2012 12th International Conference on Quality Software

Qualitative Analysis of the Impact of SOA Patterns on Quality Attributes

Matthias Galster
University of Groningen, The Netherlands
mgalster@ieee.org

Paris Avgeriou
University of Groningen, The Netherlands
paris@cs.rug.nl

Abstract—**Software architecture patterns are proven and reusable solutions to common architecture design problems. One characteristic of architecture patterns is that they affect quality attributes (e.g., performance, reliability). Over the past years, architecture patterns for service-based systems have been proposed. In this paper we report initial findings about the impact of architecture patterns for service-based systems on quality attributes of service-based systems. We related more than 70 SOA patterns from a patterns catalogue to a quality model for service-based systems. Based on the description of the patterns in the catalogue, we characterized the interaction of patterns and quality attributes. We found several patterns which do not seem to explicitly address quality attributes. Our findings can be used to select SOA patterns for architecture design. Also, our findings point to directions for further research: Our preliminary results indicate a mismatch between patterns for service-based systems and quality attributes that are considered important for service-based systems; thus, future work should focus on quality models for service-based systems and on identifying architecture patterns and pattern languages for service-based systems.**

I. INTRODUCTION

Software architecture patterns represent solutions to frequently occurring software architecture problems. Architecture patterns affect quality attributes (e.g., performance) and help achieve quality attribute requirements (i.e., requirements on quality attributes, such as a low response time for quality attribute performance). Every architecture pattern provides benefits, e.g., positive impacts on quality attributes. On the other hand, each architecture pattern comes with certain liabilities, e.g., negative impacts on quality attributes. For example, the layered architecture pattern increases maintainability but has a negative impact on performance [1].

Service-oriented architecture (SOA) is the architecture paradigm underlying service-based systems and a popular approach in software industry. Service-based systems emphasize certain quality attributes [2, 3]. Over the past years, architecture patterns for SOA[1] have been proposed. However, there is currently no systematic overview to show how well SOA patterns help achieve quality attribute requirements. Therefore, the question explored in this paper is how current SOA patterns affect quality attributes that are claimed to be important for service-based systems. We make the assumption that the importance of a quality attribute for service-based systems is determined by its inclusion or absence in the quality reference model for service-based systems presented in [2]. Also, we assume that the SOA patterns we analyze are indeed architecture patterns, i.e. they are system-wide solutions and thus affect quality attributes.

The goal of this paper is to present an initial overview of how SOA patterns affect quality attributes of service-based systems. We discuss what quality attributes are addressed most, what quality attributes are not addressed at all (even though considered relevant for service-based design) and highlight problems with current SOA patterns and quality models for service-based design.

In Section II of this paper we present background information. Section III introduces the methodology applied to relate patterns and quality attributes. Results are discussed in Section IV and limitations are acknowledged in Section V. We conclude the paper in Section VI.

II. BACKGROUND

Service-based systems are systems that are assembled from individual services, invoked using standardized communication models. These systems follow a standard-based, technology-independent computing paradigm for distributed systems. SOA is the underlying architectural paradigm of service-based systems. Important ideas of SOA are a) the identification of services aligned with business drivers, and b) the ability to address multiple execution environments.

Quality attributes are characteristics that affect the quality of software systems. Quality attributes can be about the system (e.g., availability, modifiability), business-related aspects (e.g., time to market) or about the architecture (e.g., correctness, consistency) [4]. Considering quality attributes during the design stage is crucial to produce systems that meet their desired quality attribute requirements. Achieving quality-attributes in service-based systems is critical [5] because service-based systems lack central control and authority, have limited end-to-end visibility of services, are subject to unpredictable usage scenarios and support dynamic system composition [5].

Software architecture patterns are reusable and well-established solutions to architectural design problems. Architecture patterns define an architecture problem to which they can be applied, and a solution to this problem [6]. Examples of architecture patterns include Model-View-Controller, Layers, or Pipes-and-Filters. A single architecture pattern provides a solution for a specific problem but might not resolve all requirements when architecting a system [1].

[1] We call architecture patterns for SOA "SOA patterns".

1550-6002/12 $26.00 © 2012 IEEE
DOI 10.1109/QSIC.2012.35

Therefore, there is usually more than one architecture pattern applied. A group of architecture patterns that address similar problems and the relationship between these patterns are described in pattern languages [6]. Architecture patterns are the only patterns that impact quality attributes, as they consist of system-wide solutions. For example, object-oriented design patterns are not system-wide so their impact on quality attributes is local [7].

III. RESEARCH METHOD

A. Selection of SOA Patterns

To obtain a valid and representative sample of current patterns, we use the SOA patterns catalogue presented in [8]. This catalogue is the first attempt to collect patterns for service-based systems. As argued in [9], this catalogue together with candidate patterns published at the website related to the catalogue is the most complete collection of patterns currently available. All relevant patterns found in other sources were also found in this catalogue [9].

The patterns catalogue distinguishes patterns for architectures of software services (service patterns), patterns for service composition architectures (service composition patterns), patterns for service inventory architectures (architecture that supports a collection of services that are independently standardized and governed), and patterns for service-oriented enterprise architectures (architecture of an enterprise itself, to whatever extend it is service-oriented). We only analyzed the first three types as enterprise architecture patterns were not exhaustively discussed in the catalogue.

B. Selection of Quality Attribute Model

There are many software quality attributes but not all are important for service-based systems. In general, not much work on quality models for service-based systems exists. One example of such work discusses a list of nine traditional quality attributes (e.g., performance, security) and their role in service-based systems design [3]. Thus, we chose the only available quality model for service-based systems, the S-Cube Quality Reference Model (QRM) for service-based applications [2]. The S-Cube QRM defines quality attributes in a tree structure with parent quality attributes and children attributes. Furthermore, process quality, product quality and quality in use are distinguished. We focus on product quality.

C. Data Collection and Pattern Analysis

For each analyzed pattern we recorded the name as well as a reference to its original source, the problem for which a pattern provides a solution, S-Cube quality attributes affected by the pattern, and other quality attributes affected (quality attributes not in the S-Cube QRM). The impact of patterns on quality attributes was determined by us (the authors), based on the description of patterns. To visually illustrate the impact on quality attributes, we used force resolution maps (FRM) [10]. Force resolution maps apply numerical values (e.g., -2 for strong negative impact) to indicate how a pattern behaves with respect to quality attributes.

Using an existing patterns catalogue means that our evaluation depends on the description of the patterns in this catalogue. If a quality attribute is not discussed in the patterns catalogue it does not mean that the quality attribute is not affected at all. However, if there is no explicit mentioning of a quality attribute we did not try to interpret the impact of patterns on quality attributes ourselves.

IV. DISCUSSION OF RESULTS

Due to space limitations we do not list the evaluation of all patterns but provide an aggregated overview.

- Service patterns: We analyzed 31 service patterns. For four patterns we could not determine the impact on quality attributes. Ten patterns have a negative impact on performance but only one pattern has a positive impact on performance. On the other hand, the change cycle of services is supported by four patterns.
- Service composition patterns: We analyzed 23 service composition patterns. Two patterns seem to have no impact on any S-Cube quality attribute. In contrast to service patterns, some service composition patterns have a positive impact on performance.
- Inventory patterns: We analyzed 24 service inventory patterns. Ten patterns have no obvious impact on S-Cube quality attributes. In particular, governance patterns (one type of inventory patterns) have no interaction with quality attributes.

A. Addressed Quality Attributes

Fig. 1 to Fig. 4 illustrate what quality attributes are addressed negatively ("-/--") or positively ("+/++"), and by what type of patterns. The bars in the figures show the number of patterns, separated by pattern types. Please note that Fig. 1 to Fig. 4 only list quality attributes found in patterns, rather than all S-Cube quality attributes. As mentioned before, the S-Cube QRM is described in a tree structure with parent and children nodes. Some patterns impact parent nodes in the S-Cube QRM, whereas some patterns affect the children attributes of these nodes. The quality attributes in the figures combine parent nodes (i.e., Performance, Cost in Fig. 1, Dependability in Fig. 2, and Security in Fig. 3 if directly addressed by a pattern; no parent quality attribute in Fig. 4) and children nodes. Parent nodes are marked with an asterix. Quality attributes to the right of a parent node are children of this node. Figure captions only include parent quality attributes ("other quality attributes" in the caption of Fig. 4 is a category in the S-Cube QRM).

We found 53 quality attributes (i.e., more than 65% of quality attributes in the S-Cube QRM) not addressed by patterns (e.g., accountability, traceability, data validity). However, we would like to emphasize that these quality attributes might be affected, but we did not find any indication for it. Patterns affect performance and scalability most: Thirty-two percent of the analyzed patterns negatively affect performance. On the other hand, 9% of the patterns support scalability.

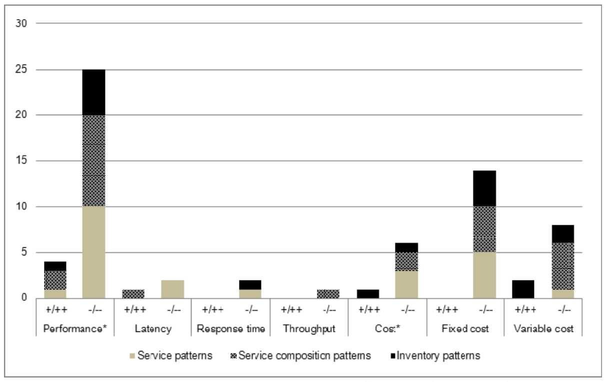

Figure 1. Impact on performance quality attributes and cost.

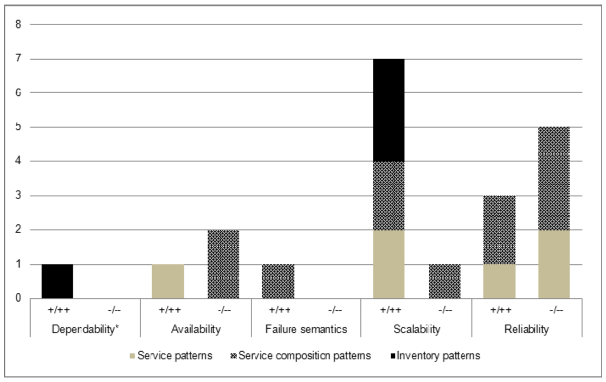

Figure 2. Impact on dependability quality attributes.

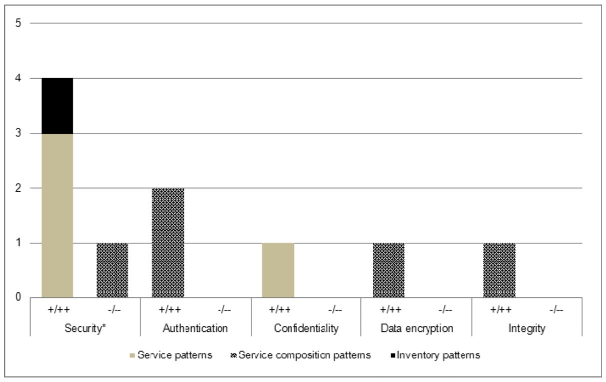

Figure 3. Impact on security quality attributes.

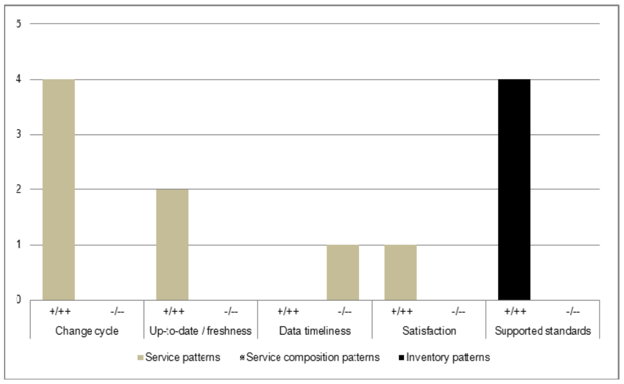

Figure 4. Impact on configuration and management, quality of use context, data-related, usability, other quality attributes.

B. *Additional Quality Attributes*

Some quality attributes addressed by SOA patterns could not be related to any quality attribute in the S-Cube QRM. Thus, we also mapped SOA design patterns to quality attributes from the ISO/IEC 9126 standard for software product quality. Note that this standard was not the only alternative quality attribute model, but during our analysis we found that additional quality attributes matched the ISO standard best. Maintainability was addressed by twelve patterns and interoperability by three patterns. Maintainability and interoperability are not explicitly mentioned in the S-Cube QRM. However, it may be possible to relate it to quality attributes included in S-Cube. For example, interoperability might be related to supported standards as included in S-Cube. However, the description of the pattern in the patterns catalogue and the quality attributes in S-Cube did not allow this.

C. *Patterns without Quality Attributes*

We identified 12 patterns (i.e., 15% of all analyzed patterns) that do not address any quality attribute, given their description in the patterns catalogue. No quality attributes were discussed in the documentation of those patterns. Several SOA patterns ("dual protocols", "canonical resources", "canonical expression", "canonical versioning", "contract denormalization", "concurrent contracts", "service messaging", "agnostic sub-controller") seem to describe implementation details and do not mention any quality attributes. For example, the pattern "dual protocol" (concerns the use of more than a single protocol to communicate with services) could be considered as an architecture pattern with impact on e.g., interoperability or modifiability, but no such details are given in the patterns catalogue.

The remaining patterns ("logic centralization", "service layers", "metadata centralization", "service encapsulation") describe general design principles (e.g., encapsulation) but do not mention an impact on quality attributes. For example, even though "service layers" is similar to the layers pattern [1] and therefore might have similar implications on quality attributes as the layers pattern (e.g., negative impact on performance), nothing is mentioned in the patterns catalogue. If no quality attribute is mentioned it could mean that the pattern has no "significant" impact on quality attributes.

D. *Applicability of Results*

- Design of service-based systems: Pattern-based architecting uses a pattern-driven approach for designing software. One key is to select patterns based on the key drivers of a system. Based on our results, patterns can be chosen according to their impact on quality attributes. Given the positive and negative impacts of patterns, trade-offs occur.
- Evaluation of service-based systems: Patterns can be used in architecture evaluations [11]. For each pattern found in the architecture, its impact on quality attributes can be determined and taken into consideration for architecture evaluation.
- Research on quality models: As shown for ISO 9126, some quality models are not applicable in practice and

are not suitable for measuring software quality [12]. Given that the patterns used in this research stem from industrial projects but quality attributes from the S-Cube QRM are not reflected in these patterns we conclude that there is a need for better quality models for service-based systems. In particular, quality models that stem from industrial practice are needed.

- Research on pattern catalogues and languages: New patterns catalogues should make the impact of patterns on quality attributes explicit to allow an informed pattern selection. Pattern languages are a group of patterns that address similar problems [6]. A pattern language for service-based systems could be proposed and extend initiatives for pattern-based SOA design. Pattern languages establish relations between patterns and describe the "big picture" of related patterns.

V. Threats to Validity

We only analyzed patterns from one patterns catalogue. Thus, we cannot claim completeness of the patterns covered. Furthermore, we only looked at patterns in isolation but not the combined use of patterns and pattern interactions. Even though the patterns catalogue presents "compound" patterns, no discussion is provided about the interaction of patterns with regard to the effect on quality attributes. Also, the patterns catalogue described patterns at different levels of abstraction or patterns that are not architecture patterns, but recurring functionality (e.g., Authorization, Authentication). Some of the patterns are certainly not architectural and some can be considered architectural depending on the context. Furthermore, as the pattern evaluation was done by us based on the pattern description, the evaluation of patterns might be subjective. We used the S-Cube QRM because it was specifically designed for service-based systems. Alternatively, we could have used the list of nine quality attributes proposed by O'Brien et al. [3] but we found these quality attributes covered by S-Cube. However, the structure of the S-Cube QRM caused some problems. For example, Security is a parent in the structure of the S-Cube QRM. Authorization and Authentication as children of Security and S-Cube considers Authorization and Authentication as classical quality attributes. On the other hand, the S-Cube QRM also states that Authorization and Authentication provide Security, i.e., Authorization and Authentication are means for providing Security, rather than quality attributes themselves. In S-Cube, Authentication is defined as "process of verifying that a potential partner in a conversation is capable of representing a person or organization". Furthermore, according to other software architecture literature, Authorization and Authentication are tactics [4], or means to achieve qualities such as Security, but not independent quality attributes of a system. In general, the definition of quality attributes in S-Cube is sometimes vague.

VI. Conclusions

This paper presents an attempt to provide insights on the impact of SOA patterns on quality attributes that are considered important for service-based systems.

The S-Cube QRM defines more than 70 different quality attributes for service-based systems. Some of them occurred many times in our study while others could not be found in any pattern. There could be several reasons why some quality attributes could not be mapped to patterns: First, it is very difficult to address all 70 quality attributes in the S-Cube QRM with only around 80 patterns provided in the patterns catalogue that we used. Second, the patterns catalogue did not particularly elaborate on affected quality attributes. The impact of patterns on quality attributes is described at a very high level only. S-Cube QRM on the other hand described quality attributes in detail.

Some patterns described in the patterns catalogue do not seem to have significant impact on quality attributes. However, in combination with other patterns they can drastically affect the quality attributes of a software system. Studying this kind of relationship is left for future work.

Acknowledgment

We thank Alexander Bograd for his help with this study. This research has been partially sponsored by NWO SaS-LeG, contract no. 638.000.000.07N07.

References

[1] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, Pattern-Oriented Software Architecture Volume 1: A System of Patterns. West Sussex, UK: Wiley, 1996.

[2] A. Gehlert and A. Metzger, "Quality Reference Model for SBA," S-Cube 2009.

[3] L. O'Brien, P. Merson, and L. Bass, "Quality Attributes for Service-oriented Architectures," in *International Workshop on Systems Development in SOA Environments* Minneapolis, MN: IEEE Computer Society, 2007, pp. 1-7.

[4] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice. Boston, MA: Addison-Wesley, 2003.

[5] S. Balasubramaniam, G. A. Lewis, E. Morris, S. Simanta, and D. B. Smith, "Challenges for Assuring Quality of Service in a Service-oriented Environment," in *2009 ICSE Workshop on Principles of Engineering Service Oriented Systems* Vancouver, Canada: IEEE Computer Society, 2009, pp. 103-106.

[6] P. Avgeriou and U. Zdun, "Architectural Patterns Revisited - a Pattern Language," in *10th European Conference on Pattern Languages of Programs* Irsee, Germany: Hillside, 2005.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software. Reading, MA: Addison-Wesley, 1994.

[8] T. Erl, SOA Design Patterns. Upper Saddle River, NJ: Prentice Hall, 2009.

[9] C. Mauro, J. M. Leimeister, and H. Krcmar, "Service-oriented Device Integration - An Analysis of SOA Design Patterns," in *43rd Hawaii International Conference on System Sciences* Honolulu, HI: IEEE Computer Society, 2010, pp. 1-10.

[10] J. Souza, S. Matwin, and N. Japkowicz, "Evaluating Data Mining Models: A Pattern Language," in *9th Conference on Pattern Language of Programs (PLOP)* Monticello, IL: Hillside, 2002, pp. 1-23.

[11] N. Harrison and P. Avgeriou, "Pattern-based Architecture Reviews," *IEEE Software,* vol. 28, pp. 66-71, 2011.

[12] H. Al-Kilidar, K. Cox, and B. Kitchenham, "The Use and Usefulness of the ISO / IEC 9126 Quality Standard," in *International Symposium on Empirical Software Engineering* Noosa Heads, Australia: IEEE Computer Society, 2005, pp. 126-132.