Toward Service Identification To Support Legacy Object-Oriented Software Systems Migration To SOA

Manel Abdellatif[1,2], Anas Shatnawi[2], Yann-Gaël Guéhéneuc[1], Hafedh Mili[2], and Jean Privat[2]

[1] Ecole Polytechnique Montréal, Montreal, Quebec, Canada
manel.abdellatif@polymtl.ca, yann-gael.gueheneuc@polymtl.ca
[2] Université du Québec à Montréal, Montreal, Quebec, Canada
shatnawi.anas@uqam.ca, moha.naouel@uqam.ca, mili.hafedh@uqam.ca, privat.jean@uqam.ca

Abstract

The migration of legacy software systems to Service Oriented Architecture (SOA) is a challenging task. On the one hand, legacy software systems, by their very definition, have architecture, design, and implementation problems. On the other hand, SOA is different from legacy architectures. One of the main challenges in modernizing legacy software systems is the identification of potential services with significant value. We want to support the migration of legacy object-oriented software systems to SOA. We base our support on software re-engineering techniques to extract from legacy software systems useful information for the service identification (SI) process. We aim (1) to provide good practices for SI through a taxonomy of SI approaches in the literature; (2) to propose an automatic SI technique based on re-engineering techniques. We built a taxonomy of SI approaches and drew a general classification of the different techniques in terms of the used inputs, the applied SI processes, the obtained outputs, and the applicability of the techniques. We also collected and evaluated several re-engineering tools used in SI research and several open-source applications migrated to SOA. The results of our analysis showed that there is no fully-automated SI approach in the literature. This motivates us to propose patterns for the identification process as a first step and then investigate how we can automatically identify potential services from legacy object-oriented software systems analysis based on those patterns and other heuristics.

1 Introduction

The evolution of software systems has become a central activity in many businesses. Legacy software systems are still a vital and important component in many businesses because the knowledge embedded in such systems is often hidden but of significant values. They cannot simply be removed or replaced as they effectively and accurately execute critical and complex business logic. Yet, they suffer from several drawbacks: maintenance cost, scalability, portability, etc. [11]. Therefore, there is a need for migrating legacy software systems to more flexible and modern platforms without losing their business values.

The migration of legacy software systems to Service Oriented Architecture (SOA) is one avenue for the modernization of legacy systems. SOA makes it possible to develop complex and inter-organizational applications by integrating services that are reusable, relatively independent, generally heterogeneous, and distributed. However, the migration of legacy software systems to SOA is a difficult and complex process that must consider many factors: the choice of a migration process, the service identification method, the quality measurements of the generated services, the implementation and integration challenges, etc.

The SI process is considered as the most challenging step of the overall migration process [9] as well as in software reuse [5]. It consists in identifying in legacy software systems potential,

reusable services that may have valuable business logic. The identified services must meet a range of expectations concerning their capability, quality of service, and efficiency of use [11].

The challenge of the SI process is to identify from legacy software systems potential services that can be developed in a cost-effective manner, are suitable for reuse, easy to maintain, and provide capability to customize the migrated applications by proper selection and orchestration of web services.

In our work, we want to develop an automatic approach for migrating legacy object-oriented software systems to SOA. Our work is based on software re-engineering techniques to extract from the legacy software useful information for the migration process. Our first objective is to provide a catalog of good and bad practices also called patterns and anti-patterns related to the SI process through the study of the state of the art SI techniques and the analysis of existing object-oriented software systems already migrated to SOA. Then, our second objective is to propose an automatic service identification technique based on legacy software systems re-engineering and our predefined patterns.

After reviewing the research works in the literature, we built a taxonomy of SI approaches and drew a general classification of the different techniques in terms of the used inputs, the applied identification processes, the obtained outputs, and the applicability of the approaches. We also collected and studied several open-source applications migrated to SOA with both legacy and migrated versions available. The results of our analysis show that there is no fully automated SI approaches in the literature. All SI approaches are either manual or require the intervention of human experts. These results motivate us to investigate best practices of SI. Then based on those best practices, we aim to find how we can automatically and effectively perform SI to efficiently assist the migration process of legacy object-oriented software systems to SOA.

The remainder of this paper is structured as follows. In Section 2, we outline our research objectives. In Section 3, we describe our methodology for service identification from object-oriented legacy software systems. In Section 4, we detail our preliminary results and describe our taxonomy of service identification as well as the collected migrated applications to SOA that we aim to analyse. In Section 5, we describe some related works to service identification. Finally, Section 6 concludes and outlines future work.

2 Research Objectives

We now list our research objectives and detail the research questions related to each one. We based our research strategy on an adapted version of the Goal Question Metrics model (GQM for short) [10]. For each research question, we define the method used to answer the question. In our work, we have two main objectives. We first aim to provide a catalog of patterns and anti-patterns for SI process to assess how a good migration to SOA could be. Then, based on the results of our first objective, we want to propose a fully automatic SI technique that can be used and integrated in the migration process to SOA.

2.1 Patterns and anti-patterns of SI process

The first goal of our study is to build a catalog of patterns for SI as well as anti-patterns to avoid in this process. Indeed, best practices of service identification are scattered in the literature and they have not been studied systematically nor on real-world systems. For this purpose, we want to answer the following research questions.

RQ 1: What are the patterns and anti-patterns of SI?
Method: Build catalog of patterns to follow as well as anti-patterns to avoid. The method consists in studying service identification approaches in the literature and drawing a classification of these techniques following several criteria: type of the used input, the applied process, type of generated outputs, quality of the results, etc. Studying the state of the art helps to know how mature the SI approaches are and helps to build a first catalog of best practices. We also collect and study existing software systems already migrated to SOA to assess how SI process is performed in practice. Consequently, we can study the gap between academia and industry in terms of the used techniques for SI.

2.2 Automatic SI based on Software Analyses

- **RQ 2.1**: How can we automatically identify services by analysing legacy systems?
  Method: Automatically identify reusable artifacts that can be represented as services from legacy code analysis. There is no need of any human expertise in the SI process and the generated web services should have good quality metrics (e.g, low coupling, high cohesion, optimal granularity, etc.). The defined catalog of service identification patterns should be used as a guideline in this step.
- **RQ 2.2**: How can we automatically perform a SI by service type?
  Method: Automatically identify reusable components from legacy software systems that can represent target types of services (e.g., business services, process services, utility services, etc.). Specific detection rules should be defined for each type of service.

3 Methodology

Our research methodology, which is depicted in Figure 1, consists of seven main stages. In the first stage, we study the state of the art and build a taxonomy of SI approaches based on software systems re-engineering. Then, we collect and analyse migrated open source applications to SOA. Based on the results of the two previous steps, we build a catalog of patterns and anti-patterns of SI process in the context of migration to SOA. Then, based on our catalog, we propose a fully automatic approach for service identification that focuses on software systems re-engineering. The service identification technique is applied on a call graph generated from legacy object-oriented software systems. Finally, a list of candidate web services is identified and has to be validated to estimate the precision and the accuracy of our service identification approach.

3.1 Taxonomy of SI Approaches based on Legacy Code Re-engineering

This section presents questions about how to collect and analyse information on the different SI approaches in the literature in terms of the used inputs, the applied processes, the type of outputs, and the applicability of the approaches. A systematic literature review is conducted to achieve this goal. We detail the research questions of our systematic literature review and their following goal in Table 1.

3.2 Analysis of Migrated Systems

The second step of our research methodology consists in collecting open-source object-oriented software systems that have been migrated to SOA. The goal is to understand how the migration

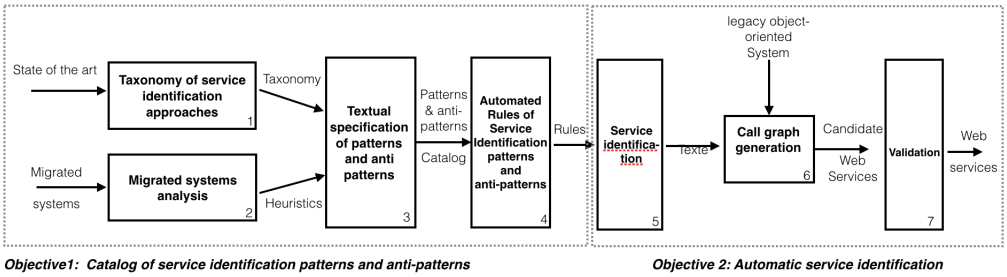

Figure 1: Research Methodology

Research question	Goal
What are the inputs for service identification techniques in the literature?	Collect and classify the inputs used in service identification approaches and explore the distribution of the used artifacts as well as the most targeted inputs.
What are the applied processes for service identification?	Classify the techniques used, extracting the quality metrics applied, knowing the analysis type and the automation degree of each approach.
What are the outputs for service identification?	Exploring types of service identification approaches in terms of services types and the technology used (SOAP, RESTful, SCA, etc.)
What is the applicability of the service identification approaches?	Estimate whether the approaches are reproducible and highly applicable in practice.
Who are the researchers working in the area of service identification?	Identify the role of each researcher in the scientific community and complement the process of monitoring scientific progress in the topic of service identification.
Where (country/university) service identification studies have been conducted?	Facilitate the decision process of which group to contact for collaboration.

Table 1: List of of the service identification taxonomy research questions and their goals

to SOA is done in practice and what are the functionalities that have been exposed to services. Both static and dynamic analysis will be applied to the collected legacy software systems. The quality of the generated services will be also studied. We aim to use for this purpose the SODA tool [2]. This tool performs an automatic detection of patterns and anti-patterns of targeted web services based on specific rule cards detection and service quality metrics.

3.3 Defining a Catalog of Patterns and Anti-patterns of SI

Based on the results of service identification taxonomy and the analysis of migrated software systems to SOA, we aim to build a complete catalog for patterns and anti-patterns of service identification in the context of migration to SOA. Textual specification of service identification patterns and anti-patterns will be collected from the analysis of the state of the art service identification techniques as well as the analysis of migrated systems to SOA. Then, we will specify rules that clearly describe each pattern by using static and dynamic properties.

3.4 Proposing a Fully-automatic SI approach based on the Analysis of Legacy Software Systems

The final step of our methodology consists in proposing a fully automatic service identification approach based on legacy software systems analysis. We rely on the previously defined catalog of patterns and anti-patterns of SI. As a preliminary step to service identification, we will identify a static call graph that represents the dependencies between program elements of a legacy software system.

Such graph provides the basis for computing the various metrics that we will need for our

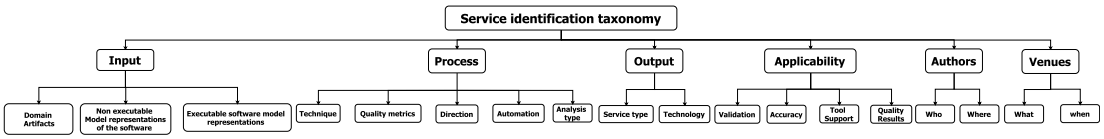

Figure 2: Taxonomy of service identification approach

service identification technique. We must consider all dependencies that exist between program elements, despite the different language, technology, and development environment mechanisms that combine such dependencies. We consider Java EE applications as target legacy software systems and develop a static analysis technique that captures a complete call graph of such applications [13].

4 Preliminary results

We will describe in this section the preliminary results of the first stage of our methodology. We are still working on the two first steps of our research methodology. After reviewing SI works in the literature, we built a general classification of the different techniques in terms of the used inputs, the applied identification processes, the obtained outputs, the applicability of the approaches, the authors of the approaches, and the venues of the proposals. The taxonomy of the different SI approaches is illustrated in Figure 2. Due to space limitation, we will not detail our taxonomy.

Then, to analyse how service identification and migration to SOA are performed, we searched open-source applications that have been migrated to SOA. We collected five object-oriented applications, which both legacy and migrated versions were publicly available. These applications are Ptidej [6], Weka [7], Moodle [1], Windup [3] and SODA [2]. The analysis of concrete migrated systems to SOA will help to assess how software architects and developers perform the service identification. It will also help us to investigate whether anti-patterns of service identification are present in those systems by analysing, for example, the quality of the generated services.

5 Related Work

Several service identification approaches have been proposed in the literature. For example, Jain et al. [8] proposed a spanning tree based approach to identify web services based on heuristic metrics and they considered static and dynamic relationships between classes. This approach is refined by applying a multi-objective Genetic Algorithm (GA) to solve the multi-objective optimization problem of web services identification. However, GA based approaches are time consuming and may not lead to an optimal solution. Zhang et al. [14] adopted a clustering analysis on legacy code to identify candidate web services. Both domain and business functions analysis are used for service identification. However, both architectural and requirement information are needed to be available for this approach as well as the software architect expertise to refine the resulting clusters. Adjoyan et al. [4] proposed a service identification technique based on object-oriented legacy code analysis. A mapping model from objects to services was used in their approach. A service is considered by [4] as a group of classes that are defined in the system source code. A clustering algorithm was used based on services quality metrics

(coupling, cohesion and composability, etc). However this approach was not validated on a big system and the proposed quality metrics are not accurate. Liu et al. [12] proposed a migration approach based on legacy system re-engineering to Restful Web services. Authors started from legacy code analysis to extract informative entities from different views: entity-relation diagrams, UML diagrams, technical documents and experts. Design URIs as well as resource types are identified under predefined rules. However, the evaluation of the semantics as well as the granularity of the generated web services is not taken into account in this approach.

6 Conclusion and Future Work

Service identification is considered as a key step in the process of migrating legacy software systems to SOA. Research in academia and industry mostly focused on defining overall guidelines for SI. There has been relatively little work into an effective and automatic approach to identify services with good quality metrics through the analysis of legacy software systems. We presented in this paper our research methodology to identify services based on software re-engineering techniques. We also briefly presented our taxonomy of SI approaches as well as a set of existing systems already migrated to SOA. We are still at the beginning of our research work. We will now propose a complete catalog of SI patterns and anti-patterns and an automatic SI technique to assist the migration of legacy object-oriented systems to SOA.

References

[1] Moodle tool. https://moodle.org, last viewed April 2017.

[2] Soda tool. http://sofa.uqam.ca/sodalast, last viewed April 2017.

[3] Windup tool. http://windup.jboss.org, last viewed April 2017.

[4] Seza Adjoyan, Abdelhak-Djamel Seriai, and Anas Shatnawi. Service identification based on quality metrics.

[5] Uday Apte, Chetan S Sankar, Meru Thakur, and Joel E Turner. Reusability-based strategy for development of information systems: implementation experience of a bank. *MIS Quarterly*, pages 421–433, 1990.

[6] Yann-Gaël Guéhéneuc. Ptidej: Promoting patterns with patterns. In *Proceedings of the 1st ECOOP workshop on Building a System using Patterns. Springer-Verlag*, 2005.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[8] Hemant Jain, Huimin Zhao, and Nageswara R Chinta. A spanning tree based approach to identifying web services. *International Journal of Web Services Research*, 1(1):1, 2004.

[9] Ravi Khadka, Amir Saeidi, Slinger Jansen, and Jurriaan Hage. A structured legacy to soa migration process and its evaluation in practice. In *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2013 IEEE 7th International Symposium on the*, pages 2–11. IEEE, 2013.

[10] Heiko Koziolek. Goal, question, metric. In *Dependability metrics*, pages 39–42. Springer, 2008.

[11] Grace Lewis, Ed Morris, Liam O'Brien, Dennis Smith, and Lutz Wrage. Smart: The service-oriented migration and reuse technique. Technical report, DTIC Document, 2005.

[12] Yan Liu, Qingling Wang, Mingguang Zhuang, and Yunyun Zhu. Reengineering legacy systems with restful web service. In *2008 32nd Annual IEEE International Computer Software and Applications Conference*, pages 785–790. IEEE, 2008.

[13] Anas Shatnawi, Hafedh Mili, Ghizlane El-Boussaidi, Anis Boubaker, Yann-Gael Guéhéneuc, Naouel Moha, Jean Privat, and Manel Abdellatif. Analyzing program dependencies in java ee applications. In *International Conference on Mining Software Repositories (MSR2017)*. IEEE, 2017 (to appear).

[14] Zhuopeng Zhang, Ruimin Liu, and Hongji Yang. Service identification and packaging in service oriented reengineering. In *SEKE*, volume 5, pages 620–625, 2005.