Knowledge-Based Systems 60 (2014) 1–9

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Log based business process engineering using fuzzy web service discovery

Omair Shafiq [a], Reda Alhajj [a,b,*], Jon Rokne [a]

[a] Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
[b] Department of Computer Science, Global University, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 23 February 2013
Received in revised form 25 October 2013
Accepted 29 November 2013
Available online 30 December 2013

Keywords:
Business process
Engineering
Mining
Discovery
Semantics

ABSTRACT

Business process engineering and mining is a technique that allows discovery, analysis and modeling of possible Business Processes based on information gathered from enterprise information systems. Most of currently available business process engineering and mining techniques either focus on machine learning techniques to mine, discover and model any possible Business Processes from raw data, or use semantically-enabled process models and service descriptions to construct and represent complex Business Processes. However, in real-life scenario, all the required services are not always available and hence exact matching of the services in order to construct Business Process is not possible. In this paper, we present our approach of using fuzzy Web Service discovery to construct and represent Business Processes. It helps in relaxing the matching criteria of Web Services, and allows service consumers to specify business requirements in a more fuzzy way, and hence increases the possibility of finding required Web Services that could construct Business Processes. The paper presents the proposed solution then reports and discusses the evaluation.

1. Introduction

Process engineering and mining [1] is about discovery and modeling of possible Business Processes from event logs. Event logs [2] are created in the processing of logging during an application executes. It allows applications to produce an execution log which is then used to monitor the execution and to debug as well as track any log events during the application execution. Event logs are the information about a system performing and executing activities and exchanging messages. It aims at analyzing operational processes that are repeatable. For example, travel booking (including flight booking, hotel booking and car rental), customer orders, opening an account, manufacturing in a factory, etc. Such kinds of processes are mostly invisible, as most information systems provide views for particular steps or stages of a Business Process.

Most information systems keep information about different steps and stages of the Business Processes as event logs. This includes different kinds of enterprise resource planning systems, customer retention management systems, enterprise middleware solutions (like IBM WebSphere middleware) and other information systems of organizations (like universities, companies, hospitals and other related organizations). Well-known standards, like business process execution language (BPEL) [3], specify actions within Business Processes with Web Services. Information from Web Service interfaces is imported and it is used to define processes in BPEL. Business Process engineering requires modeling Business Processes as Web Service interactions [4]. In most cases, a limited number of Web Services are available out of which relevant Web Services are sought in order to construct the required Business Processes. It requires an effective and dynamic matchmaking of Web Services based on given requirements in order to fulfill the required interactions within Business Processes. Lack of semantics in Web Services has been found as a major obstacle in Business Process formation, management and engineering [29]. It is essential for applications to have automation of processes and operations so that execution of repetitive and similar transactions could be made automated. Semantic Web Services (SWS) [9] try to bridge the gap by trying to automate the process of discovery, selection/ranking, composition, mediation and invocation of Web Services and provide means for the representation of executable artifacts that are accessible to intelligent user based queries as well as reasoning mechanisms. We propose to take into account discovery, selection/ranking, composition, mediation and invocation of Web Services to enable efficient Business Process Engineering and Management. If the process of Web Services execution could be made dynamic, it will be easier for Business Process Engineering and Management procedure to dynamically find Web Services for business transactions that are executed repeatedly, or the business

* Corresponding author at: Department of Computer Science, University of Calgary, Calgary, Alberta, Canada. Tel.: +1 403 210 9453.

E-mail addresses: moshafiq@ucalgary.ca (O. Shafiq), alhajj@ucalgary.ca (R. Alhajj), rokne@ucalgary.ca (J. Rokne).

0950-7051/$ - see front matter
http://dx.doi.org/10.1016/j.knosys.2013.11.021

transactions that are similar to previously executed business transactions.

Automated discovery, selection/ranking, composition, mediation and invocation of Web Services have received considerable attention in the last decade. Several approaches for service discovery have been proposed, implemented and evaluated; these include usage of semantics and data mining based heuristic techniques for the matchmaking of service consumers with service providers [5]. A significant number of Web Services have been offered by various service providers over the Web. Semantic Web based approaches have been applied to semantically annotated Web Services [6] to allow automated discovery and ranking, followed by mediation and invocation [7]. Rich semantic descriptions allow service providers to model their services in a more expressive way that makes it easier for service consumers to search for the required service using semantic reasoning and querying approaches [8]. This has turned Universal Description Discovery and Integration (UDDI) [28] and other traditional registry based solutions obsolete. Web ontology language for Web Services (OWL-S) allows specifying a set of ontologies based on OWL in order to describe different aspects of Semantic Web Services [9]. Three aspects are considered: service profile, service model and grounding which describe ''what a service does'', ''how a service works'' and ''how to access it'', respectively. The Web Service modeling framework (WSMF) [10] is another approach that allows modeling semantic Web Services. It includes a meta-ontology WSMO [7], description language – Web Service Modeling Language (WSML) [11], and an execution environment WSMX [6]. The Semantic Web Services framework (SWSF) [12] is a specification produced by the SWSL committee of the Semantic Web Service Initiative (SWSI) [12]; it has its own conceptual model and modeling language called Semantic Web Service Ontology (SWSO) and Semantic Web Service Language (SWSL), respectively. Web Service Description Language-Semantics (WSDL-S) is another approach that augments WSDL with semantics [13].

Business Process engineering and mining happens to be service driven and requires dynamic matchmaking of Web Services based on the given requirements for a Business Process in order to achieve a business goal [30]. Most of the discovery approaches for Web Services are based on strict and precise matchmaking algorithms and hence require the services to match exactly with the required goal. This is impossible because of the limited information available for describing Web Services. The descriptions for Web Services are limited and are based only on syntactic information and have several limitations on expressing the information about Web Services precisely. Existing approaches for Business Process Engineering are dependent on the discovery and usage of services which are based on static and pre-known information of services. Therefore, it causes limitations for user-applications to use newly available services, or current services with further updates.

In this paper, we propose to use partial semantics (non-functional properties) of Web Services and map the fuzzy matching theory [14] on the Web Service discovery problem in order to allow for flexibility in performing the matchmaking of Web Services for Business Processes. Fuzzy theory based matchmaking is a technique that allows matches that may be less than 100% perfect. When an exact match cannot be found from the available Web Services, the fuzzy matchmaking based approach allows to search for a match that is less than exact, based on some threshold of the fuzzy match to a percentage value less than 100%, and the discovery will have results that will match corresponding to that percentage. The primary objective of having the fuzzy theory based matchmaking is to relax the search criteria in case Web Services with required match are not available. We propose that the system should then search for Web Services that match approximately with the requirements from service consumers that are also prescribed in fuzzy terms. This approximation will give flexibility to perform matchmaking of Web Services based on the needs of Business Processes.

Fig. 1 depicts the overall scenario of Business Process management using Web Services. Our solution serves as a middleware between service consumers and service providers. Business Processes are constructed based on the requirements from service consumers. Business Processes have a set of activities that are to be fulfilled. Service consumers set the criteria as non-functional properties based on what matching is to be performed on the given Web Services. In order to allow for fuzzy matchmaking, a set of performance ratings of Web Services with respect to criterion as non-sfunctional properties are described by service consumers. The threshold is specified using the triangular fuzzy membership function (TFM) [15]. The fuzzification then allows for the mapping of crisp/concrete criteria to fuzzy criteria.

The rest of this paper is structured as follows. Section 2 describes related work and discusses the comparative analysis of the existing related approaches. Section 3 outlines the proposed solution of fuzzy theory based Web Service discovery that takes the problem of matchmaking of Web Services and maps it to the theory of fuzzy matching. Section 4 describes the experiments and discusses the evaluation and results. Section 5 is conclusions.

2. Related work

A significant amount of work has been done related to Business Process engineering using Web Service discovery, selection or composition using different matchmaking methods. Such methods range from syntactic matchmaking, data mining based approaches, to heavy weight semantic matchmaking approaches. Below we describe and analyze the related work.

Ranking of Semantic Web Services [19] uses the ontological representations of non-functional properties, and considers the non-functional properties as multi-criteria mechanism that takes the multiple non-functional properties as different possible dimensions for ranking. The proposed algorithm takes into account associated importance for non-functional properties from perspective users. Each of the discovered services is checked to discover if the requested non-functional properties specified in the goal are available in the service description or not. If a match is found, logic rules are extracted and evaluated using a reasoning technique supported by WSML. The approach is certainly valuable and enhances the discovery process, but requires users to model and pre-specify the importance of non-functional properties.

Another approach uses non-functional properties as well as users input for ranking Web Services [16]. It considers all the available non-functional properties of the available Web Services, but gives importance to the most preferred properties, while giving equal importance to the remaining properties, using the weight sum approach.

Web Service relevancy function (WsRF) [17] are used for measuring the relevancy and ranking of a particular Web Service based on the preferences of users and the corresponding Quality of Service (QoS) metrics. QoS parameters are response time, throughput, availability, accessibility, interoperability analysis, as well as cost to invoke the service. QoS parameters can be specified by clients manually using a GUI, and are taking into account computing the relevance of known Web Services or discovering Web Services over the Web [18].

In [20], the authors present a moderated fuzzy Web Service discovery approach to model subjective and fuzzy opinions, and to assist service consumers and providers in reaching a consensus. The method achieves a common consensus on the distinct opinions and expectations of service consumers and providers. This process is iterative such that further fuzzy opinions and preferences can be added to improve the precision of Web Service discovery.