Software Reference Architectures - Exploring Their Usage and Design in Practice

Samuil Angelov¹, Jos Trienekens², and Rob Kusters³

¹Software Engineering Team, Fontys University of Applied Sciences, Eindhoven, The Netherlands
s.angelov@fontys.nl
²Information Systems Group, School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
j.j.m.trienekens@tue.nl
³Management Science Faculty, Open University, Heerlen, The Netherlands
rob.kusters@ou.nl

Abstract. Software reference architectures have been around for quite some years. They have been designed and used with varying success. We have conducted an exploratory survey among software architects and developers to establish the extent to which SRA have penetrated among practitioners and to identify the benefits and problems practitioners face when using and designing SRA. In this article, we present our findings.

Keywords: Reference architecture, software architecture, survey.

1 Introduction

Software reference architectures (SRA) have become one of the ways to address problems in managing complexity, size, and scope of architectures, and in managing the dynamics of the environment in which systems are designed [1]. While comprehensive methods have been defined for the design, evaluation, and application of concrete software architectures, SRA have received relatively less attention in literature [2], [3]. The reasons for this can be probably traced in an assumption that theory on software architectures is directly applicable to SRA.

In our practice, we have faced difficulties in working with SRA [4], [5]. Similar observations inspired several, recent, and with still preliminary results, research efforts [6], [7]. This triggered our interest in finding out what practitioners think of SRA. Do they design and use SRA? Do they benefit from them? Do they face problems in using them? Currently, literature does not provide insights into the status of SRA in the software community. Finding on the use of SRA are discussed in [8], but those are from a local and domain specific nature.

To answer our questions and establish an initial picture for the state of the practice of SRA, we have embarked on conducting an exploratory, cross-sectional, web-based survey on SRA, following the guidelines for data collection in software engineering disciplines discussed in [9-14]. In our survey, we targeted the following objectives:
1. Building awareness in the community for the global status of SRA.
2. Building awareness for differences (or lack of such) in existing practices with SRA on the basis of geographical, experience, etc. specifics.
3. Building awareness for problems that practitioners experience and benefits they obtain in their work with SRA.

In this paper, we present the findings from the survey. We explain our survey methodology in Section 2. In Sections 3, 4, 5, and 6 we present the results from the survey.

2 Survey Methodology, Setup, and Execution

2.1 Data Collection, Target Population, and Sampling

A survey provides a “snapshot of the situation to capture the current status” [12]. Furthermore, “explorative surveys are used as a pre-study to a more thorough investigation to assure that important issues are not foreseen … the information is gathered and analyzed, and the results are used to improve the full investigation” [12]. This fully matched with our goals of investigating the trends and establishing the status of SRA. We have opted for a web-survey as a data collection method as it allows the collection of standardized data from a large, remotely-located population, which can be used for the establishment of the status and trends of SRA. An extensive discussion on the advantages and disadvantages of on-line surveys is provided in [13].

The target population was defined to be software architects and developers as these roles are the major stakeholder roles related to SRA. We aimed at respondents from two tiers of countries as defined in [15], i.e., Tier 1 representing major software exporting nations (e.g., USA, Germany, The Netherlands) and Tier 2 formed by emerging software exporting nations (e.g., Eastern European countries, developed Latin American countries). Our sampling approach was quota sampling (defined by our tier classification) augmented by convenience and snowball sampling [11].

2.2 Survey Design and Execution

The questions in our survey were designed on the bases of existing literature on SRA, e.g., [1, 4, 16]. We have grouped the questions in order to present to the respondents a clear survey structure. Group 1 questions collected information on the respondents’ background and context. Group 2 questions contained questions determining the interpretation of the respondents for the term "software reference architecture". Group 3 questions investigated the experience of the respondents with usage of SRA, the benefits that they have obtained, and the problems that they have faced when using them. Group 4, respectively, investigated the design of SRA and the problems accompanying the design process. The latter two groups represented the core of the survey. Finally, we have added in Group 5 questions where the respondents had an opportunity to provide their opinions and thoughts on SRA. After addressing the feedback from four pilot respondents, we ended with 32 questions distributed in the five groups.