Enriched Service Descriptions Using Business Process Configurations

Mohammed AbuJarour, Ahmed Awad
Hasso-Plattner-Institut
University of Potsdam, Germany
{firstname.lastname}@hpi.uni-potsdam.de

Abstract: Service descriptions play a crucial role in Service-oriented Computing (SOC), e.g., for service discovery, service selection, service composition, etc. However, it has been observed that service providers – the main source of service descriptions – typically release poor service descriptions, i.e., mostly technical-oriented descriptions. Several approaches have been proposed to tackle this problem by enriching poor service descriptions with additional information from other sources, e.g., communities or domain experts. In this work, we propose a novel approach to generate additional information about web services based on the configurations of their consuming business processes. For instance, we can extract annotations and context information for web services based on the configurations of their consuming business processes. In this paper, we introduce our proposed approach and its architecture based on the open-source online modeling platform Oryx and our public service registry Depot.

1 Introduction: Service Descriptions

Service-oriented Architecture (SOA) incorporates the main three roles of service-provider, -consumer, and -registry. Typically, service providers register their offered web services in one or more service registries that act as brokers between service providers and consumers. Such registries are then used by service consumers to discover web services that satisfy their (business) needs. Selected web services are then used as part of business processes (BP) running inside organizations, such as making a health insurance contract, selling items online, booking flights and hotels for journeys, etc.

Typically, service consumers maintain repositories of business process models (BPM) for their daily activities, e.g., add new client, offer special promotions, etc. Each business process is composed of a set of tasks. Some tasks are manual tasks that are performed by employees, whereas others are service tasks performed through web services. Building a new BPM incorporates three steps: 1) creating and labeling its tasks 2) determining the interaction between them, i.e., data and control flow 3) configuring the created model, i.e., *selecting* web services to perform service tasks. The created BPM is then stored in the consumer's repository for similar future requests.

Using the aforementioned scenario in practice involves several challenges, e.g., service discovery, service selection, BP configuration. Service discovery has been one of the main challenges in Service-oriented Computing (SOC). Several approaches have been proposed

149

to tackle this challenge, e.g., semantic web service discovery [KLKR07]. However, such approaches assume the existence of rich and correct service descriptions. In practice, this assumption is not realistic because it has been observed that service providers release poor service descriptions [SWGM05]. Additionally, the assumption that service consumers' queries during service discovery can be expressed formally using a query language is not realistic due to the increasing complexity of business needs and web services [Sun05].

During service discovery, a list of *candidate* web services are returned to the service consumer as a result for her submitted query. Choosing a particular web service to invoke from this list of candidates is known as service selection [SS04]. Service selection has become a complex task due to several factors, such as lack of rich service descriptions, increasing number of web services, increasing complexity of web services and modern business needs, etc. However, it has been shown that additional information about web services, e.g., annotations, helps meet this challenge [BTR10].

Business process configuration is the service selection step for service tasks in BPMs. In this step, each service task is assigned a web service to execute it. In practice, process engineers configure tasks manually using static lists of services they know already. To perform this task dynamically, process engineers require sufficient information about web services to configure their business process models accordingly. Technical-oriented information only is not sufficient, because it does not fit their backgrounds and knowledge [SBE07]. The "Internet of Services" initiative introduced the concept of "Business Services" to emphasize this requirement, namely, service descriptions should be understandable by business people, as well [SAP09].

In this work, we introduce a novel approach to use business process configurations to learn additional information about web services used in these processes. We have already introduced an approach to discover relationships among web services based on such configurations [AA11]. In particular, we consider extracting annotations for web services based on business processes that use them.

1.1 Research Context: Lack of Rich Service Descriptions

Services are typically parts of one or more distributed business processes. Identifying tasks in a BP that should be implemented as web services is a system design decision. In general, system designers follow a bottom-up or top-down approach. In the bottom-up approach, existing web services are composed to achieve more complex tasks. This composition is repeated at higher levels until the required BP is achieved. On the other hand, the top-down approach means decomposing the complex task of the BP in question into sub-tasks iteratively until simple tasks are reached. Each simple service task is then implemented as a web service. The best practice is to follow a mix of both approaches to avoid high costs of the top-down approach and inflexibility of the bottom-up approach [Jos07]. In our approach, we follow this best practice, where we allow BP designers to get an overview of existing, relevant web services, and give them the chance to add more tasks that reflect their requirements.

The fact that web services are typically parts of one or more distributed business processes

150

brings the challenges of service discovery and selection to business processes. Business processes and web services are usually separated and each of them is investigated individually. In this work, we introduce a novel approach to use business process configurations to derive additional annotations for the used web services from the service consumers' perspective (business view) to enrich their technical service descriptions released by their providers.

Several limitations can be identified in the current, traditional settings used in modern distributed information systems. In this work, we target the following limitations:

- **Lack of rich service descriptions**: This limitation has been identified in both industry and research [BTR10, KTSW08]. Semantic web services tackle this problem by means of ontologies, but such approaches are not so common in practice [Bos08].
- **Insufficient criteria for service selection**: Service consumers take several factors into account when they select web service(s) they want to use, e.g., service quality, provider's reputation, context in which the service is typically used, etc. Such information is usually provided by third parties, namely, service brokers (registries). However, context information is typically ignored.
- **Complex configuration of business processes**: Creating business process models is labor-intensive due to the increasing complexity of modern business needs. Additionally, the configuration of business process models is error-prone due to the increasing complexity of web services and because it is usually a manual task .

1.2 Contributions

The contributions of this work are:

1. A collaborative, incremental, and practical approach to enrich web service descriptions with annotations generated from the configurations of business processes that consume these web services.
2. Smooth configuration of business process models by enabling context-aware service selection.
3. Enhanced service discovery using both technical and business perspectives of service descriptions that are gathered from service providers and business process configurations, respectively.

The rest of this paper is organized as follows: In Sec. 2, we discuss potential application domains for our approach. An overview of our approach is presented in Sec. 3. Further implementation details and application scenarios are given in Sec. 4. Related work is summarized in Sec. 5. We close this paper with a summary and future work in Sec. 6.

151

2 Application Domains

Sharing information about business processes and web services used to achieve them is not usually desired by service consumers. This information might be considered one aspect of their competitive advantage. Nevertheless, our approach can be valuable in several scenarios and application domains, in particular, where there is a high potential for collaboration and a low potential for competition among service consumers, such as:

1. **Government services:** The number of governmental web services has been increasing steadily. For instance, the municipalities of the Netherlands have decided to use a common infrastructure to run their business processes [GWJV$^+$09]. Several similar cases have also evolved in other European countries, such as Germany and UK. We give a detailed use case from this domain in Sec. 4.3.

2. **Education:** Web services are usually used in education to share information and offer computationally-intensive tasks. For instance in Life Sciences, web services are used to provide information about Genes, and extract protein encodings from natural text [1]. As collaboration between educational institutions is expected instead of competition, our approach can be applied in this application domain.

3. **Online modeling platforms:** Due to the increasing popularity of the `Software -as-a-Service` trend, several online modeling platforms are available on the Internet, e.g., Oryx [2]. Such platforms enable model sharing and collaborative modeling among communities, e.g., distributed teams. In such cases, our approach can be applied inside each community.

4. **Quality-based service brokers:** Several service brokers (registries) provide quality-based service discovery and recommendation, e.g., `www.asperado.com`. Such brokers gather statistics about web services and measure their quality. These measures are then used to answer consumers' requests to find relevant web services with specific quality constraints. In this scenario, our approach can be applied directly because the involved service consumers agree already to provide information about web services they use to service brokers.

Due to policy and privacy issues, our approach cannot be applied to some application domains, such as banking. However, partial information about the used web services can be provided by service consumers, without leaking information about the internal structure of their business processes.

[1] Several examples available on: `http://www.biocatalogue.org`
[2] `http://oryx-project.org`

152

3 Annotations from Business Process Configurations

Several approaches have been proposed to enrich poor service descriptions with additional information, such as websites of service providers [ANC10], invocation analysis [AN10], ontologies [SWGM05], etc. In [ANC10] we have introduced an automatic approach to annotate public web services with additional information that are extracted by means of crawling from the websites of their providers. However, this recent approach is limited to the content provided by service providers on their websites. Therefore, we introduced another approach to generate dynamic tags for data web services based on the analysis of their invocations [AN10]. This recent approach is not applicable to functional and business web services. Hence, additional sources and types of information about web services are still required. In this work, we introduce our approach to use BP configurations to gather additional information about web services. These three sources of additional information about web services are integrated together, indexed, and used to enhance service discovery. This architecture is depicted in Figure 1.

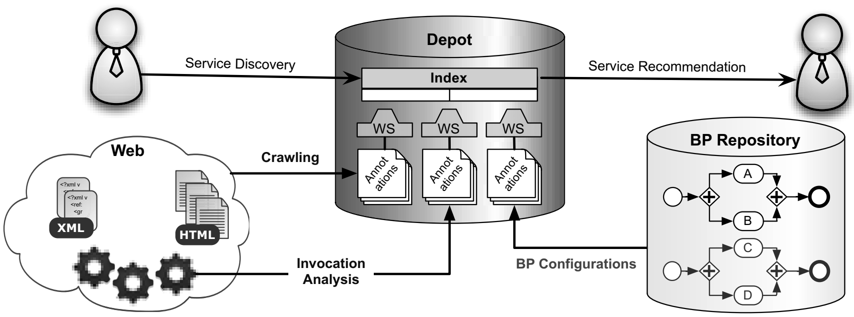

Figure 1: Multiple sources of information about web services to enhance service discovery

3.1 Approach Overview

The main goal of our approach is to enrich web service descriptions with annotations generated from the configuration of business processes that consume these services. To achieve this goal, we link business processes and web services. We give an overview of our approach to achieve this linkage in Figure 2. The scenario starts when a business process designer creates a new BPM. At that point, the designer gives a descriptive name and summary for the new process (Step 1), e.g., *establish a company in Germany*. Behind the scene, a request is sent to a service registry to find relevant web services that have been used in similar models (Step 2), e.g., *establish a company in UK*. The returned set of services is made available to the designer to select from. A by-product of this step is to accelerate the process designing step (Cf. Sec. 1.1).

However, the process designer might not use all web services suggested by the service registry in the new model. She introduces new tasks to express their particular business

153

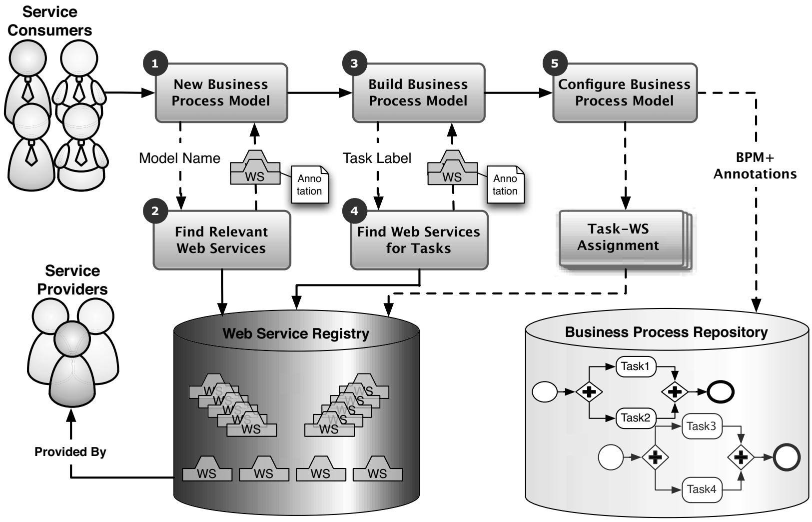

Figure 2: A high-level overview of our approach to link BPs and web services

needs (Step 3), where each task is given a label to identify it. This label is passed to the service registry to find potential web services that can perform the required functionality (Step 4). For each new task in the model, a list of candidate web services is returned to the process designer. Each web service is associated with a set of annotations that explain its functionality. These annotations are extracted and generated automatically from the websites of corresponding service providers [ANC10]. The new BPM is finalized when each service task is configured by assigning a web service to execute it (Step 5). With the finalized BPM, two types of information can be generated, namely, task-web service assignment and annotated BPM.

The task-web service assignment is passed from the modeling framework to the service registry. Task labels and documentations are then extracted and their assigned web services are annotated with this extracted information. These labels and documentations are created by service consumers that represent the application level, i.e., business people. Whereas, the initial annotations are extracted and generated from the content released by service providers, i.e., technical people. Additionally, this assignment and the title of the created BPM are used to determine the context(s) where these web services are typically used. This information is then used to enable context-aware service selection for similar cases in future BPMs.

The tasks in the created BPM are automatically annotated with the annotations of the web services they are bound with. The result is an *annotated* BPM. These annotations are crucial for BPM lookup because not only task labels are used to index and find tasks, but enriched annotations are also used to achieve this goal [Awa07].

This approach to enrich web service descriptions is collaborative and incremental. With each newly-created and configured business process model, the service registry gains addi-

154

tional annotations (e.g., labels) and information (e.g., context) about the used web services. These annotations and information are used to provide enhanced service discovery and enable smooth business process configurations due to context-aware service selection in later business process models. These advantages speed up the process design. Our approach is beneficial for both service registries and service consumers. Moreover, we believe that this is a practical approach to build rich semantic descriptions of web services, which can be considered as an additional step towards semi-automatic ontology construction. With such enriched service descriptions, more sophisticated approaches for service composition can be applied [AAMZM04].

It is worth mentioning that our approach does not suffer from a cold start. Available web services are already enriched with annotations that are collected automatically from the websites of their providers. The goal of our approach is to use these annotations to smooth BP configurations and generate additional annotations for them from the configurations of BPs that use them to reflect the service consumers' (business) perspective, as well.

3.2 Formal Model

A service registry usually contains a collection of web services. Each web service, WS_i, has one or more operations. An operation, OP_j, is given a $name$ by its developer, e.g., `GetWeather`, and a list of annotations, A (by service registry). An annotation of an operation, a, can be a short sentence or several paragraphs describing that operation. These definitions can be expressed formally as follows:

- Web service: $WS_i = \{OP_1, \cdots, OP_n\};\ n \geq 1$
- Operation: $OP_j =< name, A >;\ A : annotations$
- Annotations: $A = \{a_1, \cdots, a_m\};\ m \geq 1, a : annotation$

A business process, BP_k, has a list of tasks. Each task, T_l, has a label, documentation, and an optional operation, *op*, that executes that task if it is a service task. Formally, these definitions can be expressed as follows:

- Business Proces: $BP_k = \{T_1, \cdots, T_s\};\ s \geq 1,\ T :\ task$
- Task: $T_l =< label, doc, op >;\ doc : documentation,\ op :\ operation$

Configuring a BP can be expressed as a function that takes a BP and assigns an operation to each *service* task in that BP. This function is defined formally in Equaion 1[3].

$$Conf : BP \rightarrow \{OP_1, \cdots, OP_y\};\ \forall T_b \in BP : \\ T_b.op = \begin{cases} OP_c & \text{if } T_b \text{ service task; } OP_c \in \{OP_1, \cdots, OP_y\} \\ none & \text{otherwise} \end{cases} \quad (1)$$

The newly-introduced effect of this configuration function is expressed formally in Equation 2. The label and documentation of each configures service task are appended to

[3]We use the dot notation T.op to refer to the *operation* property of task

155

the annotation set of its assigned operation. These added annotations usually reflect the consumer's (business) perspective of these operation. This perspective complements the providers' (technical) perspective.

$$T_x.op.A = T_x.op.A \cup T_x.label \cup T_x.doc \tag{2}$$

4 Implementation and Application Scenarios

We have implemented a prototype that realizes our approach to enrich service descriptions using business process configurations. In this section, we give details about the implementation of this prototype that integrates *Oryx* – an open-source business process modeling platform and repository – and *Depot* – a public web service registry. Our prototype is available at: `https://www.hpi.uni-potsdam.de/naumann/sites/bpws/`. The front-end of our prototype is Oryx, whereas Depot represents the backend. An end user, such as business process designer, interacts with an Oryx front-end to model her business processes. Behind the scene, Oryx contacts Depot to find relevant web services for each particular business process and its individual tasks.

4.1 Oryx: A Modeling Platform

Oryx is an open-source, web-based graphical modeling tool and repository for business process models [DOW08]. Using a standard web browser, users can login to Oryx and create, modify and share business process models. Oryx supports a plethora of modeling languages, e.g, BPMN, Petri nets, UML, etc.

Business process modeling languages are defined within Oryx by means of stencil sets. A stencil set describes the visual appearance of a model element, either node or edge, the set of properties it has and connectivity rules to other model elements. The user can define a stencil set extension to adapt a specific stencil set to a certain modeling domain. We use this feature to populate a process model with the set of web services returned from the service registry. In Oryx, a task has several properties, such as label, documentation, type, implementation, status, etc. During the configuration task, the implementation of each service task should be specified.

4.2 Depot: A Public Service Registry

Depot is a service registry with multiple sources of information about web services, e.g., service providers, invocation analysis, service consumers[ANC10]. The information that Depot gathers is integrated into unified service descriptions that are used to provide enhanced service discovery. Depot uses an automatic approach to collect public web services

156

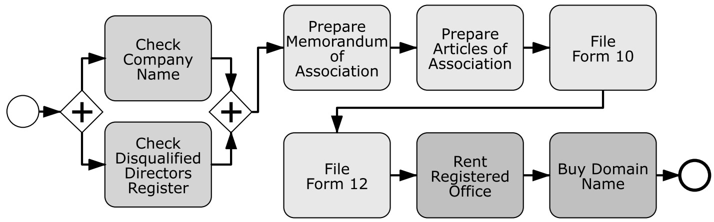

Figure 3: Example process model of establishing a UK limited company

from the websites of their providers and annotate them [ANC10].

Traditionally, service consumers provide ratings, feedback, and other quality measures about web services. Although such information is valuable, it requires additional effort and is provided by service consumers explicitly. The assumption that the majority of service consumers give this information is not realistic. In our approach, two types of information are provided by service providers *implicitly*, namely, task labels and documentations as annotations for their assigned operations, and titles of BPMs as context for web services.

4.3 Use Case: Establishing a Limited Company in UK

The European Union (EU) has been one of the main service markets. In 2006, the European Parliament and Council introduced the "Directive 2006/123/EC on services in the internal market" [Eur06], that aims to create a genuine internal market for services. Therefore, the directive seeks to remove legal and administrative barriers in order to make it easier for businesses to set up in other member states of the EU and to provide services cross-borders or on a temporary basis. The services offered by the UK Companies House [4] and other private service providers to establish a limited company in UK is an effort to apply this directive in UK.

Figure 3 shows a process model using BPMN for establishing a UK limited company. The first six activities of the process are services of the UK Companies House. In the beginning, it has to be checked whether the desired company name is not already in use and the directors are not disqualified from leading a company. For the remaining steps, electronic forms are provided in the Portable Document Format (PDF). Last two activities in the model are web services offered by private service providers, e.g., Buy domain name can be executed using the `whois` web service (`http://www.webservicex.net/whois.asmx`).

Using our approach, a process engineer can configure her BPM to establish a company in UK using the existing annotations for web services and their operations used in such a model. These annotations are generated from the websites of their providers and through invocation analysis. Although, they might be not rich enough, such annotation can be helpful in some cases. Generating additional annotations for such web services and operations

[4] `http://companieshouse.gov.uk`

157

from this BPM enrich their descriptions. In this particular use case, all used operations are associated with the context “establish a company in UK”. Additionally, each operation is annotated with the label and documentation of each task that uses it. For instance, the `whois` web service is annotated with “buy domain name”.

4.4 Our Approach in Action: Establishing a Company in Germany

According to the German Federal Ministry of Economics and Technology, establishing a company in Germany incorporates 9 major steps [5]. For instance, check the company name, notarize the articles of association and foundation agreement, notify the Office of Business and Standards, register at the Trade Office (involves check manager’s qualifications), etc. Some of these steps are similar to the ones involved in establishing a limited company in UK (Cf. Sec. 4.3), such as check the company name, check qualified managers, rent office, buy domain name. Figure 4 shows a screenshot of using our prototype to design a business process for establishing a company in Germany.

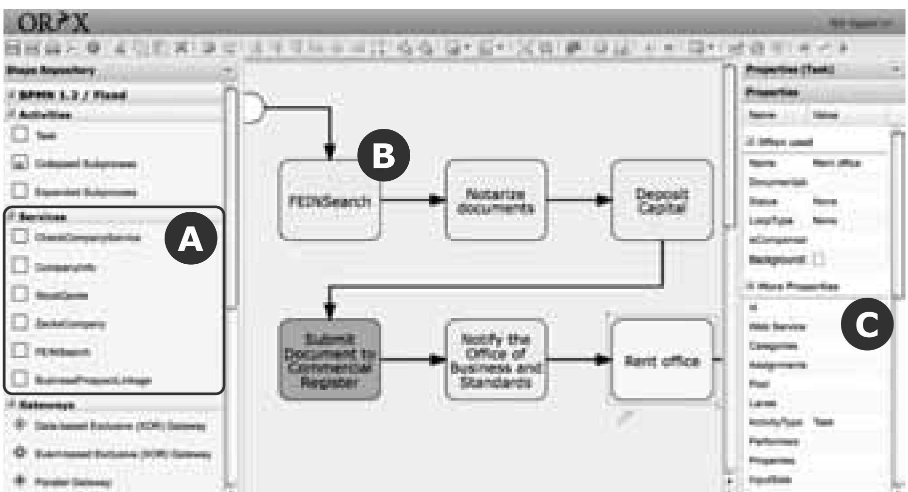

Figure 4: A screenshot of our prototype used to model the process of establishing a company in Germany. A: suggested web services, B: a pre-configured task from A, C: task properties

To create a new model for this process in Oryx, a proper title, such as “Establishing a company in Germany” is given by process designer. The area labeled with *A* in Figure 4 shows a list of web services discovered in Depot that have been used is similar contexts and are relevant to this process. For instance, the `whois` web services used in the UK example can be shown in this example despite the fact that there is no high similarity between terms appearing in “establish a company in Germany” and “whois”. Each of these web services is already configured and can be simply dragged-and-dropped to the design area. Indeed,

[5] `http://www.existenzgruender.de/englisch/`

158

the task labeled with *B* is an example of such pre-configured tasks.

Saving this model adds additional information to Depot about the considered web services, such as the new labels and the current context. This additional information helps Depot provide better results in future similar cases, such as “Establishing a company in France”.

5 Related Work

The problem of web service discovery is similar to looking for a needle in a haystack [GPST04]. Seeking the right service based on user's search criteria in which the user may be interested is still one of the main challenges in SOC [HLV07]. Several factors exacerbate this challenge [Bos08]:

- The increasing number of available web services on the web: With the increasing popularity of `Software-as-a-Service` and Cloud Computing, the number of public web services has been increasing steadily. For instance, Seekda (`http://webservices.seekda.com`) reported more than 28,000 web services;
- Keyword-based search: This factor is two-folded; from one side service providers typically release poor service descriptions that are not suitable for keywords matching. Additionally, today's complex business needs make it difficult for consumers to even choose proper keywords;
- Syntax-based search that ignores semantics of the terms used by service consumers (query) and service providers (service description).

The lack of rich service descriptions is highlighted by several researchers in the community [KTSW08]. Therefore, researchers have proposed several approaches to gather additional information about web services to handle the problem of poor service descriptions [BCPS05, DJZB05, HK03, MB05]. In [ANC10] we introduced an approach to extract and generate annotations for public web services from the content that their providers publish on their websites. We showed also that additional annotations can be generated for web services based on the analysis of the their invocations, such as tags [AN10].

Tremendous work has been proposed to meet the challenge of service discovery. This work can be grouped into four categories:

1. **Semantic approaches**: These approaches treat the problem of service discovery as a matchmaking problem [LPC06]. Service consumer's needs (goals) are captured in rich semantic expressions. These semantic expressions are then matched against the semantic descriptions of the considered web services.
2. **Information Retrieval approaches**: These approach investigate the semantic relationships between the term used by service consumers in their queries and those between the terms used to advertise the considered web services [MCZ07].
3. **Data mining approaches**: These approaches transform the service discovery task into a constraint satisfaction problem, where the submitted query is matched against

159

the collection of web services. Graph traversal and clustering methods are typically used to solve the constraint satisfaction problem, where the relationships between the considered web services are used to increase the accuracy of the returned result list [YSZX07].

4. **Linking approaches**: These approaches follow an integration approach to perform web service discovery. Typically, a set of web services that collectively satisfy a service consumer's need is presented to the consumer. Service composition is then used to consume these web services [RS04].

In this work, we target two of the main limitations in existing service discovery approaches, such as the ones mentioned above: (i) The unrealistic assumptions of the availability of complete and correct service descriptions [FKL$^+$05], and (ii) the ability of service consumers to express their sophisticated real business needs so that service discovery can be achieved [Sun05]. In the Adaptive Service Grid (ASG) project [6], the authors assume that today's complex business needs can be expressed (reasonably) by service consumers using ontologies to achieve a semantic service discovery. Our approach uses actual service usages in BPs to extract additional annotations to enrich their poor service descriptions that are used to provide enhanced service discovery, context-aware service selection, and smooth configuration of business process models.

6 Summary and Future Work

We tackle the problem of lack of rich service descriptions and its effects on the configuration of BPs. Several approaches have been proposed to enrich poor service descriptions with additional sources of information, such as community annotations, domain experts, etc. In this paper, we introduced a novel approach to enrich poor service descriptions with annotations extracted from the configurations of BPs that consume them. Two types of information can be extracted from such configurations, namely, annotations from task-web service assignment and context(s) of web services. the extracted annotations help enhance service discovery in future cases. Context information enables context-ware service selection, and smooth BP configuration.

We used Oryx and Depot to implement a prototype that realizes our approach. Oryx represents the front-end and Depot represents the back-end of our prototype. We showed how our approach can help create BPMs in several domains and introduced a detailed example to to model a process for establishing a company in Germany.

Our approach is collaborative, incremental, and practical. The more we can ship back from process models to service registries, the better service discovery we can provide. In future, we will consider not only task-service assignments but also global behavioral relationships imposed on services by process models via their tasks. We believe that this is of crucial importance to achieve service composition in a practical way.

[6] www.asg-platform.org

160

References

[AA11] Mohammed AbuJarour and Ahmed Awad. Discovering Linkage Patterns among Web Services using Business Process Knowledge. In *Proceeding of the International Conference on Services Computing*, Washington, DC, USA, 2011. IEEE Computer Society.

[AAMZM04] I. Budak Arpinar, Boanerges Aleman-Meza, Ruoyan Zhang, and Angela Maduko. Ontology-Driven Web Services Composition Platform. In *Proceedings of the IEEE International Conference on E-Commerce Technology*, pages 146–152, Washington, DC, USA, 2004. IEEE Computer Society.

[AN10] Mohammed AbuJarour and Felix Naumann. Dynamic Tags For Dynamic Data Web Services. In *Proceedings of WEWST'10*, Ayia Napa, Cyprus, 2010. ACM.

[ANC10] Mohammed AbuJarour, Felix Naumann, and Mircea Craculeac. Collecting, Annotating, and Classifying Public Web Services . In *OTM 2010 Conferences*, Crete, Greece, 2010. Springer.

[Awa07] Ahmed Awad. BPMN-Q: A Language to Query Business Processes. In *Proceedings of EMISA'07*, volume 119, pages 115–128, St. Goar, Germany, 2007.

[BCPS05] Marcello Bruno, Gerardo Canfora, Massimiliano Di Penta, and Rita Scognamiglio. An Approach to Support Web Service Classification and Annotation. In *Proceedings of EEE'05*, pages 138–143, Washington, DC, USA, 2005. IEEE Computer Society.

[Bos08] Aishwarya Bose. Effective Web Service Discovery using a Combination of a Semantic Model and a Data Mining Technique. Master's thesis, Queensland University of Technology, Queensland, Australia, 2008.

[BTR10] Stephan Buchwald, Julian Tiedeken, and Manfred Reichert. Anforderungen an ein Metamodell für SOA-Repositories. In Christian Gierds and Jan Sürmeli, editors, *ZEUS*, volume 563, pages 17–24. CEUR-WS.org, 2010.

[DJZB05] Zhang Duo, Li Juan-Zi, and Xu Bin. Web Service Annotation Using Ontology Mapping. In *Proceedings of SOSE '05*, pages 243–250, Washington, DC, USA, 2005. IEEE Computer Society.

[DOW08] Gero Decker, Hagen Overdick, and Mathias Weske. Oryx - An Open Modeling Platform for the BPM Community. In *BPM*, volume 5240 of *LNCS*, pages 382–385. Springer, 2008.

[Eur06] European Parliament and Council. Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market. ISSN 1725-2555, December 2006.

[FKL+05] Dieter Fensel, Uwe Keller, Holger Lausen, Axel Polleres, and Ioans Toma. WWW or what is wrong with web service discovery? In *Proceedings of the W3C Workshop on Frameworks for Semantics in Web Services*, 2005.

[GPST04] John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos, and Athanasios Tsakalidis. Web Service Discovery Mechanisms: Looking for a Needle in a Haystack? In *International Workshop on Web Engineering*, 2004.

[GWJV+09] Florian Gottschalk, Teun A. Wagemakers, Monique H. Jansen-Vullers, Wil M. Aalst, and Marcello Rosa. Configurable Process Models: Experiences from a Municipality Case Study. In *Proceedings of CAiSE'09*, pages 486–500, Berlin, Heidelberg, 2009. Springer-Verlag.

161

[HK03] Andreas Heß and Nicholas Kushmerick. Learning to Attach Semantic Metadata to Web Services. In *International Semantic Web Conference*, pages 258–273. Springer, 2003.

[HLV07] Stephan Hagemann, Carolin Letz, and Gottfried Vossen. Web Service Discovery - Reality Check 2.0. In *Proceedings of NWESP '07*, pages 113–118, Washington, DC, USA, 2007. IEEE Computer Society.

[Jos07] Nicolai M. Josuttis. *SOA in Practice: The Art of Distributed System Design*. O'Reilly, 2007.

[KLKR07] Ulrich Küster, Holger Lausen, and Birgitta König-Ries. Evaluation of Semantic Service Discovery- A Survey and Directions for Future Research. In *Postproceedings of WEWST'07*, 2007.

[KTSW08] Dominik Kuropka, Peter Tröger, Steffen Staab, and Matthias Weske. *Semantic Service Provisioning*. Springer, Germany, 2008.

[LPC06] Chuanchang Liu, Yong Peng, and Junliang Chen. Web Services Description Ontology-Based Service Discovery Model. In *Proceedings of WI '06*, pages 633–636, Washington, DC, USA, 2006. IEEE Computer Society.

[MB05] Brahim Medjahed and Athman Bouguettaya. A Dynamic Foundational Architecture for Semantic Web Services. *Distrib. Parallel Databases*, 17(2):179–206, 2005.

[MCZ07] Jiangang Ma, Jinli Cao, and Yanchun Zhang. A Probabilistic Semantic Approach for Discovering Web Services. In *Proceedings of WWW '07*, pages 1221–1222, New York, NY, USA, 2007. ACM.

[RS04] Jinghai Rao and Xiaomeng Su. A Survey of Automated Web Service Composition Methods. In *Proceedings of SWSWPC'04*, pages 43–54, 2004.

[SAP09] SAP Research. Unified Service Description Language. `http://www.internet-of-services.com`, 2009.

[SBE07] Sebastian Stein, Katja Barchewitz, and Marwane El Kharbili. Enabling Business Experts to Discover Web Services for Business Process Automation. In *Proceedings of WEWST*, Halle (Saale), Germany, 2007.

[SS04] Raghuram M. Sreenath and Munindar P. Singh. Agent-based Service Selection. *J. Web Sem.*, 1(3):261–279, 2004.

[Sun05] Sun Microsystems. Effective SOA Deployment using an SOA Registry Repository, A Practical Guide, 2005. White paper.

[SWGM05] Marta Sabou, Chris Wroe, Carole Goble, and Gilad Mishne. Learning Domain Ontologies for Web Service Descriptions: An Experiment in Bioinformatics. In *Proceedings of WWW'05*, 2005.

[YSZX07] Jianjun Yu, Hao Su, Gang Zhou, and Ke Xu. SNet: skip graph based semantic web services discovery. In *Proceedings of SAC '07*, pages 1393–1397, New York, NY, USA, 2007. ACM.

162