2010 International Conference on Advances in Computer Engineering

Challenges (Research Issues) in Web Services

Demian Antony D'Mello
Department of Computer Science
St. Joseph Engineering College
Mangalore - 575 028, INDIA.
Email: demian.antony@gmail.com

Ananthanarayana V. S.
Department of Information Technology
National Institute of Technology Karnataka
Mangalore - 575 025, INDIA.
Email: anvs@nitk.ac.in

V. Lakshmi Narasimhan
Department of Computer Science
East Carolina University
Greenville, NC 27834, USA.
Email: narasimhanl@ecu.edu

***Abstract*—The Web services paradigm promises to enable rich, flexible and dynamic inter-operation of highly distributed and heterogeneous applications using Web standards. The ability to discover, select and compose inter-organizational and heterogeneous services available on the Web at runtime in order to facilitate application reuse is an interesting problem in Web services. An effective mechanism for discovery, selection and composition of Web services, based on the requester's complex functional and nonfunctional requirements is still an issue of open research. In this paper, authors explore the major Challenges (open problems) related to Web service description, discovery, selection and compositions.**

I. Introduction

A Web service [1] is defined as an interface which implements the business logic through a set of operations that are accessible through standard Internet protocols. The eXtensible Markup Language (XML) based protocols namely Universal Description, Discovery and Integration (UDDI) [2], Web Service Description Language (WSDL) [3] and Simple Object Access Protocol (SOAP) [4] are the *three* major building blocks of Web services. The conceptual Web services architecture [5] is defined which is based upon the interactions between *three* roles: *service provider*, *service registry* and *service requester*. The interactions among them involve the *publish*, *find* and *bind* operations. The limitations of registry based conceptual architecture are: (a) The lookup of Web services is based only on the syntactic matching of service functionality. (b) No support to distinguish advertised Web services. (c) Offers only limited support for Web service compositions.

II. Major Challenges in Web Services

Towards the widespread adoption of Web services, the following major issues (challenges) are to be addressed. (a) The effective mechanisms for dynamic lookup (discovery) of service descriptions from the service registry. (b) A mechanism to distinguish functionally similar Web services based on the requester's nonfunctional properties like QoS and service offers. (c) A mechanism to compose available Web services on the fly based on the requester's complex functional demands. (d) A mechanism to evaluate nonfunctional properties of composite Web services defined on composition patterns for publishing. These *four* major challenges are illustrated with a real world scenario in travel domain.

A. Motivating Example

Consider the travel scenario where the major actors are: the traveler, the service providers of various travel services and the travel agent service providers. Many travel service providers and travel agent service providers advertise their services with central repository (UDDI). In order to discover relevant travel services effectively, the Web service functionality needs be described in a quite natural form to minimize the ambiguity during matchmaking of service advertisements with service requests. Nowadays, travel service providers are growing enormously over the Web, which results in multiple services offering same functionality. To select the desired travel service for the requester, a mechanism is required to distinguish and rank the travel services.

Many a times travelers contact travel agents for the execution of their travel plans. Travel agents normally offer multiple, related services like train reservations, flight bookings, taxi booking etc. Sometimes travel agents also provide tour packages involving different travel services for the travelers. To be in competition with other travel agents, a travel agent enforces constraints (requirements) on the services to be selected for the individual activities of composite travel service i.e. tour package. In order to publish desired quality composite Web services with profit, a mechanism is required to identify suitable travel services for the constituent travel activities based on the nonfunctional constraints.

The travel service providers generally advertise a few related services for the travelers. The travelers sometimes require a set of multiple related or unrelated functions to be satisfied by the travel services. The lookup for such complex service request may fail due to unavailability of travel service for the travelers such needs. In such a scenario, complex service request needs to be executed by composing and coordinating different available travel services for the individual tasks to provide a complete tour package. To compose available services on the fly, a mechanism is required to generate an executable composition plan involving various Web services for the requested functionalities.

B. Web Service Discovery Mechanism

The increasing number of Web service providers over the Web has prompted the need for research in service description and discovery. Existing techniques for publishing and finding Web services (WSDL and UDDI) rely on static descriptions

978-0-7695-4058-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ACE.2010.75

of service interfaces which enable the requesters to find and bind services based on functional needs. The matchmaking mechanisms for discovery are explored through many ways such as keyword and category based, behavioral signature, domain description based, interface signature based, semantic description based, service behavior based, IOPE based, context information based, usability information based, quality based and personalization based approaches.

The discovery mechanisms normally find suitable Web services based on either functional or nonfunctional needs. It is necessary to represent service functionality in a natural language form instead of expressing functionality in terms of semantic rules, ontological concepts and syntactical forms. The natural way of describing service functionality enables the requester to frame the service request as per the actual need. To improve the performance (Recall) of Web service discovery, there is a need to define well-formed semantics for the description of Web service operation functionality. The functional semantics of operation should also contain objects involved in the operations, actions on objects, object qualifiers (attributes or features) and action nouns. To facilitate effective Web service discovery, the Web service architecture need to be augmented to store the knowledge required for operation description matchmaking. In order to describe Web service operations using functional semantics, the new XML elements are to be defined for the WSDL 2.0 document.

C. Web Service Selection Mechanism

Many a times the Web service discovery mechanism explores multiple Web services for the requester with no distinction. The requester has to select suitable Web service from functionally similar services based on his demands defined on either quality (QoS) or service offer. In B2B or B2C e-commerce scenario, the questions on QoS level of a Web service such as "can I get the result in 25 seconds?" usually is critical to the requester before a service is invoked. Similarly, question on service offers like "can I get best discount on service" is quite practical in B2C e-commerce scenarios. The Web service selection is the process of choosing one Web service from functionally similar Web services for the binding (execution). The Web service selection techniques are defined based on personalization, requester's trust policy and connection policy, requester's past experience with the Web service execution and Web service quality. The QoS plays a major role in ranking of functionally similar Web services which requires the augmentation of conceptual service oriented architecture. The QoS driven Web service selection approaches normally adopt matchmaking mechanisms which are defined on the requested QoS properties and Web service's advertised QoS properties. The QoS based Web service selection mechanism for the requester's simple QoS requirements rank the functionally similar Web services based on single or multiple QoS properties.

In order to distinguish Web services, there is a need to define QoS model in an unambiguous and precise way. Moreover the service offer model involving different service offers needs to be defined to distinguish functionally similar business driven (B2C scenario) Web services. The requester's QoS requirement can be considered for the selection involving mandatory QoS properties i.e. involving only AND operators. There is a need to define the Web service selection mechanism to distinguish and rank the functionally similar business Web services based on the service offers. In real world scenario, the requester may also impose choice based requirements i.e. involving OR combinations of QoS properties and service offers for the Web service selection. In order to select the most suitable Web service for the requester's nonfunctional requirements defined on QoS or service offers involving AND and OR combinations, a ranking mechanism has to be defined which compares the quality or profit offered by the providers of the functionally similar Web services. Towards the execution of selection mechanism, a model is required to represent the requester's QoS or service offer requirements involving AND and OR combinations and preferences. To facilitate QoS aware or service offer aware Web service publishing and selection, the conceptual Web service architecture needs to be augmented accordingly to facilitate effective selection of Web services.

D. Publishing the Composite Web Services

The composite Web services involving composition plan normally select Web services for the individual tasks at run time which satisfy local (task level) or global (end-to-end) QoS constraints. The QoS and service offers of such composite Web services are dependent on the individual Web services selected for each task of the composition plan. The dynamic nature of Web services enforces the providers of composite Web services to estimate and update (publish) the QoS and service offers regularly. To update the QoS and service offers of composite Web services, a composite service provider requires a tool to estimate QoS of composite Web service and to advertise a competitive service offer based on his requirements. The QoS of composition can be evaluated based on the requirements involving single QoS property or AND combinations of multiple QoS properties. While evaluating QoS of composition, the Web services are selected for the tasks based on either localor global QoS constraints. As the service offers also play a role in selection of business Web services, the composite business Web service has to be created by selecting good quality and profitable services for all the tasks of the composition.

Publishing of composite Web services involve publishing of its functional, QoS and service offer specific information into repository (registry). To obtain estimation on QoS and service offers, QoS and service offer aggregation schemes are required for the composite Web service involving different flow patterns (composition patterns). A selection mechanism has to be defined to select the most suitable Web service for the tasks of composition plan based on the provider's non-functional requirements and preferences. To facilitate selection mechanism, a model has to be defined which represents the provider's QoS and service offer requirements involving AND and OR operators with preferences. A model is also required to

represent the composition plan involving various composition patterns in computer memory to facilitate QoS and service offer estimation mechanism.

E. Web Service Composition Mechanism

The requester's service request sometimes includes multiple functionalities to be satisfied by the Web service. In many cases the Web service has a limited functionality which may not meet the requester's complex functional needs. To achieve complex business goals in real world applications, the execution of multiple Web services should be orchestrated through service composition. The Web service composition can be defined as the creation of new Web service by combining the available services (service operations) that realizes the complex service request. The service composition techniques are broadly classified as *Static* and *Dynamic* composition based on the time when the Web services are composed. Static composition takes place during design time when the architecture and the design of the system is planned. Dynamic composition takes place at run time when the requested service is not provided by the single provider. The effective dynamic Web service composition is a major challenge towards the success of Web services. The following *seven* different issues have a large impact on dynamic Web service composition. They are: (a) Describing Web services and complex service request for effective composition. (b) Generation of composition plan for the complex service request. (c) Modeling (specification) of composition plan (orchestration models). (d) Selection of Web services for the composition. (e) Coordination and Conversation modeling. (f) Execution of composition and (g) Transaction management.

The description of complex service request and Web services should facilitate the generation of composition plan for execution. The functional semantics, domain ontology and parameter ontology based Web service description methods are used for Web services composition. The generation of composition plan for the requester's complex service request at run time is a crucial issue in composition. The Web services are mainly composed by generating the composition plans based on syntax (input and output), constraints, contextual information, user satisfaction and interaction, semantic (ontology centric), behavior (process) signature and business rules. The composition models describe the flow (order) of execution of services which participate in the composition. The popular models (languages) for specification of Web service composition are Business Process Execution Language for Web Services (BPEL4WS) and Web Service Flow Language (WSFL). The other composition and interaction models are: Petri Nets and Colored Petri Nets (CPN), Labeled Behavior Diagrams (LBD), Mathematical model, UML Activity and State Chart diagrams, Finite automata, Markov process model, Temporal Logic of Actions (TLA), Web Services Flow Graph (WSFG) and Hierarchical Task Network (HTN).

The Web services for the individual tasks (activities) of the composition are normally selected based on local or global (end to end) QoS requirements. In real world applications the interactions among Web services is more complex than single standalone Web service invocations. The order of message exchanges between different Web services need to be coordinated for the successful execution of composition. The Web service conversations are normally modeled using state machines. WS-coordination is the specification which creates a framework for supporting protocols for central and distributed coordination. The execution of composition involves binding and execution of individual services which are identified for various tasks. The important requirement of a composition is to check whether the service composition ensures atomicity and application consistency as the services are sometimes long running (long lived) and normally heterogeneous in nature. Thus it is difficult to enforce transactions with conventional ACID properties in a service composition. A new transaction model has to be defined in order to support both the properties of ACID transactions and relaxed ACID properties (compensating transactions) as demanded by the loosely coupled, long lived, heterogeneous business transactions. WS-Transaction and Business Transaction Protocol (BTP) are the major specifications to coordinate Web services into applications.

The UDDI based Web service business model needs to be augmented through inclusion of broker to solve the important issues related to service composition. The UDDI business model permits providers to advertise atomic or composite Web services involving interdependent or independent activities. The execution order (flow) of Web service operations provides significant information to generate composition plan for the requester's complex service request at run time. Thus a need arises to describe operation flow of Web service within the WSDL document which can be used to compose available Web services at run time to satisfy the requester's complex functional needs.

III. Conclusion

The effective mechanisms for the key issues related to Web service publishing, discovery, selection and compositions encourage the widespread adoption of Web services. The various challenges explored in this paper requires an augmentation (inclusion of new architectiral roles and operations) to the conceptual Web service architecture through The augmented architecture should support effective mechanisms for Web service description, discovery and composition with a selection strategy to distinguish functionally similar Web services.

References

[1] Gustavo, A., Casati, F., Kuno, H. and Machiraju, V. (2009). "Web Services: Concepts, Architecture and Applications." *Springer Verlag*, 2009.

[2] Riegen, C.V. et al (2002). "UDDI Version 2.03 Data Structure Reference." *OASIS Open 2002-2003*, http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm.

[3] Booth, D. and Liu, C. K. (2007). "Web Services Description Language (WSDL)." *Version 2.0 Part 0: Primer. W3C Recommendation*, http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626.

[4] Mitra, N. and Lafon, Y. (2007). "SOAP Version 1.2 Part 0: Primer (Second Edition)." *W3C Recommendation*, http://www.w3.org/TR/2007/REC-soap12-part0-20070427/.

[5] Kreger, H. (2001) "Web Services Conceptual Architecture (WSCA 1.0)." www.ibm.com/software/solutions/webservices/pdf/wsca.pdf.