



HAL
open science

Business process implementation using an ontology-driven Web service selection algorithm

Sopheha Chhun, Chantal Cherifi, Néjib Moalla, Yacine Ouzrout

► **To cite this version:**

Sopheha Chhun, Chantal Cherifi, Néjib Moalla, Yacine Ouzrout. Business process implementation using an ontology-driven Web service selection algorithm. 5ème Journées Francophones sur les Ontologies (JFO 2014), Nov 2014, Hammamet, Tunisia. hal-01534554

HAL Id: hal-01534554

<https://hal.science/hal-01534554>

Submitted on 7 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Business process implementation using an ontology-driven Web service selection algorithm

Sophea CHHUN — Chantal CHERIFI — Néjib MOALLA —
Yacine OUZROUT

*DISP Laboratory
University of Lyon 2
Bron, France*

{Sophea.Chhun, Chantal.BonnerCherifi, Nejib.Moalla, Yacine.Ouzrout}@univ-lyon2.fr

ABSTRACT. The constraints on business process applications in terms of automation and reusability makes their development a challenging issue. To manage the variety of business process requirements, agile methods are required. Classically, business processes are implemented in a static way, what makes any evolution costly to realize. To cope with those needs, we propose the use of Web services to implement business processes. Web services and the designed business processes are stored in ontologies. An efficient multi-criteria service selection algorithm performs a semantic matching between the ontologies. It selects the most appropriate Web services among the existing ones for implementing a business process. This algorithm considers the business context, functional and QoS properties of the Web services. The user is asked for weights on QoS attributes. WordNet is integrated to solve the synonym problems. To be closer to reality and for more accurate results, our algorithm takes into account the changing values of QoS over time.

KEYWORDS: Business process, Multi-criteria Service selection, Ontology, QoS

1. Introduction

A business process is an activity (task) or set of activities that accomplish a specific organizational goal. Activities are linked together and are executed following a pre-defined order. Business tasks can be manual tasks, service tasks, script tasks or even sub processes. A business process may realize a business goal of a single organization or of a group of organizations. Legacy systems that implement enterprise business processes reach some limitations regarding nowadays ongoing business changes. Enterprises evolve in a highly dynamic environment with fast mutations. Depending on their size, their complexity, the number of partners involved, need change, these systems may rapidly become difficult to maintain, inefficient, and consequently costly. To stay competitive, enterprises need solutions designed to give the agility to cope with ongoing changes. Newer technologies and more efficient methods to perform business tasks are required. The modeling of business processes (BPM) is a step in the improvement of business performances. The modeling intends to represent enterprise processes with the aim of optimizing the efficiency of connecting activities in the provision of products or services. Different business process modeling languages exist including Business Process Modeling Notation (BPMN) [1], Petri Net [2], Workflow, Unified Modeling Language (UML), Petri Net Modeling Language, Business Process Modeling Language (BPML) and Business Process Execution Language (BPEL). Each language provides different notions, syntax, and complexity for modeling a business process. Web services have been an opportunity, caught by many organizations, to modernize and transform their legacy systems. This technology provides a mean to migrate and to externalize core business logic and competencies. A Web service, as defined by the W3C, is a software system designed to support interoperable machine-to-machine interaction over a network. Functions available throughout an organization are presented through standard-based services within a Service a Service Oriented Architecture (SOA). Any internal or external application can then consume them. In this work, we couple the expressiveness power of BPM with the agility of SOA in a unified and valuable framework called BPMNSemAuto. BPMN is in charge of expressing the required business processes. SOA concentrates on the available functionalities and on the way they are exposed to different applications susceptible to invoke them. The management of business processes tends to be even more valuable if it can be automated, and if the processes are fueled by the most accurate and up-to-date available services. The automatization allows the user defining a process flow and generating the complete working system. Unfortunately, current SOA technologies only allows manual keyword Web services search. To cope with this challenge, we introduce semantics within our framework. Business processes as well as Web services are described by ontologies [3-7]. An automatic reasoning for ontology matching is then possible to find appropriate Web services. This selection step is a crucial activity within SOA. Candidate services with similar functionalities are numerous. They are developed by different providers using their own vocabularies to name the services, their operations and parameters. The challenge is to select the most satisfactory services who will implement the business tasks. Quality of Service (QoS) is also a criteria in the selection process. It is an

important non-functional aspect used to differentiate functionally similar Web services. QoS embodies response time, execution time, availability, reliability, security, number of call... Some QoS properties can change over time. For example, from one call to another, execution time may increase or decrease for various reasons. This brings a dynamic aspect important to be considered. Note that for the selection process, users can be asked for their requirements on QoS. Web service selection is a research issue that has triggered a large body of work since the early 2000s. Selection algorithms take into account one or a combination among the business context (keywords expressing the category of a Web service), the functional and non-functional (QoS) properties [8-17]. But to our knowledge, none of the existing works consider the changing value of QoS over time. Our work is in this line. We propose a multi-criteria service selection algorithm relying on three criteria. The first one is the context; the context specified by the user is compared with keywords extracted from the category's name, the category's description and the name of a Web service. The second one is the functional properties; required parameters of a business task are compared with parameters of available Web services in terms of number, name and data type. The use of Wordnet¹ solves synonym problems. The third one is QoS attributes; we consider availability, execution time and number of calls. Note that we introduce the QoS properties in the ontology that we designed to describe the Web services.

The rest of this paper is organized as follows. Section 2, "Background", gives fundamental information about Business Process Modeling and Web services Description. In Section 3, "Framework Overview", an overview of BPMNSEmAuto is presented. Section 4 "Semantic Representation of User' Requirements" is dedicated to the modeling of the user needs. In Section 5, "Semantic Web Service Selection", a detailed and thorough presentation of the selection algorithm is provided. The paper is ended by a conclusion and an introduction to the future works.

2. Background

2.1. Business Process Modeling

Business process modeling is based on two dominant formalisms, graph-based and rule-based [18]. The rule-based is grounded by the formal logic. The graphical approach provides a graphical interface enabling users to model their business processes in an intuitive way. This is undoubtedly an asset which makes it far more popular than the rule-based approach. In [18], the authors conduct a comparison study based on different criteria between several graph-based and rule-based business process modeling. Among the graphical modeling, BPMN seems to be the one that provides the greatest ease of use for business users. BPMN was presented as a standard in 2004 by the Object Management Group. It bridges the gap between the design and implementation of business processes. The primary goal of BPMN is

¹ <http://wordnet.princeton.edu>

to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. The BPMN specifications divide the business process elements into four categories: (i) Flow objects that define the behaviors of a business process. A flow object can be an event, an activity or a gateway; (ii) Connecting objects that connect two flow objects or a flow object with other resources. There are three types of connecting objects: sequence flow, message flow and association; (iii) Swimlanes that group the primary modeling elements. There are two kinds of swimlanes, pool and lane; (iv) Artifacts allow to provide additional information about the process. They are categorized into three sub-groups: processed data, group of activities and annotations. Two or many business tasks are linked to each other by gateways (parallel, inclusive, exclusive, complex and event based gateway). The version 2 of BPMN was released in 2011. It has around 100 different modeling constructs, including 51 event types, 8 gateway types, 7 data types, 4 types of activities, 6 activity markers, 7 task types, 4 flow types, pools, lanes, etc [19].

2.2. Web Services Description

In order to be discovered and used, Web services are published by providers to UDDI registries. The registries host three types of information. The first one describes the enterprises and the services that they supply. The second one describes logical sets of Web services and their categories; it provides a textual description and the objectives of the services. The third one is the technical information necessary to invoke a Web service. The registry maintains pointers to the Web service description. Currently, the Web services description is based on the Web Service Description Language (WSDL) standard. WSDL specifies functional and non-functional properties. The functional properties describe the operations that a service exposes with their input and output parameters. Parameters are described by their name and datatype. Non-functional properties concern the location of the service, the communication protocol and the data format specifications. Others solutions have been proposed that enrich the descriptions with semantics. Indeed, WSDL providing only syntactical information, it lacks the semantic expressivity needed to represent the Web Services capabilities. This situation can lead to possible misinterpretation. Semantics is introduced by using ontologies that support shared vocabularies and allow automatic reasoning. The Semantic Web service field includes substantial bodies of work through three conceptual approaches, Ontology Web Language for Services (OWL-S) [20], the Web Services Modeling Ontology (WSMO) [21], Semantic Annotation for WSDL (SAWSDL) and WSDL-Semantic (WSDL-S). OWL-S and WSMO adopt a top-down perspective. The semantics is described independently of the service development. It is grounded with the corresponding WSDL description. SAWSDL and WSDL-S adopt a bottom-up approach where the WSDL file is enriched with semantic information. Some proposals can be find in the literature to add QoS attributes to the descriptions.

3. Framework overview

BPMNSemAuto is composed of four modules. A complete description of BPMNSemAuto is available in [22]. The four modules are represented in Fig. 1 with their internal interactions as well as the interactions with users.

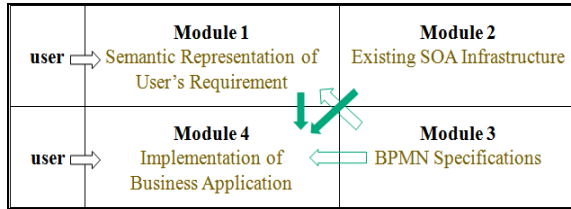


Fig. 1. The four modules of the BPMNSemAuto framework

The “Existing SOA infrastructure” module 2, Fig. 2, represents the pool of the available Web services published by service providers in the registry UDDI (*Universal Description, Discovery and Integration*). The Web services are grabbed from registries in order to be stored in an ontology called WSOnto. In [22], we described the details about the data completion and the structure of WSOnto. The specificity of WSOnto is its ability in storing the QoS features of Web services, along with their functional properties and their description.

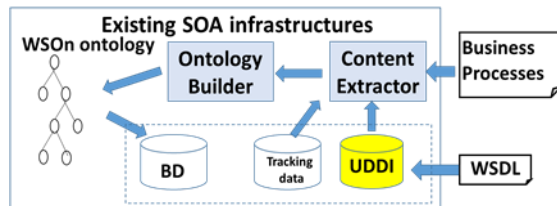


Fig. 2. Module 2: Existing SOA infrastructure.

The “Semantic Representation of Users Requirements” module (Module 1) captures the user needs and transform them into an ontology of business process called BPOnto. This part is detailed in the following section.

In the fourth module (Fig.3), “Implementation of Business Application”, the Semantic Matching Engine performs a matching between BPOnto and WSOnto ontologies. The service selection is the responsibility of the Service Selection". It finds the most appropriate Web services corresponding to the business tasks. The service composition is done by the Service Compositor. The goal is to create a composite service when no atomic service can satisfy the requirements of the user. This module finally generates the executable business process with the Business Process Transformer. The executable application is accessible through the User Interface. In the following, we focus on the Service Selector.

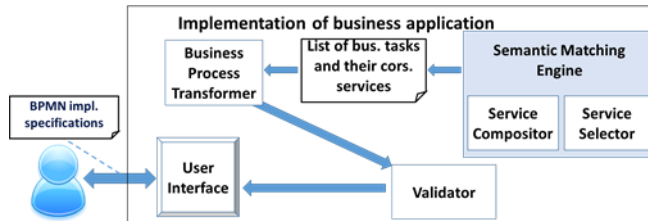


Fig. 3. Module 4: Implementation of business application.

The third module “BPMN Specifications” stores the specifications of BPMN 2.0. They are used to validate the designed business process of the user.

4. Business process requirements

The ultimate goal of this module is to represent the user needs under the form of an ontology. This ontology will be matched with the WSOnto ontology in order to find appropriate Web services able to implement the desired business tasks. The architecture of the module and the steps towards the semantic representation of the users’ requirements are illustrated by Fig. 4. The module is composed of two parts, a User interface and a Semantic Transformer.

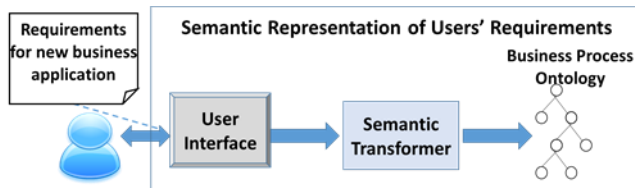


Fig. 4. Module 1: Semantic representation of users requirements

4.1. Design of a business process

A business process is first designed by using any support business process modeling tools such as Activiti² or JDeveloper³. For each business task, users must provide a textual description in terms of inputs, outputs, and context as keywords. The input and output parameters are represented by the couple (variable-name, variable-value). The context is defined by a set of keywords that describe the business task. Fig. 5 illustrates a business process of sending an email, and the description of each task. The first task is the authentication task with two input parameters (username and password), one output parameter (authentication) and a context defined by the keywords, “authentication” and “login”. The second task is the sending of the email; it contains three input parameters (address of the sender,

² <http://activiti.org>

³ <http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html>

address of the receiver and the content of the email), and one output parameter (reply).

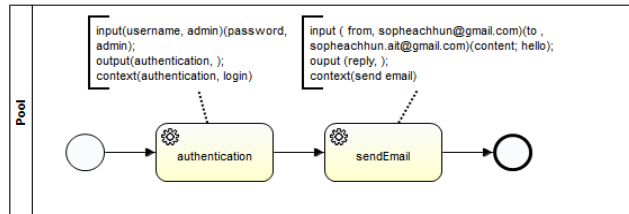


Fig. 5. A business process contains two tasks with attached information

The output of this step is an XML file describing the designed business process and its business tasks.

4.2. User interface

The User Interface allows the user to input the XML file generated by the previous step. At this point, the user is invited to specify the weights of the QoS attributes (availability, execution-time, total number of calls for each service).

4.3. Semantic transformer

The Semantic Transformer generates the BPOnto ontology that represents the designed business process of user. It takes as input the XML file that has been specified during the previous step. This task is performed by using an existing XSLT transformation which generates the BPOnto ontology in turtle format, based on the BPMN 2.0 ontology proposed in [23]. BPMN 2.0 ontology is an ontology that defines the specifications of BPMN 2.0.

4.4. Business process validation

In order to be sure that the generated business processes are executable, a validation process is performed on BPOnto. An ontology reasoner is used to detect whether the designed business process diagram is constructed with respect to the specifications of BPMN 2.0. In the current version of the framework, Pellet [24], FaCT++ [25] or Hermit reasoners can be used.

5. Semantic Matching for service selection

The semantic matching between BPOnto and WSOnto, Fig. 6, takes place after the business process validation. During the matching, the service selection algorithm looks for correspondences between each business tasks and each stored Web services. It provides a list of the most appropriate Web services.

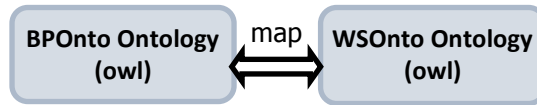


Fig. 6. Matching between BPOnto and WSOnto.

The pseudo code of the service selection algorithm is presented in Fig. 7.

```

1 Traverse BPOnto & WSOnto
2 for each BT (business task) of business process
3   // perform context matching to find web services (WS)
4   // that match with the user's context
5   for each category of WS
6     if WSOnto.category.keyword match BPOnto.BT.context != empty
7       if WSOnto.WS.keyword match BPOnto.BT.context != empty
8         // perform the functional matching
9         for each WS that matches with user's context
10          score1 = functional_matching(WS);
11          if score1 is in Range defined in function (1)
12            // perform the non functional calculating
13            score2 = non_functional_calculating(WS);
14            total_score = score1 + score 2;
15          end for
16        end for
17      Rank the web service based on the total_score;
18      choose the web service with max total_score;
19    end for

```

Fig. 7. Pseudocode of the service selection algorithm.

5.1. Context matching

The context of the user is matched with the keywords attached to each Web service and service's category in WSOnto. It is a similarity measure between two strings. Wordnet is integrated in the process to detect the synonyms. If at least one match is found, the process is considered to be successful.

5.2. Functional matching

The functional matching is the process of comparing respectively the inputs and the outputs of the users with the inputs and the outputs of the Web service's operations. Three types of comparison are performed. The first one is the number of inputs and outputs. The second one is a string comparison between the name of the inputs and between the name of the outputs. The third one is a string comparison between the data types. Fig. 8 illustrates the score calculation of one input or output. The notation "Nb" refers to the number of inputs or outputs. "Equal" means that the user's request and the considered Web service's operation have the same number of inputs (respectively outputs). "Less" means that the user's request has less inputs (respectively outputs) than the considered Web service's operation. "More" means that the user's request has more inputs (respectively outputs) than the considered Web service's operation. The notation "String" means that we perform a string matching. For the matching of the number of parameters, in both cases (input and output), "Equal" has the highest score. It is considered as the perfect match. We consider the "More" case as more satisfying than the "Less" case. For the name and

data type string matching, two situations are considered, the equality and the non-equality.

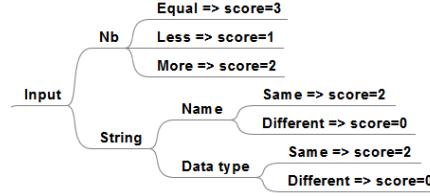


Fig. 8. Scoring method in the functional matching.

For each input or output, there is a minimum score, an acceptable score and a maximum score. The process of calculating the score for inputs and outputs is the same. The minimum score for one input is equal to 1; this happens when the name and data type of the input do not matched and the number of inputs of the user is less than the number of inputs of the service's operation. The acceptable score acceptable score range is defined in (1).

$$\text{Range} = [(1+4*(\text{Nb_inputs}+\text{Nb_outputs})) (3+4(\text{Nb_inputs}+\text{Nb_outputs}))] \quad (1)$$

5.3. QoS score

The score for the QoS attributes (SCORENFS) is obtained from the score of the aggregate percentage change (SCOREAPC) plus the score of the utility function (SCOREUF) at the time $t=0$; equation (6).

The SCORE_{NFS} is calculated in five steps, as followed.

Step1: Calculate the utility function by using the equation (2). This utility function is proposed in [25] for calculating the QoS value.

$$UF = \sum_{i=1}^{\alpha} w_i * \left(\frac{q_{ai}(R) - \mu_{ai}}{\sigma_{ai}} \right) + \sum_{j=1}^{\beta} w_j * \left(1 - \frac{q_{bj}(R) - \mu_{bj}}{\sigma_{bj}} \right) \quad (2)$$

Where:

- α : the number of QoS attributes that are required to maximize their value;
- β : the number of QoS attributes that are required to minimize their value;
- w : the weight of QoS attributes ($0 < w_i, w_j < 1$) and $\sum_{i=1}^{\alpha} w_i + \sum_{j=1}^{\beta} w_j = 1$;
- μ : the average value;
- σ : the standard deviation of QoS attributes for all service candidates;
- $\alpha+\beta$: the total number of QoS attributes;
- q : QoS value.

Step 2: Calculate the percentage change (PC) values of the QoS values of a Web service that changes over n time gaps (t_0 to t_{n-1}). The formula to calculate the percentage change value is defined in equations (3). The PC allows to consider the variation of a QoS value of a Web service over times. The PC value describes how much a value is changed compare to its previous value. The time here refers to the

time when the process for calculating the QoS value was run. The granularity of the time gap can be the week or the month depending on the companies' choice. The Web services can be created in different times. Therefore, to compare between two Web services, only the last n numbers of time gaps are considered when calculating the percentage change value.

$$PC = \frac{\text{new_value}}{\text{old_value}} - 1 \quad (3)$$

Step 3: Calculate the aggregate percentage change (APC) value of each Web service over the last n numbers of time gaps; equation (4).

$$APC = PC_{t0,1} + PC_{t1,2} + \dots + PC_{tn-2,n-1} \quad (4)$$

Step 4: Calculate the QoS score ($SCORE_{NFS}$), using the equation (5). For all the Web services that validate the context matching and whose functional score is in the range defined in (1), the non-functional score is calculated (Fig. 9). The calculation is done at the service operation level.

The $SCORE_{APC}$ is equal to the value of APC and the $SCORE_{UF}$ is calculated as defined in the Fig. 9.

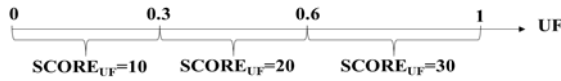


Fig. 9. Scoring the utility function value.

$$SCORE_{NFS} = SCORE_{APC} + SCORE_{UF} \quad (5)$$

6. Conclusion & Future Work

In this paper, we present a proposition for realizing business processes by leveraging the Web service paradigm. Our proposition makes use of ontologies to represent the business processes as well as the set of available Web services. Those choices allow to cope with two major challenges faced by companies: the agility of the applications to manage ongoing changes and the automation of the processes. Business processes are designed using the well-known standard modeling notation, BPMN. A transformation of the designed business processes results in an ontology expressing the user needs. The correctness of the business process diagram is checked with the help of an ontology reasoner. Our framework implements a service selection algorithm based on a semantic matching between ontologies. This algorithm is based on three criteria: the context, functional and non-functional properties. It performs the matching between the context expressed by the user and the category of a Web service, the matching between the number of inputs and between the number of outputs, the matching between the parameters name, the matching between the data types. It calculates the value of three QoS attributes for

each Web service. The originality of the calculation of the QoS attributes values relies on the fact that we take into account the variation of the values over a number of time gaps. The context is determinant to pursue or not the selection process. If the matching goes on, a score is provided to the Web services by the functional matching and by the QoS calculating. According to the scores and to the weights on the QoS attributes provided by the user, candidate Web services are ranked. The richness of BPMN and the power of the ontologies coupled with a sophisticated selection algorithm, is a promising proposition to reach dynamic and automatic business processes implementation. Nevertheless, some improvements remain to be done on the framework. The business process checking validates only the diagram. It does not check the consistency of the inputs and outputs of the whole business process. A future work will be to propose a method to perform the consistency checking. Currently, in WSOnto, the Web services are organized into service categories that are linked to a set of keywords. With an increasing number of stored Web services, some keywords of different categories will overlap at one point. This contrasts with our original purpose of using keywords in context matching to filter the set of appropriate Web services, and to reduce the execution time in the service selection process. To solve this issue, we plan to propose a Web service domain and link Web services to domains.

Acknowledgment

This project has been funded with support from the European Commission (EMA2-2010- 2359 Project). This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. Bibliography

- [1] B. P. Model, "Business Process Model and Notation (BPMN) Version 2.0," *Object Management Group specification*, 2011.
- [2] W. Reisig and G. Rozenberg (Eds.), *Lectures on Petri Nets I: Basic Models. Advances in Petri Nets*, vol. 1491. Springer Verlag, Berlin, 1998.
- [3] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them?," *Intelligent Systems and Their Applications, IEEE*, vol. 14, no. 1, pp. 20–26, 1999.
- [4] D. Kalibatiene and O. Vasilecas, "Survey on Ontology Languages," *Perspectives in Business Informatics Research*, pp. 124–141, 2011.
- [5] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data & Knowledge Engineering*, vol. 46, no. 1, pp. 41–64, Jul. 2003.
- [6] S. Youn, A. Arora, P. Chandrasekhar, P. Jayanty, A. Mestry, and S. Sethi, "Survey about Ontology Development Tools for Ontology-based Knowledge Management," *University of Southern California*, 2009.
- [7] M. Sabou, "Building Web Service Ontologies," SIKS Dissertation Series, 2006.

- [8] A. Pradnya Khutade and B. Rashmi Phalnikar, "QOS BASED WEB SERVICE DISCOVERY USING OO CONCEPTS," *International Journal of Advanced Technology & Engineering Research (IJATER)*, vol. 2, no. 6, pp. 81–86, 2012.
- [9] S. Chaari, Y. Badr, F. Biennier, C. BenAmar, and J. Favrel, "Framework for web service selection based on non-functional properties," *International Journal of Web Services Practices*, vol. 3, no. 2, pp. 94–109, 2008.
- [10] T. Broens, S. Pokraev, M. Van Sinderen, J. Koolwaaij, and P. Dockhorn Costa, "Context-aware, ontology-based service discovery," *Ambient Intelligence*, vol. Ed. Springer, pp. 72–83, 2004.
- [11] S. Neelavathi and K. Vivekanandan, "An Innovative Quality of Service (QoS) based Service Selection for Service Orchestration in SOA," *International Journal of Scientific and Engineering Research*, vol. 2, no. 4, pp. 1–8, 2011.
- [12] T. Yu and K. J. Lin, "Service selection algorithms for composing complex services with multiple QoS constraints," *Service-Oriented Computing-ICSOC 2005*, pp. 130–143, 2005.
- [13] O. Cabrera, M. Oriol, X. Franch, L. López, J. Marco, O. Fragoso, and R. Santaolaya, "WeSSQoS: A Configurable SOA System for Quality-aware Web Service Selection," *arXiv preprint arXiv:1110.5574*, 2011.
- [14] M. Alrifai, T. Risse, P. Dolog, and W. Nejdl, "A scalable approach for qos-based web service selection," in *Service-Oriented Computing-ICSOC 2008 Workshops*, 2009, pp. 190–199.
- [15] T. Yu, Y. Zhang, and et al., "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, p. 6, 2007.
- [16] Y. Tian and M. Huang, "Enhance discovery and retrieval of geospatial data using SOA and Semantic Web technologies," presented at the Expert Systems with Applications, 2012.
- [17] D. A. D'Mello and V. S. Ananthanarayana, "Semantic Web Service Selection Based on Service Provider's Business Offerings," *IJSSST*, vol. 10, no. 2, pp. 25–37, Mar. 2009.
- [18] R. Lu and S. Sadiq, "A survey of comparative business process modeling approaches," in *Business Information Systems*, 2007, pp. 82–94.
- [19] A. Correia and F. B. e Abreu, "Adding Preciseness to BPMN Models," *Procedia Technology*, vol. 5, pp. 407–417, Jan. 2012.
- [20] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 web ontology language primer," *W3C recommendatton*, vol. 27, pp. 1–123, 2012.
- [21] H. H. Wang, N. Gibbins, T. R. Payne, and D. Redavid, "A formal model of the Semantic Web Service Ontology (WSMO)," *Information Systems*, vol. 37, no. 1, pp. 33–60, Mar. 2012.
- [22] S. Chhun, N. Moalla, and Y. Ouzrout, "QoS ontology for service selection and reuse," *Journal of Intelligent Manufacturing*, pp. 1–13, 2014.
- [23] C. Natschläger, "Towards a BPMN 2.0 Ontology," in *Business Process Model and Notation*, 2011, pp. 1–15.
- [24] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.
- [25] D. Tsarkov and I. Horrocks, "FaCT++ description logic reasoner: System description," in *Automated reasoning*, Springer, 2006, pp. 292–297.