Enterprise Modeling Support for SOA Migration

Sybren de Kinderen and Monika Kaczmarek-Heß

Information Systems and Enterprise Modelling, University of Duisburg-Essen, Essen, Germany
{sybren.dekinderen,monika.kaczmarek}@uni-due.de

Abstract. The migration to a Service Oriented Architecture (SOA) is a complex undertaking in terms of aligning business and IT concerns as well as analysis of technical aspects. Conceptual modeling can be helpful for supporting SOA migration by (1) bridging the gap between business and IT concerns, and (2) analyzing the as-is and to-be IT infrastructures. We contribute language requirements derived from SOA migration literature, and extend an IT infrastructure Modeling Language, ITML, to support SOA migration. We illustrate the extended ITML with a documented SOA migration case.

Keywords: enterprise modeling, service-orientation, SOA migration

1 Introduction

Since its emergence and subsequent mainstream acceptance during the early 2000s [1, p. 75], adopting a Service Oriented Architecture (SOA) remains an important concern for many organizations [2, 3]. Service orientation (ideally) fosters organizational flexibility and agility by promoting (1) re-use of the functionality offered by services [5], (2) modifiability of (IT) functionality, which is realized by loose coupling [6, p. 64].

Successfully migrating an organization towards service orientation has however proven challenging [7, 8]. Conceptual modeling has the potential to play an important role in supporting SOA migration, in particular to foster (1) communication between business experts and IT experts [10], so as to ensure that IT functionality is driven by business concerns and vice versa; (2) to perform an in-depth analysis of the as-is and to-be IT elements [9, 11] so as to, e.g., identify functionality of legacy systems in need of wrapping [11]. Although various modeling languages exist that allow for expressing service orientation from various angles (cf. [12–14]), these languages often on purpose forgo the level of detail that is required in analysis of an IT infrastructure for the needs of SOA.

In order to address this gap, we focus on the following question: *What should be the scope and characteristics of a modeling language able to support SOA migration projects?* To address it, we first identify a set of requirements based on the analysis of SOA migration literature and of characteristics of SOA migration projects and then, extend an already existing modeling language, namely ITML [4], to account for missing aspects. The proposed extensions have been evaluated against the requirements and with an extensively documented SOA example.

13th International Conference on Wirtschaftsinformatik,
February 12-15, 2017, St. Gallen, Switzerland

de Kinderen, S.; Kaczmarek-Heß, M. (2017): Enterprise Modeling Support for SOA Migration, in Leimeister, J.M.; Brenner, W. (Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, S. 346-349

2 Modeling Support for SOA Migration: Requirements

'Migration' is usually understood as moving an existing operational system to a new technological or computing platform, while retaining the data and functionalities of the moved system (cf. [15]). SOA migration projects requires knowing at least the as-is and to-be states of the IT infrastructure and how these support business concerns (cf. [9, 11]).

Table 1. Requirements on a conceptual modeling language supporting SOA migration

RQ	*Requirement and Candidate Concepts*
	I Requirements For As-Is Models
1	The modeling language should allow for expressing IT landscape elements. Candidate concepts: Database, Database management system, Middleware, Server
2	The modeling language should allow for expressing the dependencies between IT landscape elements. Candidate concepts: uses, provides, runs on
3	The modeling language should account for non-functional properties of legacy systems. Candidate concepts: mission criticality, source code availability, impl. lang., code complexity
	II Requirements For To-Be Models
4	A modeling language should provide dedicated concepts that allow to model a service and its relevant types. Candidate concepts: Service, WebService, Interface
5	The modeling language should allow for relating a service to its underlying implementation, in accordance with the migration strategy. Candidate concepts: Wrapper, provides, runs on, uses
6	The modeling language should account for quality attributes of service oriented concepts. Candidate concepts: various QoS characteristics
	Overall Requirements
7	The modeling language should allow for expressing dependencies between the IT landscape and the organization action system. Candidate concepts: supports, context of use

The as-is state usually encompasses information about legacy systems. Legacy systems are systems that are usually hard to modify and expensive to maintain. However, at the same time these systems are often mission-critical and thus, must be operational at all times [9]. The to-be state reflects the service oriented design of the architecture. As understanding the as-is and to-be states is important for carrying out the migration, the aim of the modeling language should be to: (1) provide knowledge on the current state of the IT infrastructure with the focus on legacy systems (cf. RQ1 & RQ3) and (2) express the to-be state of the service orientation and reflect the changes that should be performed following the selected migration strategy (cf. RQ4).

For the as-is and to-be state IT infrastructure, we are interested in expressing the observable functionality of IT infrastructure elements, which translates into the requirements: *IT infrastructure elements* (RQ1) and their *interdependencies* (RQ2).

Furthermore, for any meaningful analysis of the possible behavior of IT infrastructure elements we need to analyze non-functional attributes [11] (RQ3).

Concerning the to-be IT landscape, the language should provide (rudimentary) expressiveness for service orientation. This entails to (1) express services and related subtypes, in addition to their quality attributes, through the language (RQ4), as well as to (2) relate these services to elements of the IT infrastructure (RQ5). The latter

relation is important for the aim of the language: to analyze how an IT infrastructure should be changed to realize the functionality offered by a service.

In addition, SOA migration should be considered from both an IT infrastructure and a business perspective (cf. RQ7). Business processes largely drive what is implemented in terms of IT support [8, 16], and vice versa.

3 Extended ITML

We now briefly illustrate SOA extensions of our language, called the extended IT Modeling Language (ITML), which is based on [4]. For illustration purposes, in Fig. 1 we modeled a to-be SOA of the ACME insurance company [6, pp. 541–578]. We focus on three features of extended ITML, illustrated by the labels 1-3 (cf. Fig. 1).

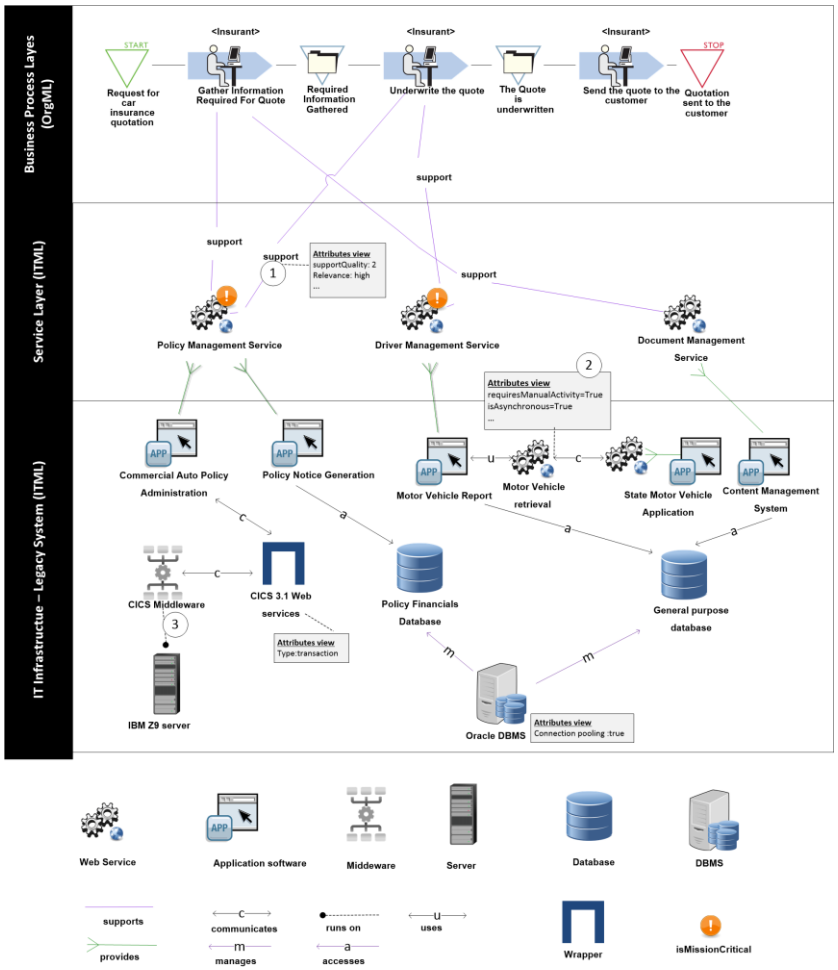

Figure 1 ACME insurance's desired service orientation, modeled in ITML

Extended ITML allows for (1) expressing, when used in tandem with the other MEMO languages [17], how IT functionality supports business processes (cf. RQ7). E.g., the web services "policy management service" and "driver management service" both support – Label 1 – the business process "underwrite the quote" (with *relevance:high*), (2) expressing non-functional attributes (RQ3&RQ6), such as – Label 2 – the necessity of a web service to support asynchronous communication, as well as a wrapper supporting bi-directional communication, and (3) inventorying relevant IT infrastructure assets and their relations (cf. RQ1&RQ2&RQ5), such as – Label 3 - the desire to keep using legacy bulk transaction processing functionality via a wrapper.

References

1. Erl, T.: Service-oriented architecture: concepts, technology, and design. Pears. Edu. (2005)
2. MacLennan, E., Van Belle, J.P.: Factors affecting the organizational adoption of service-oriented architecture (SOA). ISeB **12**(1) 71–100 (2014)
3. Alwadain, A., Fielt, E., Korthaus, A., Rosemann, M.: Empirical insights into the development of a service-oriented enterprise architecture. DKE **105** 39-52 (2016)
4. Heise, D.: Unternehmensmodell-basiertes IT-Kostenmanagement als Bestandteil eines integrativen IT-Controllings. Logos, Berlin (2013)
5. Razavian, M., Lago, P.: Towards a conceptual framework for legacy to SOA migration. In: ICSOC/ServiceWave Workshops, pp. 445–455. Springer (2010)
6. Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J.: Applied SOA: Service- Oriented Architecture and Design Strategies. Wiley Publishing (2008)
7. Hirschheim, R., Welke, R.J., Schwarz, A.: Service-oriented architecture: Myths, realities, and a maturity model. MIS Quarterly Executive **9**(1) (2010)
8. Rabelo, R.J., Noran, O., Bernus, P.: Towards the next generation service oriented enterprise architecture. In: IEEE 19th Int'l EDOC Workshop, pp. 91–100. IEEE (2015)
9. Khadka, R., Saeidi, A., Jansen, S., Hage, J..: Migrating a large scale legacy application to SOA: Challenges and lessons learned. In: WCRE 2013. pp. 425–432. IEEE (2013)
10. Razavian, M., Gordijn, J.: Consonance between economic and it services: finding the balance between conflicting requirements. In: REFSQ, pp. 148–163. Springer (2015)
11. Razavian, M., Lago, P.: A systematic literature review on SOA migration. Journal of Software: Evolution and Process **27**(5) 337–372 (2015)
12. The Object Management Group: Service oriented architecture Modeling Language (SoaML), version 1.0.1 (2012)
13. Terlouw, L.I., Albani, A.: An enterprise ontology-based approach to service specification. IEEE Transactions on Services Computing **6**(1) 89–101 (2013)
14. The Open Group: ArchiMate 2.1 Specification: Open Group Standard. The Open Group Series. Van Haren, Zaltbommel (2013)
15. Bisbal, J., Lawless, D., Wu, B., Grimson, J.: Legacy information systems: Issues and directions. IEEE Softw. 16(5) 103–111 (1999)
16. Papazoglou, M.P., Van Den Heuvel, W.J.: Service-oriented design and development methodology. Int J Web Eng Tech **2**(4) 412–442 (2006)
17. Frank, U.: The MEMO Meta modeling Language (MML) and Language Architecture. 2nd Edition. ICB-Research Report 43, University of Duisburg-Essen (2011)