The Journal of Systems and Software 108 (2015) 134–147

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Automatic service derivation from business process model repositories via semantic technology

Henrik Leopold[a,*], Fabian Pittke[b], Jan Mendling[b]

[a] VU University Amsterdam, The Netherlands
[b] WU Vienna, Austria

ARTICLE INFO

Article history:
Received 8 May 2014
Revised 2 June 2015
Accepted 3 June 2015
Available online 10 June 2015

Keywords:
Service derivation
Process model repositories
Semantic technologies
Natural language analysis

ABSTRACT

Although several approaches for service identification have been defined in research and practice, there is a notable lack of fully automated techniques. In this paper, we address the problem of manual work in the context of service derivation and present an approach for automatically deriving service candidates from business process model repositories. Our approach leverages semantic technology in order to derive ranked lists of useful service candidates. An evaluation of the approach with three large process model collection from practice indicates that the approach can effectively identify useful services with hardly any manual effort. The evaluation further demonstrates that our approach can address varying degrees of service cohesion by applying different aggregation mechanisms. Hence, the presented approach represents a useful artifact for enabling business and IT managers to quickly spot reuse potential in their company. In addition, our approach improves the alignment between business and IT. As the ranked service candidates give a good impression on the relative importance of a business operation, they can provide companies with first clues on where IT support is needed and where it could be reduced.

1. Introduction

Service-oriented Architecture has been discussed for roughly a decade as a concept to increase the agility of a company in providing goods and services to external partners and organizing internal operations. In this context, a service is understood as an action that is performed by an entity on behalf of another one, such that the capability of performing this action represents an asset (O'Sullivan et al., 2002). The focus on services is supposed to improve business and IT alignment as, for instance, the degree of coupling between business and IT architecture is reduced and the effect of changes becomes more transparent.

In the past, many approaches for identifying services from various input sources have been defined (Kohlborn et al., 2009). Among others, these approaches leverage requirements documents (Adamopoulos et al., 2002; Chang and Kim, 2007), different types of conceptual models (Azevedo et al., 2009; Yun et al., 2009; Lee et al., 2008), and source code (Aversano et al., 2008; Chen et al., 2009; Zhang et al., 2005). Due to their general availability, particularly process models have turned out as a valuable source for identifying services. However, a core problem is that many of the approaches building on process models focus only on single steps of the service derivation process and still require a considerable amount of human effort to derive a promising service candidate. Taking the large size of process model repositories in practice into account (Rosemann, 2006; Houy et al., 2011), this means that these techniques do not scale up to the size of a whole company as the manual inspection of hundreds or even thousands of process models does not represent a feasible solution. Hence, there is a strong need for techniques that can automate the process model-based service identification as far as possible.

In this paper, we follow up on a proposal to address the problem of manual work in the phases of service derivation based on process models (Leopold and Mendling, 2012). We consider a situation in which an extensive set of process models is available that describes the company's processes at a operational level (Rosemann, 2006; Dijkman et al., 2012). To provide a technique that is widely applicable, we do not require any complementary data or artifacts beyond the process model collection. By building on the methodological considerations from (Sanders, 2009), we present an approach for the automatic derivation of service candidates, augmented with a set of metrics that provide initial clues about priorities. Recognizing that process models in practice are often suffering from heterogeneous terminology (Pittke et al., 2013), our approach extends the work from (Leopold and Mendling, 2012) by leveraging semantic technology. As a result, we are able to derive services based on semantic

* Corresponding author. Tel.: +31 20 59 87767.
E-mail addresses: h.leopold@vu.nl (H. Leopold), fabian.pittke@wu.ac.at (F. Pittke), jan.mendling@wu.ac.at (J. Mendling).

http://dx.doi.org/10.1016/j.jss.2015.06.007
0164-1212/

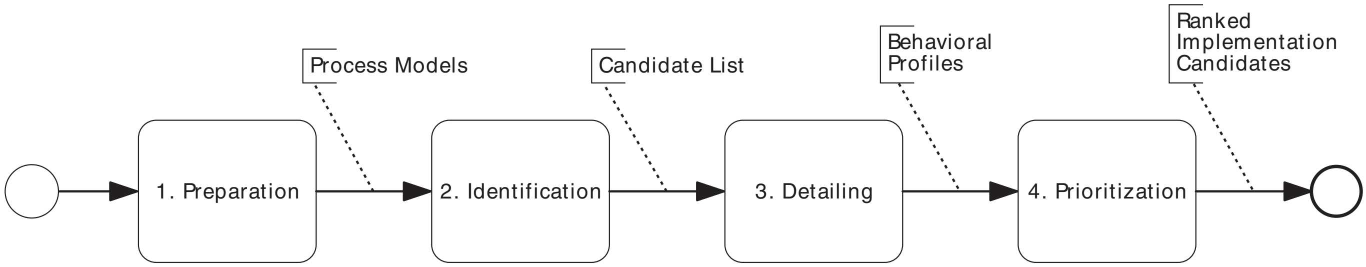

Fig. 1. The four phases of service derivation, adapted from (Kohlborn et al., 2009).

relationships and are no longer bound to trivial string comparisons. The approach is meant as a decision support tool for business and IT managers to quickly spot reuse potential in their company. In this way, the approach aims to speed up service derivation drastically, and to easily scale up to large sets of process models of the whole company. We use three large process model collections from practice to demonstrate the capabilities of our approach.

The rest of the paper is structured as follows. In Section 2, we give an overview of existing service derivation approaches and discuss shortcomings of available approaches. In Section 3, we present the service derivation approach on a conceptual level. Section 4 discusses the results of testing our prototypical implementation on three large process model collections from practice. Section 5 discusses the implications of our work. Finally, Section 6 elaborates on the limitations of the presented approach before Section 7 concludes the paper.

2. Background

In this section, we provide an introduction into the topic of service derivation. First, in Section 2.1, we give an overview of current service derivation approaches. Subsequently, in Section 2.2, we point out the shortcomings of current approaches in order to highlight the need for a novel process model-based service derivation technique.

2.1. Service derivation approaches

For deriving services, various approaches have been proposed in prior research. Many of them explicitly differentiate between business and software services. This distinction is brought forth from different perspectives. A business service is understood as a *specific set of actions that are performed by an organization* (Feuerlicht, 2005), while a software service describes a *part of an application system that is utilized by several entities independently* (Kohlborn et al., 2009). The concept of a business service puts more emphasis on the economic perspective as the software service is more related to information technology. Typically, the derivation of business services tends to take more of a top-down approach, i.e., it starts with a rather general analysis of the business by, for instance, investigating the value chain (Bell, 2008; Erl, 2005; Ramollari et al., 2007; Gu and Lago, 2010). The derivation of software services, by contrast, is rather bottom-up. It often builds on the analysis of concrete artifacts such as source code or system components. This distinction between business and software services is also apparent in many of the methodological contributions on service derivation.

The methodological contributions can also be considered from the perspective of the overall service derivation process (Kohlborn et al., 2009). Fig. 1 depicts the service derivation process consisting of four phases: preparation, identification, detailing, and prioritization. The derivation of services usually starts with a *preparation phase*. In this phase, an information base for the service analysis is established. This information base may include different types of business documents such as enterprise architectures, organizational structures, or business processes. The subsequent *identification phase* is concerned with identifying capabilities and service candidates. In the following *detailing phase*, the relationships and interactions between services are identified. This includes the detection of overlaps with existing services and the proper incorporation of new services into the existing SOA landscape. Finally, the *prioritization phase* is utilized to decide which services should be considered for implementation and with which priority.

Table 1 gives an overview of existing service derivation approaches. On the one hand, it shows the main input type, the degree of automation, and whether the approach is targeting software services (SWS) or business services (BS). On the other hand, we also consider the capabilities of the respective approach to support specific phases of the service derivation process. Altogether, we differentiate between three main input types for service derivation techniques: conceptual models, application data, and general requirements and capabilities.

In general, service derivation techniques building on *conceptual models* pursue the strategy of clustering the comprised elements in order to obtain a set of service candidates. Depending on the type of the conceptual model, different techniques are employed. One of the most frequently used artifacts in this context is the process model. Against the background of the multitude of automatic process model analysis techniques, it is surprising that many of the identification techniques building on process models suggest a manual analysis (see Azevedo et al., 2009; Erradi et al., 2007; Jamshidi et al., 2008; Klose et al., 2007; Sewing et al., 2006; Zimmermann et al., 2004; Kohlmann and Alt, 2007). Typically, they propose different heuristics on how to derive service candidates from process model activities. For instance, Azevedo et al. (2009) suggest the grouping of activities based on the model structure. By contrast, Klose et al. (2007) define an evaluation template for considering each activity in detail. Although many techniques build on the manual analysis, there are also techniques available providing automatic support. For instance, Yousef et al. (2009) use an ontology to generate a service model from a set of process models. Similarly, Bianchini et al. (2009) employ the lexical database WordNet and a reference ontology to identify component services. Kleinert et al. (2013) take a different perspective by employing a tree-based structure, a so-called Refined Process Structure Tree (RPST), to identify process regions that represent suitable service candidates. Dwivedi and Kulkarni (2008) introduce heuristics that also consider the hierarchical relationships among processes. Even though these approaches make use of automated techniques, the scope of the automation is limited to a particular set of steps. Moreover, they cover the phase of service identification only.

In addition to the numerous derivation techniques focusing on process models, there are also approaches building on other conceptual models. For instance, Lee et al. (2008) derive services from feature models by grouping features according to their binding time. In a similar way, Jain et al. (2003) employ a spanning tree and group the classes of object models in order to identify web service candidates. An alternative approach is taken by Kim and Doh (2007). They define a rule-based approach to derive service interfaces from use-case diagrams. Similar to process model-based approaches, the latter