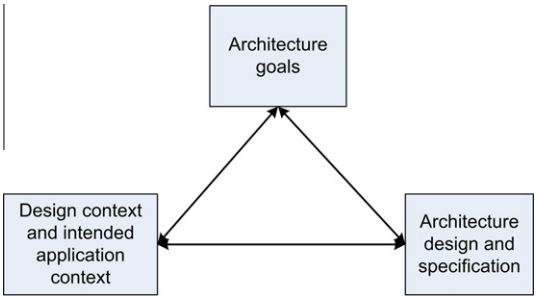

Fig. 1. The relationships between architecture goals, design and contexts.

work, we investigate these relationships and the ways they influence the success of a reference architecture (where by success, we mean the acceptance of the architecture by its stakeholders and its usage in multiple projects).

We call a reference architecture "congruent" if its goals are relevant for the context of the reference architecture and its design properly reflects both the architecture context and goals, i.e., we require congruence between context, goals, and design of a reference architecture. To address the problem of designing congruent reference architectures, in this paper, we propose a "framework for congruent reference architectures". The framework consists of two elements, i.e., a multi-dimensional architecture classification space and a set of predefined reference architecture types (and variants of these types). The multi-dimensional space allows classification of reference architectures according to their context, goals, and design. The reference architecture types are specific combinations of values from the multi-dimensional space. Reference architectures that fit into one of these types are congruent and, as we show in this work, have higher chances for success. Lack of a fit into a type indicates an incongruent architecture and is a portent for the architecture weak usage and criticism from the architecture stakeholders.

The framework that we propose serves multiple purposes. It can be applied for the analysis of the congruence of existing reference architectures. Furthermore, it can be used as a "tool" for the design of congruent reference architectures as well as for the re-design of existing reference architectures into congruent reference architectures.

The work presented in this paper is a follow-up of the work presented in [5] – it substantially extends and improves the results presented there. In the current paper, we include a comprehensive discussion on the concept of reference architecture to better define it and provide a clear disambiguation with respect to similar concepts. We present a more detailed and precise (and, hence, more operational) description of the framework for classifying reference architectures. Furthermore, we provide a description of the framework usage. The validation of the framework is tentative in earlier work – in the current work it is performed in a more profound way and the set of reference architectures used in the validation is extended.

The structure of this paper is as follows. The lack of an accepted definition for the term "reference architecture" by the research community has led to a semantic overloading of the term. Therefore, to position clearly our work, we start with defining the term "reference architecture" in Section 2 and discuss it with respect to a number of closely related terms. Section 3 and 4 contain the description of our framework. In Section 3, we describe the multi-dimensional space for classification of reference architectures. In Section 4, we present the predefined types of reference architectures. In Section 5, we discuss how the framework can be used. In Section 6, we validate the framework by applying it for the analysis of 24 reference architectures. Section 7 contains a discussion on related work. The paper ends with conclusions.

2. Reference architectures – a definition and disambiguation

A commonly accepted definition for software reference architectures does not exist [6–8]. In this section, we first provide the definition of reference architectures that we use throughout the paper. Next, from the perspective of this definition, we discuss a number of terms that are sometimes considered synonymous or closely related to the term "reference architecture". We present our view on the overlap and differences in the semantics of these terms.

2.1. A definition for reference architectures

In [3], a reference model is defined as "*a division of functionality together with data flow between the pieces*" and a reference architecture as "*a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them*".

We have selected this definition for our work as it addresses the generic nature of a reference architecture, states explicitly its software nature, defines the minimum architectural elements that should be used in its design, and covers the various types of software reference architectures that exist in practice. Other definitions cover not only software but also business and technology aspects (e.g., [9]), do not require the usage of specific architectural elements (e.g., [10]), or restrict too strongly the space of reference architectures (e.g., focusing primarily on best-practices architectures [7,8]). More importantly, while other definitions implicitly or explicitly equate the term "reference architecture" to other related terms, this definition allows us to clearly delineate the borders of the term "reference architecture". Next, we discuss the relation of a number of terms to the term "reference architecture" from the perspective of our definition.

2.2. Reference architectures and concrete architectures

Typically, an architecture is designed and used for the development of a specific software application. We call this type of architectures "concrete architectures" or simply "architectures" (also known as "application architectures" [11]). What if a concrete architecture is used for the design of several applications? Does it become a reference architecture? In other words: Where is the border between concrete and reference architectures?

In [7,8,1] we have identified the generic nature of reference architectures as a main feature distinguishing them from concrete architectures. Their generic nature implies their applicability in multiple, different contexts, reflecting the requirements of the stakeholders in these contexts. The generic nature of reference architectures is achieved by designing them at higher levels of abstraction (abstracting from the differences introduced by the contexts). Thus, we can label an architecture as reference, only if it is defined to abstract from certain contextual specifics allowing its usage in differing contexts.

2.3. Reference architectures and architectural patterns

Architectural patterns are a widely used concept in the software architecture community. According to Avgeriou and Zdun [12], architectural patterns are "well-established solutions to architectural problems". In [13], an architectural pattern is defined as a construct that "expresses a fundamental structural organization or schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them". In [3], an architectural pattern is defined as "a description of element