

Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case

Samuil Angelov, Jos J. M. Trienekens, Paul Grefen

Department of Technology Management,
Eindhoven University of Technology, The Netherlands
{s.angelov, j.j.m.trienekens, p.w.p.j.grefen}@tue.nl

Abstract. Reference architectures provide major guidelines for the structure of a class of information systems. Because of their fundamental role, reference architectures have to be of high quality. Before accepting a reference architecture, it has to go through a rigorous evaluation process. A number of methods exist for the evaluation of software architectures. In this paper, we analyze the main differences between concrete software architectures and reference architectures. We discuss the effects of these differences on the evaluation of reference architectures and show that existing methods cannot be directly applied for the evaluation of reference architectures. For the evaluation of a reference architecture for e-contracting systems, we used the Architecture Tradeoff Analysis Method with a number of adaptations and extensions. We present our approach and share our experiences from this evaluation process. Based on the analysis and our experiences gained, we present our vision for a method for the evaluation of reference architectures.

Keywords: software architecture, reference architecture, evaluation method.

1 Introduction

Every system has an architecture [20]. The software architecture of a program or computing system is “the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them” [6]. An architecture can be documented in an “architectural description”. While an architectural description may be used after a system has been developed (e.g., for system maintenance purposes), its value is greater when it is defined and used prior to system development. It facilitates discussions on the system to be developed among its stakeholders [8]. By agreeing a priori on a software architecture, stakeholders can be certain that they have agreed on the functionalities and design choices that they would expect to be implemented in the system. In this paper, we use the term *concrete architecture* to refer to the architectural description of a concrete software system.

Architectural design choices have direct repercussions on the system to be designed. That is why it is important to evaluate the architecture of a system before system development starts. Architecture evaluation allows timely and cheap discovery

and resolution of potential problems in the system to be developed. “Architecture evaluation is a cheap way to avoid disaster” [8]. An architecture that passes successfully through an evaluation process sets the fundamentals for the development of a high-quality system. In recent years, a number of methods for the evaluation of software architectures have been proposed [5], [10], [16].

Reference architectures have emerged as a special type of architectures that provides major guidelines for the specification of concrete architectures of one class of systems. Depending on the context in which they are defined, we differentiate between two types of reference architectures: practice-driven and research-driven reference architectures. Practice-driven reference architectures are defined when sufficient knowledge has been accumulated in a domain to propose the “best of best-practices” architecture [19]. They are designed to provide a standardized view on a class of systems. Research-driven reference architectures provide a “futuristic” view on a class of systems that are expected to become important in the future, but by the time of the architecture definition are seen as hard to build (e.g., due to functional complexity). These architectures aim at facilitating the design of the first systems from a class of systems.

Nowadays, software is evolving rapidly regarding its size and complexity. Software components are often developed by different software providers and integrated at a later stage in a system. Systems have to communicate with other systems. The system complexity, and the need for integrability of system elements and for system interoperability have led to a growing number of practice- and research-driven reference architectures (e.g., [4], [12], [13], [14], [15], [22], [23]).

Reference architectures influence the design of a set of concrete architectures and, thus, the design of a set of systems. That is why designers of a reference architecture have to present evidence for its qualities by evaluating it. However, existing methods for the evaluation of concrete architectures cannot be applied directly for the evaluation of reference architectures. The main reason for this is the generic nature of reference architectures. This characteristic of reference architectures leads to a number of differences between reference and concrete architectures. Existing methods for the evaluation of concrete architectures are not designed to deal with these specific characteristics of reference architectures. To the best of our knowledge, no method dedicated to the evaluation of reference architectures currently exists.

In this paper, we present our experiences with the evaluation of a reference architecture. We start with an analysis of the specific characteristics of reference architectures and their evaluation. This analysis allows us to motivate the need for attention to the evaluation of reference architectures and to provide the foundations for a dedicated method for the evaluation of reference architectures. Next, we present the case of the evaluation of a reference architecture for e-contracting systems and discuss our approach in this case. Based on our experiences in this case and the analysis of reference architectures, we present our vision for a method for the evaluation of reference architectures. We believe that the results presented in this paper will provide valuable pointers for the evaluation of reference architectures and will contribute to the design of a method for the evaluation of reference architectures.

The paper is organized as follows. In Section 2, we discuss concrete and reference architectures. We compare them and identify differences between them. In Section 3, we discuss the evaluation of concrete and reference architectures. We show that due

to the differences between them, existing methods for the evaluation of concrete architectures cannot be applied directly for the evaluation of reference architectures. In Section 4, we present our experiences with the evaluation of a specific reference architecture. Based on this, we present our vision for a method for the evaluation of reference architectures. The paper ends with conclusions.

2 Concrete and Reference Architectures

In this section, we present concrete and reference software architectures and discuss the goals and outcomes of their design. We compare them and identify a number of differences between them.

2.1 Concrete Architectures

In the 1990's, complex and large software systems were becoming widely spread [8]. This has led to an increased interest in the design and documentation of software architectures as a means to facilitate system development and maintenance.

An architecture description (or briefly “an architecture”) defines a set of *functionalities* and addresses certain *system*, *business*, and *architectural qualities* that are required by the stakeholders [6]. System qualities (e.g. availability, modifiability) are qualities that stakeholders require in the system to be developed. Business qualities (e.g. cost, time-to-market) are business goals that affect the system architecture. Architectural qualities (e.g. conceptual integrity, buildability) are qualities of the architecture itself.

The design of high-quality concrete software architectures has been given significant attention in the literature [6], [20], [21].

2.2 Reference Architectures

According to [6] a reference model is “*a division of functionality together with data flow between the pieces*”, and a reference architecture is “*a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them*”. A reference architecture is based on the functionalities and data flows defined in a reference model and applies architectural styles and patterns that help in addressing the main qualities expected from the architecture (see Fig.1). A “good” reference architecture can bring a number of benefits [19]. It may facilitate the design of high-quality concrete architectures; it may facilitate communications between domain professionals, etc.

A reference architecture can be defined before the existence of practical experiences with the design of concrete architectures. The design of such a reference architecture is inspired by existing research efforts. Thus, these reference architectures are research-driven. These architectures follow the “top-down” approach presented in Fig.1, i.e., a reference architecture is based mainly on a reference model and on existing architectural patterns. We call these reference architectures Futuristic

Reference Architectures (FRAs), as their goal is to make an attempt to “look into the future” and to foresee the major design principles that will be of importance in the design of concrete architectures for a specific domain. Examples of a FRA are [4], [18], [22].

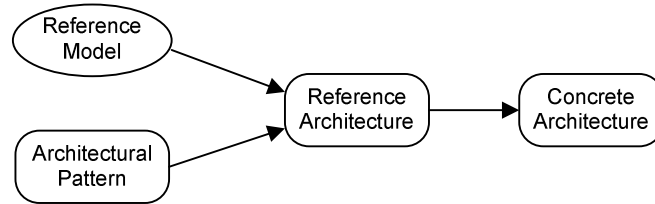


Fig. 1. The relationship between reference models, reference architectures and concrete architectures (adapted from [6]).

Often, reference models and reference architectures are defined based on accumulated practical experience in domains, i.e., they are practice-driven. In this paper, we call practice-driven reference architectures Practice Reference Architectures (PRAs). As the design of PRAs is inspired from practice, the design process can be seen more as following a “bottom-up” approach in which concrete architectures play the major role for the design of a reference architecture (see Fig.2). Another consequence from the “practice” roots of PRAs is that they might address legacy issues in their design. PRAs are usually (but not necessarily) elaborated by recognized standardization bodies that facilitate developments within a domain, or by consortiums established by powerful companies within the domain which aim at establishing or even enforcing standards within the domain. Examples of a PRA are [13], [14], [23].

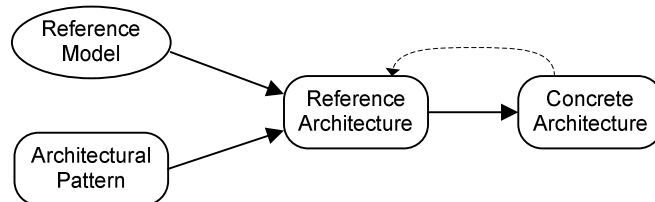


Fig. 2. The influence of concrete architectures in the case of PRAs.

PRAs and FRAs have certain differences with respect to their origin and goals. In the case of PRAs, the functionalities that may be part of a system are known. PRAs are based on existing “best practices” often interwoven with existing legacy issues. Thus, we can view the origin of PRAs as *descriptive*. In the case of FRAs, only limited existing practices can be used (i.e., architectural patterns, architectures of prototypes). As there are not complete solutions that exist in practice, we can view the origin of FRAs as *prescriptive*.

Table 1. Origin and goals of PRAs and FRAs (P - prescriptive, D – descriptive).

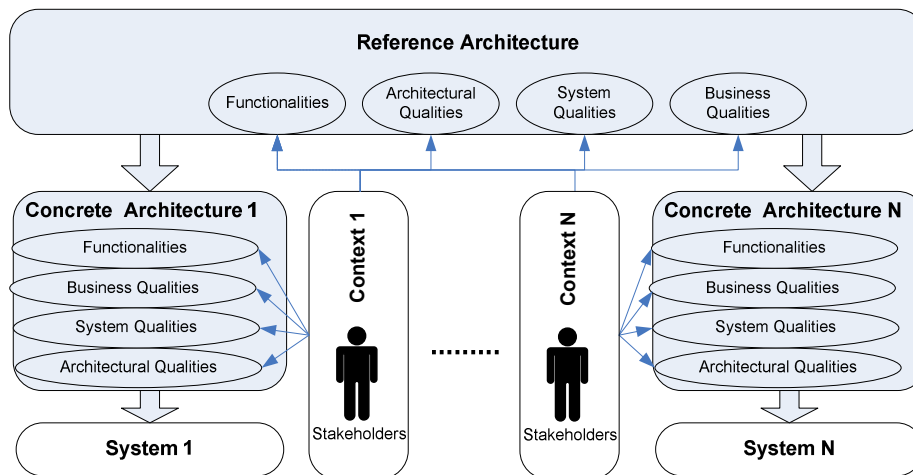
	Origin	Goals
PRA	D	P
FRA	P	D

Generally, any architecture of a system-under-development has prescriptive goals with its design. However, an in-depth look into PRAs and FRAs reveals an interesting nuance. PRAs are designed to facilitate faster system design and development and to address standardization problems in a domain. Thus, their main goal is to serve as *prescriptive* tools. FRAs are designed to facilitate the design of architectures of *first* systems in a domain. FRAs provide detailed descriptions of their “novel” functionalities. These details are required to clarify the innovative elements in the architecture as well as to convince the domain users for the qualities of systems based on the FRA (e.g., their “buildability”). Due to their avant-garde features, FRAs will often never assert themselves as accepted reference architectures. Thus, FRAs are designed to serve as *descriptive* tools and have more limited goals as prescriptive tools. We represent the nuances in the origin and goals of PRA and FRA in Table 1.

2.3 Comparison of Concrete and Reference Architectures

There are a number of differences between reference architectures (PRAs and FRAs) and concrete architectures. Next, we present these differences. The results from this section provide the foundations for our discussion in Section 3.

Difference 1: Reference architectures are of a generic nature. A reference architecture is designed to address the functionalities and qualities desired by all stakeholders in their specific contexts (see Fig.3).

**Fig. 3.** The role of stakeholders and contexts for reference and concrete architectures.

Difference 1 is a fundamental difference and is the basis of a number of specific differences:

Difference 2: *There is not a clear group of stakeholders of a reference architecture.* As stakeholders can be seen all companies from the domain, all companies developing software for the domain, etc. However, it is not possible to involve all these stakeholders in the definition of a reference architecture (due to logistic, political, etc. reasons).

Difference 3: *Due to their generic nature, reference architectures are defined on a high level of abstraction.* They may provide details only for specific elements. For example, as discussed in Section 2.2., in the case of FRAs, novel elements with complex structure may be paid a closer look. In the case of PRAs, elements critical for the standardization goals of the architecture may be defined in greater detail.

Difference 4: *A reference architecture has to address more architectural qualities than a concrete architecture.* These additional architectural qualities are due to the generic nature of reference architectures and their wider audience. For example, an “applicability” quality would be of importance for a reference architecture to indicate the level of applicability of the architecture to different contexts in the domain. This quality is superfluous for a concrete architecture as a concrete architecture is designed to be applicable for a specific context.

Because of these differences between concrete and reference architectures, reference architectures are considered by some authors as very distant from concrete architectures: “reference architectures are not architectures; they are useful concepts that capture elements of an architecture” [6].

3. Evaluation of Architectures

In this section, first, we discuss the goals and outcomes of the evaluation of concrete architectures and methods that can be used for their evaluation. Next, we discuss the goals and outcomes of the evaluation of reference architectures. We show that due to the differences between concrete and reference architectures, existing methods for the evaluation of concrete architectures cannot be directly applied for the evaluation of reference architectures.

3.1 Evaluation of Concrete Architectures

System development is an expensive process in terms of costs and time. Evaluation of the architecture of a system prior to its development allows “measuring” the expected level of achievement of the system functionalities and system, business, and architectural qualities required by the stakeholders. Timely discovery of failure to achieve desired functionalities and qualities means saved time and resources in the development process and avoids frustrations among stakeholders.

A number of methods exist for evaluation of software architectures. These methods differ in their evaluation techniques as well as in their goals. Most methods (e.g., SAAM, ALMA) rely on questioning techniques (asking the stakeholders qualitative questions) and use scenarios as their main tool [10]. Few methods (e.g., SAEM) rely

on measuring techniques that support quantitative measurement and evaluation of architectures. Particular methods are designed to evaluate only specific architectural qualities. For example, SAAM and ALMA are suitable for the evaluation of the modifiability quality [16]. Other methods support the evaluation of multiple qualities (e.g., ATAM and SBAR). According to [10], some methods can be integrated easier in the design process than other methods (e.g., SBAR, ATAM). In [10], the authors conclude that ATAM (Architecture Trade-off Analysis Method) has as advantages its integration of questioning and measuring techniques, the wide set of qualities that can be evaluated through it, and the possibility of integrating the method easily in the design process. An overview and comparison of existing methods can be found in [5], [10], [16].

It must be noted that existing evaluation methods provide techniques mainly for the evaluation of system qualities. The definition and evaluation of business and architectural qualities has received little attention in the literature. CBAM [6] can be distinguished as a method for the evaluation of costs, benefits, and risk business qualities. ATAM [8] addresses explicitly the evaluation of the “conceptual integrity” architectural quality. Though it is not explicitly stated in the method, the generation of scenarios in ATAM can be used for the evaluation of the “completeness” architectural quality as well.

3.2 Evaluation of Reference Architectures

In order to establish an effective reference architecture with respect to many concrete architectures, a reference architecture should have a high degree of excellence. To identify the aspects that may require additional attention before its release and to prove its final value, a reference architecture requires evaluation. In Section 2.2, we discussed that a reference architecture contains a description of functionalities and addresses certain system, business and architectural qualities. Thus, concrete and reference architectures have to be evaluated for the same aspects. However, as discussed in Section 2.3, concrete and reference architectures have certain differences. These differences lead to a number of problems that do not allow the direct application of methods for the evaluation of concrete architectures in the case of reference architectures. Next, we explain our motivation for this statement.

Problem 1: One of the problems for applying an existing method for the evaluation of reference architectures is caused by the lack of a clearly defined group of stakeholders (see Difference 2). ATAM and most other methods heavily rely on the participation of all stakeholders in its evaluation. However, reaching all stakeholders of reference architectures and convincing them to participate in an evaluation is problematic. In both cases (PRAs and FRAs), the big number of stakeholders makes it impossible to address all of them. Furthermore, in the case of PRAs, often, stakeholders will not unite around a common reference architecture due to political and contextual differences (rivalry, different legacies, etc.). In the case of FRAs, most stakeholders will have limited incentives (as there are no direct benefits for them) and capabilities (due to lack of visionary thinking and knowledge) to contribute to the architecture evaluation.

Problem 2: As discussed in Section 3.1 most evaluation methods make use of scenarios. However, the generic nature of reference architectures (see Difference 1) and their high level of abstraction (see Difference 3) make the generation of a usable set of scenarios difficult. Due to the generic nature of reference architectures, evaluators have the choice to either define a large set of “concrete” scenarios for the possible contexts in which the reference architecture can be applied or define highly general scenarios which cover all these contexts. In the first approach, the huge number of possible contexts results in a huge number of scenarios. This makes defining and prioritizing them a problematic task. In the second approach, the generality of scenarios makes it hard to evaluate their adequate support in the architecture. This problem has already been observed even in the evaluation of concrete architectures of information systems, whose complexity leads to the definition of highly general scenarios [7]. The abstract nature of (parts of the) reference architectures further aggravates the problem of generating concrete scenarios. In the case of FRAs, the lack of practical knowledge for the contexts in which concrete architectures will be defined makes generation of scenarios a “guessing game”.

Problem 3: In Section 3.1, we mentioned that from the existing methods for the evaluation of concrete architectures only ATAM addresses explicitly the evaluation of the “conceptual integrity” architectural quality and implicitly of the “completeness” architectural quality. However, reference architectures have to address more architectural qualities than concrete architectures (see Difference 4). Consequently, existing methods fall short in providing techniques for the evaluation of the architecture qualities of PRAs and FRAs.

This brief discussion shows that existing methods on the evaluation of concrete architectures are not directly applicable for an evaluation of reference architectures. In the recent years, software product lines gained the attention of research and industry [6], [9]. Software product line architectures (also called family architectures) are abstractions of concrete architectures that allow architecture reuse for a number of software products that share a common foundation. In [11], software product line architectures are positioned between reference architectures and concrete architectures and a dedicated method for their evaluation called FAAM is proposed. The more generic nature of software product line architectures suggests that FAAM may be more suitable for the evaluation of reference architectures than methods for the evaluation of concrete architectures. However, similar to methods for the evaluation of concrete architectures, FAAM does not deal with the evaluation of architectural qualities. Furthermore, in FAAM, the stakeholders are expected to be involved actively in the evaluation process. Thus, FAAM does not resolve the problems identified in this section and cannot be applied for an evaluation of reference architectures.

4. An Approach to the Evaluation of Reference Architectures

In our previous work, we have faced the problem of evaluating an E-contracting Reference Architecture (ERA) [4]. This section starts with a brief presentation of the

context of ERA. For the evaluation of ERA, we used the Architecture Tradeoff Analysis Method (ATAM). As ATAM is designed for the evaluation of concrete architectures, we had to apply a number of adaptations and extensions on it. We present our evaluation approach and share our experiences from it. Based on our experiences and the discussion in Section 3, we present our vision for a method for the evaluation of reference architectures.

4.1 The E-contracting Reference Architecture (ERA)

Business-to-business e-contracting uses information technology for improving the efficiency and effectiveness of contracting processes of companies. A reference architecture that provides guidelines for the design of concrete architectures of highly automated e-contracting systems will significantly facilitate the software development process and will introduce a standardized view on e-contracting systems. So far, the domain of highly automated e-contracting has been addressed mainly by the research community. Over many years, industry considered it to be too complex and rigid and hardly applicable in practice. However, there is currently an increasing interest in the industry in the development of more advanced contracting systems with a higher level of automation.

In [1], we presented our initial design of ERA. According to the discussion in Section 2.2, ERA can be classified as a FRA. In the next section, we explain the approach that we took for the evaluation of ERA and the results from it. A detailed description of the evaluation process and of the final version of ERA can be found respectively in [3] and [4].

4.2 The Evaluation of ERA

Our initial approach was to evaluate ERA by means of an existing method for the evaluation of concrete architectures. We chose to use ATAM as a method for the evaluation of ERA because of its advantages in a number of aspects over other methods (see Section 3.1), and because of its successful application in many projects [5]. Our team had no previous experiences with ATAM. After introducing ourselves to ATAM, we foresaw a number of problems (presented in Section 3.3). Realizing the lack of a dedicated method for the evaluation of reference architectures, we decided to attempt the evaluation of ERA with ATAM and to make adaptations and extensions on ATAM where the “reference architecture” context required it. Next, we describe the phases in the evaluation of ERA (based on ATAM) and the adaptations and extensions that we made on ATAM.

Phase 1: Identification of the stakeholders of ERA that will be involved in its evaluation; preparation of the evaluation process.

Activities: We invited 3 researchers with experience in e-contracting and software architectures in the role of software architects/designers with whom we performed Phases 2a and 2b of ATAM. For Phase 3 of ATAM, we involved a group of 25 contract business professionals.

Adaptations of ATAM: ATAM assumes the identification of stakeholders and their participation in the evaluation process to be a rather straightforward process. As discussed in Problem 1 (see Section 3.2), this is not the case in reference architectures. To solve this problem, we identified first the roles for the different stakeholders, i.e., contract (procurement) managers, software architects/designers/managers, contract engineers, legal officers, and CEOs. We estimated that people acting in these roles in a company might be affected by the development and introduction of an e-contracting system. Next, we searched for people representative for these roles. As ERA is a FRA, we were looking for people with visionary thinking who had (or can build) understanding for the advanced elements in ERA. We managed to approach people with such qualities for Phases 2a and 2b. However, approaching business professionals for the evaluation of ERA was difficult. As discussed, a FRA may never be accepted as a reference architecture. Thus, there was no direct incentive for business professionals to participate in the evaluation of ERA. To offer an incentive for the business professionals to spend time on the evaluation of ERA, we organized a five-hour tutorial on electronic contracting within the scope of the "4th Annual Contract Management Conference" [2]. The main goal of the tutorial was to present the basic aspects of e-contracting. A discussion on the architecture of advanced e-contracting systems was announced as secondary element of the tutorial. The educational goal of the tutorial appeared to be a sufficient incentive for attracting 25 attendees. The attendees had the following role distribution: 8 contract managers (2 of the contract managers had also a function as IT experts in their departments); 4 procurement managers, 2 contract engineers, 3 CEOs; 2 legal officers; 2 government representatives; 4 professionals with other business functions.

Phase 2a: Elicitation of the architecture and required qualities; identification of architectural approaches.

Activities: We evaluated the set of required qualities in ERA. At the end of this step, we had a list of system and architectural qualities (system qualities: security, flexibility, modifiability, integrability, high automation, interoperability; architectural qualities: conceptual integrity, completeness, buildability, applicability, usability, acceptability). After this phase, the amount of quality attributes required in ERA has increased compared to our initial list of required qualities. Furthermore, our understanding for the required qualities in ERA has improved. Next, we defined scenarios for the qualities that we identified. Finally, we identified the architectural approaches that we used in ERA.

Adaptations of ATAM: Initially, the lack of concrete context resulted in the definition of highly-general and (mostly) equally important scenarios (related to Problem 2). ATAM does not provide any guidelines on the definition and prioritization of scenarios in such general settings. Our approach to resolve this problem was to select a number of contexts in which ERA can be applied and define concrete scenarios relevant for them. We selected three trading domains (i.e., the advertising, the logistics, and the insurance domains) for the e-contracting aspects of which we had knowledge from previous research. As contracting practices differ mainly per domain, selection of different domains was a sufficient concretization in our case. However, the domains that we selected were by no means representative for the large set of domains where e-contracting can take place. So, the selection was a

pragmatic choice. Having in mind the generic nature of reference architectures, we looked only for scenarios that were applicable for all three selected domains.

Phase 2b: Analysis of the architectural approaches and their effect on the selected qualities.

Activities: We analyzed the suitability of the architectural approaches used in ERA for achieving the security, flexibility, automation, modifiability, integrability, and interoperability qualities. We discussed the risks, non-risks, sensitivity points and tradeoffs of ERA. These activities led to a number of improvements of ERA. Lack of a consistent strategy for the exchange of data among components and invocation of components was discovered. The requirements for modifiability and integrability were not addressed consistently throughout the architecture as well. To address these problems, a number of additional architectural styles and patterns were introduced in ERA. We discovered also that a new component had to be added to the architecture to satisfy the interoperability requirement. Thus, the evaluation of desired qualities of ERA through ATAM led to substantial improvements in the architecture. After improving ERA, we re-evaluated it and elaborated a final list of risks, non-risks, sensitivity points and tradeoffs in ERA. This list contained issues that were all within our scope of expectations.

Adaptations of ATAM: As discussed in Problem 3, ATAM is designed to address mainly system qualities and the conceptual integrity architectural quality. That is why, at this stage, we selected and evaluated only the qualities of ERA that can be evaluated through this step in ATAM, and we skipped the remaining qualities.

Phase 3: Verification of the results from Phase 2 by involving “business” stakeholders.

Activities: As already mentioned, in this phase, we organized a tutorial on e-contracting [2]. In this tutorial, first, we presented the main aspects of e-contracting to the audience (time used: 2 hours). As a next step, we presented ERA (time used: 1 hour). In the remaining time of the tutorial (2 hours), we organized a discussion session on the qualities which the participants would expect from an e-contracting system. We asked the participants to generate scenarios which they think would be relevant for such a system and to rank them. Each participant was given the right to cast 3 votes in total. The results from the workshop helped us to make final adaptations to ERA. The qualities that the tutorial participants identified overlapped with the qualities already identified in Phase 2a. We used the scenarios generated in Phases 2a and 3 to evaluate the functional completeness quality of ERA. For our surprise, the scenario that was ranked highest by the participants was not addressed properly in ERA. As a consequence, we introduced a new component in ERA to address the scenario identified by the participants.

Adaptations of ATAM: The results from this phase proved that this phase was a useful element of the evaluation process. However, in this phase, we faced a number of problems for the resolution of which we had to make adaptations to ATAM.

ERA is defined for the design of highly-automated contracting systems. The idea of such advanced systems is currently addressed only in the research world. Though we spent a significant amount of time in presenting the essential aspects of e-contracting and the goals of ERA, the participants did not accept this idea and discussed a system with a low-level of automation. This can be explained by the lack of visionary thinking by some of the participants and by the conflict of interests

between the goals of the participants and our goals. The participants had to suggest scenarios for a futuristic system which was beyond the imagination of many of the participants. Furthermore, the participants were interested in practical solutions that can be implemented straightforwardly. Thus, they were interested in discussing a less advanced system that could be developed on the basis of existing technology. As a consequence, we had to re-formulate the scenarios defined by the participants for the situation of a highly automated e-contracting system. Some scenarios made no sense in the case of highly-automated system and had to be removed from the list. From the 22 scenarios that were generated, finally we considered 16 to be relevant.

We gave all participants 3 votes that they could use for the prioritization of the scenarios. In ATAM participants are given votes equal to around 30% of the number of scenarios defined (this would mean 4-5 votes in our case). The reason to decrease the number of votes was the lack of time combined with the inability of some participants to easily rank the scenarios. Some participants could not prioritize the scenarios as they did not have a good base for reasoning. In the voting process, we did not take into consideration the different roles of the participants. As not all people had the visionary thinking required for this evaluation, any attempt to influence the voting process based on the different roles might have had a negative rather than a positive effect on the process (e.g., people with more visionary ideas might get less votes in case they are representing the same role).

We had to omit a number of steps advocated in ATAM. We did not present to the participants ATAM in detail and the results from the previous steps of ATAM. We estimated that presenting explicitly ATAM and our previous results from the evaluation process would consume too much time (which we did not have) and would de-motivate people to present their own views on e-contracting systems. Due to the lack of time, we did not perform an in-depth discussion on the ranking of the scenarios. ATAM advocates analysis of the architectural approaches in this phase. However, we decided to skip this step in our evaluation as the knowledge of all participants was insufficient for such analysis. These adaptations are related to Problems 1 and 2.

Additional uses and extensions of ATAM: To evaluate some of the qualities not evaluated in Phase 2b, we made use of certain steps in ATAM and extended these steps with techniques for the evaluation of the qualities. For the evaluation of the remaining qualities, we had to add additional steps and techniques to ATAM.

We used Phase 2a and Phase 3 of ATAM to evaluate the usability and acceptability architectural qualities of ERA. We considered the time required to introduce ERA to the stakeholders and the level of their understanding of the architecture as metrics for the achievement of this quality.

As it was not possible to address all domain stakeholders and obtain their agreement on the scope of ERA (see Problem 1), we concluded that completeness of ERA has to be evaluated beyond the scenario-based technique provided in ATAM. We extended ATAM with two additional techniques for the evaluation of the completeness of ERA. First, we used a reference model on e-contracting and reasoned whether the concepts in it were addressed in ERA. Second, we compared ERA to existing concrete e-contracting architectures and showed that ERA addresses the functionalities defined in these architectures. We could not evaluate the applicability and buildability qualities in any step in ATAM and had to *extend* it with additional

techniques. To demonstrate the applicability of ERA to different contexts, we applied it for the analysis of existing concrete e-contracting architectures and for the design of hypothetical architectures in the sample domains that we selected in Phase 2a. In [6], the evaluation of the buildability quality is done through ATAM. However, in the case of ERA, many components support functionalities that are addressed only in recent research developments from different research domains and that implement complex, little tested (or even not defined) algorithms. We realized that a component might be easily buildable in one business context and very complex to build for another context. Certain domains make use of simple trading scenarios, while others involve complex interactions among companies for the establishment and enactment of a contract. A brainstorming session during Phase 2a of ATAM on the buildability quality of ERA could not be carried out, as none of the participants had detailed and up-to-date knowledge on the status quo of the research results for the components supporting complex functionalities. In order to present convincing evidence for the buildability of ERA, we performed a literature survey on the existing research results. This step required a substantial amount of time. The results from this step were a list of existing research and industry results that can be used as a basis for building some of the components and a list of "buildability risks" that exist due to the impossibility to estimate the buildability quality of some components for all business contexts.

Discussion on the evaluation process: ERA evolved substantially after its evaluation. Its structure, conceptual integrity, and functional completeness improved. The adaptations and extensions to ATAM that we used in our case allowed us to better evaluate our reference architecture for the identified required qualities. In contrast to the original ATAM process that has relatively fixed time duration, these additional activities required substantial time and resulted in a long-running evaluation process (around 3 months).

We concluded that the heterogeneity in knowledge and background of the participants in Phase 3 led to the inability of some of them to contribute to the evaluation process. Particular steps in ATAM were beyond the skills of all participants in this phase. We failed in involving a sufficient number of highly motivated, experienced, information technology knowledgeable, and visionary practitioners for Phase 3.

ERA has not been applied for the design of e-contracting systems yet. We are currently disseminating it to parties that may be interested in it. Its application for the design of concrete e-contracting architectures will give an indication for the quality of the architecture and thus for the quality of the evaluation process.

4.3 Generalization of the Approach

In this section, we generalize our findings from the evaluation of ERA. Our experience with the evaluation of ERA showed that though ATAM is not designed explicitly for the evaluation of reference architectures, its application may bring substantial improvements to a reference architecture. However, direct application of ATAM is not possible. A number of adaptations and extensions on ATAM are required for its successful application for the evaluation of reference architectures. That is why we think that a method for the evaluation of reference architectures may

use ATAM with certain adaptations as a foundation and extend ATAM with a number of steps and techniques. Our discussion in Section 2 shows that due to the differences between PRAs and FRAs, the adaptations and extensions of ATAM will vary for the evaluation of PRAs and FRAs.

Usage of ATAM: ATAM is applied for the evaluation of system qualities like interoperability, modifiability, performance, conceptual integrity, etc. Scenarios are used for evaluation (partially) of the completeness quality. Furthermore, we suggest using the discussion meetings in ATAM to evaluate the level of understanding and acceptance of the architecture by the stakeholders (i.e., usability and acceptance qualities).

Adaptations of ATAM: For an evaluation of reference architectures, ATAM has to be adapted at two points, i.e., identification of stakeholders and scenario definition and prioritization. We suggest that evaluators adapt ATAM in the following ways:

- Define roles for the stakeholders from the domain and invite representatives for these roles that will participate in the evaluation process. In the case of PRAs, representatives from leading industry solutions are recommended for the evaluation sessions. Selection of a “good” set of representatives is the main challenge in this situation (i.e., stakeholders that are willing to collaborate and have common goals). In the case of FRAs, leading researchers should play a main role in Phase 2 of ATAM and experienced and interested in future developments business stakeholders in Phase 3. The identification and involvement of stakeholders who have the knowledge and the motivation to contribute to the evaluation of the FRA is a challenge in this situation.
- Select a number of contexts and define scenarios for these contexts. Prioritize scenarios within a concrete context, and merge the prioritized scenarios in a general set of scenarios.

Extensions of ATAM: ATAM needs complementary activities and techniques for evaluation of certain architecture qualities. In our experience, we have faced the need to extend ATAM for the evaluation of three architecture qualities, i.e., completeness (besides scenario-based evaluation), applicability, and buildability. As for many PRAs and FRAs these qualities will be of importance, next, we present the activities and techniques that we propose as extensions of ATAM.

- To thoroughly evaluate completeness, we suggest the usage of existing, “best-practice”, concrete architectures and comparing their functionalities to the functionalities of the reference architecture. However, in the case of FRAs, there might be too few (or none at all) relevant concrete architectures. That is why, for FRAs, we suggest also the usage of a recognized reference model and analysis of the architecture for its support. This reference model has to be different from any reference model used in the design of the reference architecture.
- To evaluate the applicability of FRAs and PRAs, we propose the definition of a number of concrete architectures for specific contexts based on the reference architecture and an evaluation of the applicability of the reference architecture in these contexts. For PRAs (and when possible for FRAs), existing concrete architectures in specific contexts can be analyzed directly from the perspective of the reference architecture.
- To evaluate the buildability quality, a number of concrete contexts have to be selected in which the buildability of components is discussed. The contexts should

be selected on the basis of the complexity of the required system functionalities. In the case of FRAs, evaluators should address not only existing technology but also existing research results (prototypes and theoretical developments) and provide examples about how these can be used or adapted to support the functionalities defined in the reference architecture.

5. Conclusions

In this paper, we distinguish between two types of reference architectures, i.e., Practice Reference Architectures (PRAs) and Futuristic Reference Architectures (FRAs). We compare reference architectures to concrete architectures and show that the specific characteristics of reference architectures do not allow existing methods for the evaluation of concrete architectures to be used straightforwardly for the evaluation of reference architectures. We share our experiences and conclusions from a case on the evaluation of a FRA. Based on our experiences from this case, we outline a number of adaptations and extensions that in our opinion must be applied on the existing evaluation method ATAM for the evaluation of reference architectures.

Currently, we are involved in the evaluation of another FRA called e-Sourcing Reference Architecture (eSRA) [18]. The first results from its evaluation confirmed our findings regarding the differences between concrete and reference architectures. We also faced the problems discussed in this paper in using ATAM. We currently use the case of the evaluation of eSRA to further elaborate our approach for the evaluation of reference architectures.

As future work, we aim at defining a detailed method for the evaluation of reference architectures. To reach this goal, a number of points in our approach require further attention. Guidelines for the identification and involvement of stakeholders in the cases of PRAs and FRAs must be elaborated. We shall base our future work in this direction on existing literature on stakeholder analysis, e.g., [17]. Definition and prioritization of scenarios is paramount for the approach but is currently not a precisely defined process. Guidelines for the selection of contexts are required. Our experiences showed that different criteria for the selection of contexts can be applied in the different evaluation steps. This indicates that different guidelines for the selection of contexts in the different evaluation steps should be defined. Metrics for the evaluation techniques suggested by us must be defined as well. The specific characteristics of PRAs and FRAs indicate that a method for the evaluation of reference architectures has to recognize the differences between them and has to address these differences in separate ways at certain points. Our current conclusions are based on the evaluation of FRAs. In our future work, we plan to test our findings for the evaluation of PRAs as well.

References

1. Angelov, S.: Foundations of B2B Electronic Contracting, PhD Thesis. Eindhoven University of Technology, Eindhoven (2006)

2. Angelov, S.: Defining E-Contracting and Measuring its Significance. In: Post-Conference Workshop at the 4th Annual Contract Management Conference, Dubai. Institute for International Research (2007)
3. Angelov, S.: Evaluation of the E-Contracting Reference Architecture. Technical report, Beta Working Paper, WP 225, Eindhoven University of Technology (2007)
4. Angelov, S., Grefen, P.: An E-contracting Reference Architecture. *Journal of Systems and Software* (available online 26 February 2008). To appear
5. Babar, M., Gorton, I.: Comparison of Scenario-Based Software Architecture Evaluation Methods. In: 11th Asia-Pacific Software Engineering Conference (APSEC'04), Washington, pp. 600-607. IEEE Computer Society (2004)
6. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice* (second edition). Addison-Wesley Professional (2003)
7. Bengtsson, P., Bosch, J.: Scenario-Based Software Architecture Reengineering. In: Fifth International Conference on Software Reuse, 1998, Victoria, Canada, pp. 308--317. (1998)
8. Clements, P., Kazman, R., Klein, M.: *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley Professional (2002)
9. Clements, P., Northrop, L.: *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional (2001)
10. Dobrica, L., Niemelä, E.: A Survey on Software Architecture Analysis Methods. *IEEE Transactions on Software Engineering*, 28, 7, 638--653 (2002)
11. Dolan, T.: *Architecture Assessment of Information-System Families - A Practical Perspective*, PhD Thesis. Eindhoven University of Technology, Eindhoven (2001)
12. Ferrara, F.: The Standard 'Healthcare Information Systems Architecture' and the DHE middleware. *International Journal of Medical Informatics*, 52, 1, 39--51 (1998)
13. Grefen, P., Remmerts de Vries, R.: A Reference Architecture for Workflow Management Systems. *Data & Knowledge Engineering*, 27, 1, 31--57 (1998)
14. Hollingsworth, D.: *The Workflow Reference Model*. Technical report, Workflow Management Coalition Documents, TC00-1003, Workflow Management Coalition (1995)
15. Bontempo, C., Zagelow, G.: The IBM Data Warehouse Architecture. *Communications of the ACM*, 41, 9, 38--48 (1998)
16. Ionita, M., Hammer, D., Obbink, H.: Scenario-Based Software Architecture Evaluation Methods: An Overview. In: Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering, Orlando, Florida, USA. (2002)
17. Kusters, R., Solingen, R., Trienekens, J.: Identifying Embedded Software Quality: Two Approaches. *Quality and Reliability Engineering International*, 15, 6, 485-492 (1999)
18. Norta, A.: *Exploring Dynamic Inter-Organizational Business Process Collaboration*, PhD Thesis. Eindhoven University of Technology, Eindhoven (2007)
19. Reed, P.: *Reference Architecture: The Best of Best Practices*. (2002)
<http://www.ibm.com/developerworks/rational/library/2774.html>
20. Rozanski, N., Woods, E.: *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley Professional (2005)
21. Shaw, M., Garlan, D.: *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall (1996)
22. Wu, H.: *A Reference Architecture for Adaptive Hypermedia Applications*, PhD Thesis. Eindhoven University of Technology, Eindhoven (2002)
23. Zimmermann, H.: OSI Reference Model - the ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28, 4, 425--432 (1980)