Knowledge-driven SOA Migration

Maryam Razavian

supervised by Patricia Lago
Department of Computer Science, VU University Amsterdam, the Netherlands
m.razavian@few.vu.nl

Abstract. Migration of legacy assets to SOA embodies a key software engineering challenge namely rehabilitation of pre-existing enterprise assets into service based systems. Existing service analysis and design methods mostly focus on development of new services while they lack in transforming services from already existing enterprise assets. We argue that, a comprehensive SOA migration methodology is essential for creating a well-constructed SOA out of pre-existing enterprise assets. To this end, the necessary knowledge concerning the migration should be identified, captured, analyzed and generalized. We propose an approach for migration that exploits relevant knowledge to transform an existing legacy system into a new service-enabled system. For this purpose, we will devise a reference knowledge model capturing types of knowledge that drive SOA migration.

1 Introduction

Facilitating the reuse of existing business functions from legacy systems is considered as one of the key features of the service oriented paradigm. Many organizations aim at modernization of legacy systems to SOA to achieve the advantages offered by SOA and still reuse the enterprise assets embedded in the pre-existing legacy systems. Since early use of SOA, migration of legacy assets to services has caught a lot of attention in research and industrial community. However, it is mostly assumed among researchers and developers that, existing enterprise assets are made to act as services simply by creating wrappers and leaving the underlying implementation untouched. It is worth mentioning that, the main goal of SOA is to promote highly standardized, loosely coupled services to foster easy composition of distributed applications [1]. Unless legacy elements are "inherently" suitable for use as a service with the aforementioned characteristics (which usually are not), it takes a considerable effort to provide an enterprise asset's functionality through services. As a result, a comprehensive SOA migration methodology is of critical importance for creating a well-constructed SOA out of pre-existing enterprise assets.

Let us consider the following example:
A pre-existing legacy system stores and manages a patient's medical records. The existing knowledge about this legacy system includes the description of the provided functionality, the business processes the system supports (e.g. update medical record, analyze patient history to formulate a diagnosis), and the business value for the current/potential

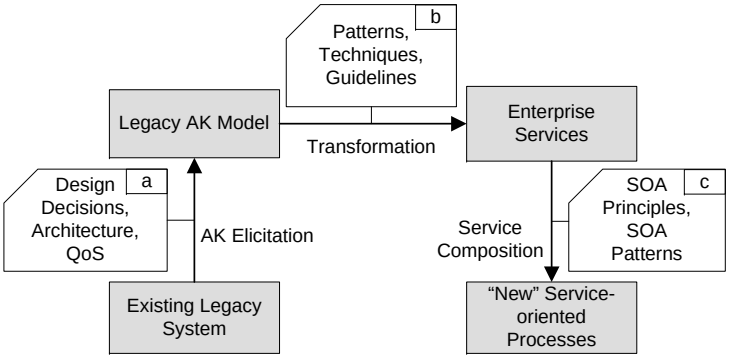

Fig. 1. SOA Migration Process

users (like the medical doctor) embedded in the legacy system. Such information can then be translated into business processes, business data and user added-value, which in turn can be associated with, respectively, task-centred services, data-centred services and end-user applications. For instance, if the knowledge regarding business data and functionalities associated to patients is not extracted, a smooth transition to patient data service is not possible. All the knowledge concerning the existing patient information system, the target SOA environment (e.g. types of services) and transformation techniques (e.g. wrapping) affect the migration process in terms of "what"is going to be migrated and "how" the migration is performed.

Despite its simplicity, this example already reflects essential requirements and challenging issues of the migration methodology from a knowledge management perspective:

- The main types of knowledge concerning the pre-existing legacy system should be identified (Box a, Fig.1)
- The knowledge concerning elements of service based systems should be extracted (Box c)
- Methods and guidelines should be devised for transforming legacy elements into types of services (Box b)

We argue that a comprehensive knowledge management approach is of great importance for SOA migration methodology. The reason behind this is that, knowledge management helps to rationalize the investigation of legacy assets as candidate services, isolate their properties and transform them into meaningful business services while adhering to service relevant aspects and challenges. This research is part of a project named the Service-enAbling PreexIsting ENterprISe Assets- Methodology (SAPIENSA) aiming at proposing a migration methodology for creating a well-constructed SOA out of pre-existing enterprise assets. Our contribution in this project is to effectively exploit architectural knowledge to migrate an existing enterprise system into a new service-enabled system.

2 Proposed Solution

In this section we further explain our proposed solution. To clarify the idea, we recall the example presented in Section 1. The following types of knowledge could shape the migration process of the medical information system: 1) the body of knowledge concerning the properties of existing system including lost abstractions (structural design, business processes, business rules, etc.) and know-how (design decisions, discarded alternatives, etc.) 2) patterns, templates, and methods that address altering and reshaping the extracted legacy elements to relevant type of services 3) SOA specific characteristics and properties. We aim at devising a reference knowledge model capturing types of knowledge that drive SOA migration. To this end, we propose to analyze different migration processes (from industrial cases) and extract the implicit and explicit types of knowledge, which shape the migration. We propose the phases depicted in Fig. 1 for the service-enabling methodology using architectural knowledge (AK). We argue that these phases facilitate extraction of a reference AK model as it helps gathering architectural insights and transforms them to well-constructed services. These phases along with their associated research challenges are discussed in the followings.

2.1 Architectural Knowledge Elicitation

The main goal of this phase is to identify the legacy elements that are relevant for migration to SOA and make the decisions regarding how to transfer to services. To this end, we aim at identifying, isolating and describing the knowledge about the typical "standard" elements that belong to pre-existing enterprise assets, and generalize the reusable knowledge. In order to address these aspects, the following research questions should be resolved:

- *RQ1) what are the knowledge elements that are relevant to be made explicit?* For instance, solution-related knowledge such as architecture, design decisions, etc., as well as problem-related knowledge including functional requirements (e.g. business processes and business rules) and non-functional requirements (e.g. privacy and security), should be externalized.

In order to enable an effective knowledge capturing and sharing, the knowledge extracted in this phase should be explicitly represented. Architectural knowledge models aiming at representing decisions in general are considered as a potential means of codifying the elicited knowledge. For instance, the AK core model developed in our previous work [2] is relevant for making AK explicit. Knowledge elements specific to an organization or an application domain should be defined as specific extensions to the core model described above. We consider service-oriented enterprise architectures as SOA extensions.

2.2 Transformation

The transformation phase embraces the actual migration of legacy assets to service based assets. The AK acquired during the AK elicitation phase will be used

as a fabric for creating new business services. This transformation could be in the form of reshaping design elements, restructuring the architecture and/or altering business models and business strategies. It should be noted that, each one of these forms of transformation belongs to a specific level of abstraction. For instance, a legacy element can be transformed to services by altering its encapsulation using wrapping techniques. In the same way, a composition of legacy elements can be transformed to a service or composition of them. At a higher level, an existing business model is transformed to a to-be business model based on new requirements as well as opportunities offered by service based systems. In all these cases the knowledge element extracted within AK elicitation phase is converted to another knowledge element in the SOA environment. Feasibility of each type of migration depends on many additional factors, such as past design decisions and knowledge about extra functional aspects like technological constraints or quality attributes. It becomes therefore even more important to capture the relevant AK in pre-existing and new systems in order to assess the feasibility of different transformation techniques. Recall that in this research work, we aim at isolating and describing the AK about the typical "standard" elements that belong to pre-existing enterprise assets; in the same way, we want to describe the AK about the "standard" elements of an SOA, including the architectural patterns relevant for SOA transformations like for instance service topologies and worst-case quality concerns. By carrying out a number of industrial case studies, we will define how we can typically transform a standard element of preexisting enterprise assets, into a "standard" element of an SOA. This will result in a number of transformations, as represented in the horizontal, continuous arrow in Figure 1. In this way, the "SOA migration decision making process" will be supported by a knowledge base of SOA transformations capturing necessary technical and nontechnical AK. In order to address all these aspects of transformation, following research questions should be addressed:

- *RQ2) what are the elements which drive the transformations?* For instance, past design decisions, quality attributes or technical constraints can have influence on the solution regarding the transformation problem.
- *RQ3) what are the possible types of transformation regarding different levels of abstraction?*
- *RQ4) how to represent the body of knowledge associated to transformation?*

2.3 Service Composition

Processes need to be designed in terms of constellations of the interacting services obtained in the transformation phase. This is supported in the service composition phase, which provides the ability to restructure, compose and decompose services. Certainly, bodies of knowledge addressing SOA environment, service association, message partnership, transaction ownership, etc., have to be captured. This phase is researched by another partner of the SAPIENSA project.

3 Related Work

Our work aims at enabling the migration process with knowledge capturing, sharing and reusing. Therefore, we position ourselves in the fields of architectural knowledge management and migration process. Accordingly, we discuss the related work in these two research fields. A vast body of work in the area of SOA migration mostly focuses on exposing legacy code as (web) services. Sneed proposes an approach that includes salvaging legacy code, wrapping the extracted code and making the code available as services [3]. Typically, the focus of these works is limited to implementation aspects of migration which usually covers techniques to alter a segment of legacy code to web services. A further family of approaches aims for covering the whole migration process. These approaches are comprised of two main sub-processes: top-down service development and bottom-up service extraction [4, 5]. The main goal of the bottom-up sub process is to support understanding of existing legacy system and extracting services from legacy code using code analysis and architectural recovery techniques. Within the top-Down process the target services are designed and implemented based on existing legacy components as well as new requirements in the problem domain.

Recently, there is an increasing focus on management of architectural knowledge including architectural design decisions and their rationale [6, 2]. However existing work focuses on capturing and representing the AK as such it is relevant for any application domain. To the best of our knowledge, there are limited results in identifying the architectural knowledge relevant for SOA. A specific type of AK is contained in architectural and design patterns, which document standard solutions to recurring problems in software design. Some research is emerging in identifying SOA patterns [7, 8], addressing several important aspects of service development and management. In these approaches, disciplined design decisions and compliance to SOA patterns are considered key criteria for successful development. Still, the documented knowledge mostly captures reusable technical solutions, while the related rationale or decision process that led to the solution is missing.

4 Research Plan

This research is carried out in a consortium of two universities and industrial partners. These industries provide us with case studies and give regular feedback. My research work will adopt an iterative incremental approach where the industrial partners provide industrial cases on migration of legacy systems to SOA. The main goal here is to identify the state of practice regarding the migration problem. More specifically, we are seeking to find out how migration process is performed in practice.

Also, we aim at identifying the kinds of activities that are performed and the types of knowledge that are captured. To this end, we plan to perform the following tasks:

- Elicit AK of pre-existing enterprise assets from case studies and model it as SOA extensions to the AK core model (RQ1)
- Devise from case studies the transformations to candidate services, SOA patterns and related AK (RQ2, RQ3, RQ4)
- Integrate candidate services into the target service-oriented system

After a startup phase defining the methodology to be used and the AK reference model for pre-existing systems and SOA, three iterations will occur (one for each project year). In each iteration, the cases carried out will be followed by a reflection period aimed at giving feedback on migration process and AK reference model, and defining the possible transformation methods. A final phase will consolidate all results in a generalized and reusable solution. The expected contributions of this research work are as follows:

- SOA model representing the AK relevant for SOA migration
- Patterns codifying architectural practices as ready-to-use solutions for the migration problem
- Prototypes and test beds supporting the reference knowledge model for SOA, including a pattern repository, a visual modeling environment and AK search

5 Acknowledgement

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research and Development (Jacquard) program on Software Engineering Research via contract 638.001.206 SAPIENSA: Service-enAbling PreexIsting ENterprISe Assets.

References

1. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal **16**(3) (2007) 389–415
2. de Boer, R.C., Farenhorst, R., Lago, P., van Vliet, H., Clerc, V., Jansen, A.: Architectural knowledge: Getting to the core. In: QoSA. (2007) 197–214
3. Sneed, H.M.: Integrating legacy software into a service oriented architecture. In: CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering, Washington, DC, USA, IEEE Computer Society (2006) 3–14
4. Zhang, Z., Yang, H.: Incubating services in legacy systems for architectural migration. In: APSEC '04: Proceedings of the 11th Asia-Pacific Software Engineering Conference, Washington, DC, USA, IEEE Computer Society (2004) 196–203
5. Andreas Winter, J.Z.: Model-based migration to service-oriented architecture. In: the International Workshop on SOA Maintenance Evolution (SOAM 2007), 11th European Conference on Software Maintenance and Reengineering (CSMR 2007). (2007)
6. Tyree, J., Akerman, A.: Architecture decisions: Demystifying architecture. IEEE Softw. **22**(2) (2005) 19–27
7. Zimmermann, O., Grundler, J., Tai, S., Leymann, F.: Architectural decisions and patterns for transactional workflows in SOA. (2007) 81–93
8. Erl, T.: SOA Design Patterns. Prentice Hall PTR, Upper Saddle River, NJ, USA (2009)