International Journal of Technology in Computer Science and Engineering
ISSN: 2349-1582 Volume 04, No 2, June 2017

A Literature Review on Automatic Composition of Web Services

[1]R. Satheesh Kumar, [2]S. R. Boselin Prabhu, [3]G. Keerthana

[1]Associate Professor, Department of Computer Science and Engineering, Sahrdaya College of Engineering and Technology, Kerala, India

[2]Associate Professor, Department of Electronics and Communication Engineering, VSB College of Engineering Technical Campus, Tamilnadu, India

[3]Fulltime Research Scholar, Anna University Chennai, Tamilnadu, India

Abstract

In today's Web, Web services are created and updated on the fly. It's already beyond the human ability to analysis them and generates the composition plan manually. Automated composition we mean generating an executable process that satisfies a given composition requirement by communicating with a set of existing Web services. A powerful mean for these purposes is represented by sharing, reusing and composing value-added services made available on the Web, i.e. Web services Semantic Web services head in this direction, aiming to realize automatic discovery, selection and composition of existing services. Semantic Web Service composition can be used when no individual available service can satisfy a specific client request, but (parts of) available services can be composed and orchestrated in order to do it. A number of approaches have been proposed to tackle that problem. Different planning techniques have been proposed so far which address the problem of automated composition of web services. This paper gives an overview of recent research efforts of Automatic Web service composition.

Keywords

Automatic service composition, process discovery, distributed system.

1. Introduction

Complex mission plans may need to incorporate information from various sources and domains to achieve a task. This information is available through a variety of web services in the Service-Oriented Architecture (SOA), but the ability to automatically compose them into a single coherent task is not readily available. Web services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web.

Nowadays, an increasing amount of companies and organizations only implement their core business and outsource other application services over Internet. Thus, the ability to efficiently and effectively select and integrate inter-organizational and heterogeneous services on the Web at runtime is an important step towards the development of the Web service applications. In particular, if no single Web service can satisfy the functionality required by the user, there should be a possibility to combine existing services together in order to fulfill the request [1].

However, three major challenges are involved in web services dynamic composition: first one ensure whether there are a set of available web services that if executed in a particular setting and provide the requested service, second, how to model these existing web services and third what effective algorithm can produce new web services by utilizing the existing services models. Research in planning is more and more focusing on the problem of the automated composition of web services. given a set of services that are published on the Web, and given a goal, generate a composition of the available services that satisfies the goal. Several methods for Web service composition have been proposed in recent years.

In this paper we will present an overview for automatic composition of service. The automation means that either the method can generate the process model automatically, or the method can locate the correct services if an abstract process model is given. This paper is organized as follows. Section 2 presents an abstract architecture for Web service composition. Section 3 is the phases in Web service composition.

2. Web Service Composition Architecture

Here is a general framework for automatic Web services composition. This framework is in high-level abstraction, without considering a particular language, platform or algorithm used in composition process. The aim of the framework is to give the basis to discuss similarities and differences of the available service composition methods.

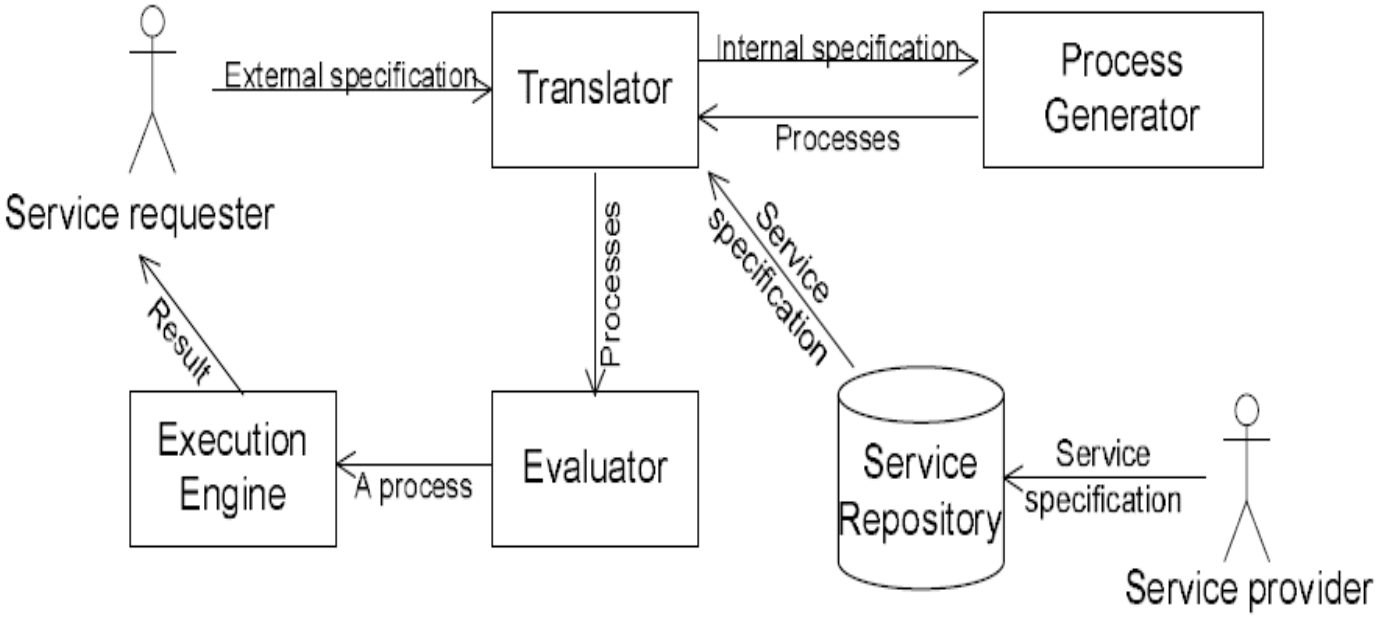

Fig.1 Web Service Composition Architecture

A general framework of the service composition system is illustrated in Fig. 1. The composition system has two kinds of participants, service provider and service requester. The service providers propose Web services for use. The service requesters consume information or services offered by service providers. The system also contains the following components: translator, process generator, evaluator, execution engine and service repository. The translator translates between the external languages used by the participants and the internal languages used by the process generator. For each request, the process generator tries to generate a plan that composes the available services in the service repository to fulfill the request. If more than one plan is found, the evaluator evaluates all plans and proposes the best one for execution. The execution engine executes the plan and returns the result to the service provider.

3. Phases in Web Service Composition

3.1 Presentation of single service:

In this the service providers will advertise their atomic services at a global market place. There are several languages available for advertising, for example, UDDI [4] or DAML-S ServiceProfile [4]. The essential attributes to describe a Web service include the signature, states

and the non-functional values. The signature is represented by the service's inputs, outputs and exceptions. It provides information about the data transformation during the execution of a Web service. The states are specified by precondition and post condition. We model it as the transformation from one set of states to another in the world. Non-functionality values are those attributes that are used for evaluating the services, such as the cost, service quality and security issues.

3.2 Translation of the languages:

Most service composition systems distinguish between the external and internal service specification languages. The external languages are used by the service users to enhance accessibility of the users in the sense that the users can express what they can offer or what they want in a relatively easy manner. They are usually different from the internal ones that are used by the composition process generator, because the process generator requires more formal and precise languages, for example, the logical programming languages. So far, the users have already get used to the standard Web service languages, such as WSDL and DAML-S. Thus the translation components between the standard Web service languages and the internal languages have to be developed. In the meantime, the service requester can also express the requirement in a service specification language.

3.3 Composition process model:

A process generator then tries to solve the requirement by composing the atomic services advertised by the service providers. The process generator usually takes the functionalities of services as input, and outputs process model that describes the composite service. The process model contains a set of selected atomic services and the control flow and data flow among these services.

3.4 Evaluation of composite service:

It is quite common that many services have the same or similar functionalities. So it is possible that the planer generates more than one composite service fulfilling the requirement. In that case, the composite services are evaluated by their overall utilities using the information provided from the non-functional attributes. The most commonly used method is utility functions. The

requester should specify weights to each non-functionality attributes and the best composite service is the one who is ranked on top.

3.5 Execution of composite service:

After a unique composite process is selected, the composite service is ready to be executed. Execution of a composite Web service can be thought as a sequence of message passing according to the process model. The dataflow of the composite service is defined as the actions that the output data of a former executed service transfers to the input of a later executed atomic service.

Web Service Composition

This section puts the work in this paper in context with related work in the field of automatic service composition and service computing. Many models have been developed to facilitate automatic service composition. In one approach Web Services are modeled using DAML-S and DAML+OIL, with a subset of DAML-S defined in first-order logic. Other approaches model services using state transitions as in Roman [4] or Mealy machines in Colombo [3]. An interesting approach models services using Petri nets, and constructs a service net" with input and output places corresponding to a service's initial and final state, respectively [5]. In future many research will investigate the possibility of introducing a formal model to support validation and verification of a service composition.

The processes found by an automatic service composition algorithm will eventually be executed by a centralized or distributed execution engine. The distributed execution of processes and workflows is an active area of research. Self-Service [2] presents a peer-to-peer architecture that supports the distributed execution of service compositions using distributed routing in the peer-to-peer network.

By automated composition of Web services we mean the generation of a new composite service that interacts with a set of existing component services in order to satisfy given composition requirements. More specifically, we will assume that component services are described as

BPEL4WS processes.1 Given the descriptions of the component processes and the composition requirements, we automatically generate a new BPEL4WS process implementing the required composition.

Conclusion

Automatic composition of Web services is a challenging research problem. Due to increasing number and heterogeneity of available Web services we rely on service semantics to automatically compose new services. Web Services are software platform-independent applications that export a description of their functionalities and make it available using standard network technologies. Developing composite processes interacting with complex real world web services requires a time consuming analysis of the component services, both for what concerns their interaction protocol and the data structure of their messages. Moreover, it requires a detailed implementation of the new composite service that takes into account all the possible interaction evolution.

References

[1] Ting-Xin Song, Chun - Mei Wei , Automatic Composition of Web services Based on Rules Mapping Hubei University of Technology Wuhan, 430068, China

[2] Zhijun Ding, Junli Wang, Hong Song ,Al Planning for Web Service Automatic Composition Using Petri Nets, Shandong University of Science & Technology, 2007

[3] Muhammad Ahtisham Aslam, Jun Shen, Ren Auer Michael Herrmann 'Betriebliche Informationsysteme, Universitdt Leipzig, Germany, An Integration Life Cycle for Semantic Web Services Composition,2007

[4] Liquan Han, Zhongyu Xu,Qing'an Yao,College of Computer Science and Engineering, An Approach to Web Service Composition Based on Service-ontology, 2009

[5] Jiamao Liu, Chenhui Fan, Ning Gu, *Dept. of Computing and Information Technology, Fudan University,* Web Services Automatic Composition with Minimal Execution Price**,** Proceedings of the IEEE International Conference on Web Services.