A Comparative Evaluation of State-of-the-Art Approaches for Web Service Composition

Sayed Gholam Hassan Tabatabaei, Wan Mohd Nasir Wan Kadir, Suhaimi Ibrahim
Department of Software Engineering, Faculty of Computer Science and Information Systems,
University of Technology Malaysia (UTM), 81310 Skudai, Johor, Malaysia
gtsayed2@siswa.utm.my, wnasir@utm.my, suhaimiibrahim@utm.my

Abstract

In today's Web environment, many enterprises decide to implement and publish their applications on the Internet using Web services technology. In many cases, a single service is not sufficient to fulfill the user's request. To solve this problem, services should be combined together. Therefore, composition of Web services is one of the recent critical issues. A number of approaches have been presented, to tackle this problem. In this paper, we categorize these approaches into four categories (Workflow-based, AI-planning based, Syntactic-based, and Semantic-based). Then, we compare these approaches based on some criteria (like QoS, scalability, and correctness). Investigation of that classification will help researchers who are working on service composition to deliver more applicable solutions.

1. Introduction

Nowadays, the term "Web service" has been used very often. The W3C [1] defines a Web service as "a software system designed to support interoperable Machine to Machine interaction over a network."

Basically, Web service operation can be described as follows. First of all, a *client* program via a yellow page (UDDI) [2] finds a *Web services server* that can fulfill certain requirements, and acquire a detailed specification from WSDL [3] about the service. Then, the client sends a *request* to the server through a standard message protocol (SOAP) [4], and in return receives a *response* from the server.

Web services are a set of tools that can be used in a number of styles such as RPC (present a distributed function call interface), SOA (where the basic unit of communication is a message, rather than an operation), and REST (where the focus is on interacting with stateful resources, rather than messages or operations).

Currently, an increasing number of companies and organizations implement their applications over Internet. Thus, the ability to select and integrate inter-organizational and heterogeneous services on the Web efficiently and effectively at runtime is an important step towards the development of the Web service applications. Recent researches study how to specify (in a formal and expressive enough language), compose (automatically), discover and ensure the correctness of Web services. In this paper we focus on Web Service composition (WSC).

When no atomic Web service (WS) can fulfill the user's requirements, there should be a possibility to combine existing services together in order to satisfy the request requirement. This trend has inaugurated a considerable number of research efforts on the Web service composition both in academia and industry.

A composite service, in many ways, is similar to a workflow [5]. The definition of a composite service includes a set of atomic services together with the control and data flow among the services. Similarly, a workflow has to specify the flow of work items. Some approaches, based on AI planning, consider WS as a software component that takes the input data (preconditions) and produces the output data (effects). In the research related to WSs, several initiatives have been conducted with the intention to provide platforms and languages for WSC such as Business Process Execution Language for Web Services (BPEL4WS) [6]. Currently, some languages have ability to support semantic representations of the WSs on the Internet such as the Web Ontology Language for Web Services OWL-S [7] and the Web Service Modeling Ontology WSMO [8].Considering all of these efforts, yet the WSC is still a highly complex task.

In this paper, we focus on the WSC problem and offer a survey of recent approaches that provide automation to WSC. We then compare them with respect to a set of criteria. By offering this overview and classification of existing proposals for WSC we hope this research helps researchers to deliver well-built approaches.

978-0-7695-3372-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSEA.2008.68

2. Classification of the WSC approaches

We can classify the WSC approaches using the following four aspects. Notice that, the boundaries between these four aspects of classification are not always strictly defined. BPEL4WS, for example, can be also considered as a workflow-based approach.

2.1. Workflow-based WSC approaches

Workflow-based composition methods can be distinguished to the static and dynamic workflow generation [9]. The *Static Composition* means that the requester before starting the composition planning should build an abstract process model. However, in *Dynamic Composition*, creating process model and selecting single WSs are done automatically. In this section we describe two principal approaches, namely:

- *EFlow* [10] is a platform for the specification, enactment and management of WSC which uses a static workflow generation method. Indeed, WSC is modeled by a graph that defines the order of execution among the nodes in the process. The graph is created manually but it can be updated dynamically. The graph may include service (represent the invocation of WS), decision (specify the alternatives and rules controlling the execution flow) and event nodes (enable service processes to send and receive several types of events). Arcs in the graph denote the execution dependency among the nodes. The definition of a service node contains a search recipe that can be used to query actual service. As the service node is started, the search recipe is executed, returning a reference to a specific service.

- *Polymorphic Process Model (PPM)* [11] uses a method that synthesizes the static and dynamic WSC. The static setting is supported by reference process-based multi-enterprise processes. These processes encompass abstract sub processes that have functionality description but have not been implemented. The abstract subprocesses are implemented by service and bined at runtime. The dynamic part of PPM is supported by service-based processes. Here, a service is modeled by a state machine that specifies the possible states of a service and their transitions. Transitions are caused by service operation invocations or internal service transitions. In the setting, the dynamic service composition is enabled by the reasoning based on state machine.

2.2. AI-Planning-based WSC approaches

Currently, several approaches based on AI planning have been presented to solve the problem of WSC. Most of these approaches rely on the model of state-transition system. In this system there are finite or recursively countable set of states, actions and events along with a transition function that maps a state, action, event tuple to a set of states. The goal of planning is to find which actions to apply to which states in order to achieve some objective, starting from some given situation. Basically, classical planning is based on the initial modeling of the STRIPS [27] system.

In the following we introduce some well-known WSC approaches based on AI planning. Excellent surveys of AI-planning-based approaches to tackle the problem of WSC can be found in [9, 28].

- *Situation Calculus* is a first-order language for reasoning about action and change. Golog is a high-level logic programming language based on the situation calculus that enables the representation of complex actions. It builds on top of the situation calculus by providing a set of extra-logical constructs for assembling *PrimitiveActions*, defined in the situation calculus, into *ComplexActions* that are compositions of individual actions.

McIlraith et. al. [29, 30, 31] adapt and extend the Golog language for automatic construction of WSs. In fact, this approach is based on the notion of generic procedures. The authors address the WSC problem through the provision of high-level generic procedures and customizing constraints.

One advantage of using situation calculus is the additional expressivity and the ability to do arbitrary reasoning about first-order theories. However, it is impossible to describe non-functional attributes of such programs or use these attributes for flexible matching.

- *HTN-DL*: Sirin [32] proposes the HTN-DL formalism which combines Hierarchical Task Network (HTN) planning, and Description Logics (DL) to automatically overcome the problem of WSC.

The hierarchical structure of HTN planning domains can describe composite service descriptions. Composite Web services can be mapped to HTN methods whereas atomic WSs are mapped to HTN operators. HTN-style domains fit in well with the loosely coupled nature of WSs. The DL is used to describe both actions and states with an expressive knowledge representation language. The service categorization and non-functional attributes of services are described in a task ontology that allows flexible matchmaking. The state of the world is also represented as a DL knowledge base.

2.3. Syntactic-based WSC approaches

Currently there are two main approaches in the field of syntactic-based WSC [12]:

Web Service Orchestration: combines available WSs by adding a central coordinator (the orchestrator) that is responsible for invoking and combining the single subactivities.

Web Service Choreography, instead does not assume a central coordinator but rather defines complex tasks via the definition of the conversation that should be undertaken by each participant; the overall activity is then achieved as the composition of peer-to-peer interactions among the collaborating WSs. A Choreography also describes the external visible behavior of the WS.WS-CDL [13] is example of this approach.

One of the most important orchestration languages namely BPEL4WS is defined as follows.

- *BPEL4WS*: This syntactic-based language was designed to enable the coordination and composition of a set of WSs. Moreover, this language is based on WSDL [3], which is essentially an interface description language for WS providers. In fact, BPEL4WS is a merge between XLang and WSFL, but all of them are considered as a web service flow language [14]. WSC using BPEL4WS enables the definition of a new web service by composing a set of existing ones. The interface of the composite service is described as a collection of WSDL *PortTypes*.

A BPEL4WS process defines the roles involved in a composition as abstract processes. A buyer and a seller are examples of two roles. They are expressed using partner link definitions. We can have a role for each Web service that is composed and does some activity. In order to integrate services, they are treated as partners that fill roles [15]. In sum, business process is used to create an organizer that point to each service endpoint that will be actually executed.

2.4. Semantic-based WSC approaches

The Semantic Web [16] allows the representation and exchange of information in a meaningful way, facilitating automated processing of descriptions on the Web. Annotations on the Semantic Web express links between information resources on the Web and connect information resources to formal terminologies. These connective structures are called ontologies.

Ontologies are used as data models throughout these types of approaches, meaning that all resource descriptions and all data interchanged during service usage are based on ontologies. The extensive usage of ontologies allows semantically enhanced information processing and support for interoperability. In this section we consider two principal approaches, namely:

- *OWL-S* is an OWL service ontology for describing various aspects of Web services [17]. OWL-S has tried to adopt existing Semantic Web recommendations yet still maintain bindings to the world of Web services by linking OWL-S descriptions to existing WSDL descriptions [18]. In the following, we describe the four top-level concepts of the OWL-S ontology :

SERVICE: The SERVICE concept serves as an organizational point of reference for declaring WSs. Every WS is declared by creating a SERVICE instance.

SERVICE PROFILE: declares what a SERVICE does in order to advertise and serves as a template for service requests at a high level, therefore enabling discovery and matchmaking.

SERVICE MODEL: SERVICE could be described by a SERVICE MODEL which describes how a service works to enable invocation, enactment, composition, monitoring, and recovery.

SERVICE GROUNDING: In order to map to the Web service world, an OWL service can *support* a grounding which maps the constructs of the PROCESS MODEL to detailed specifications of message formats, protocols, and others.

- *WSMO* defines a model to describe semantic WSs, based on the conceptual design set up in the WS Modeling Framework WSMF [20]. Following the key aspects identified in the Web Service Modeling Framework, WSMO identifies four top-level elements as the main concepts [19](See Figure 1):

Ontologies: provide the (domain specific) terminologies used and are the key elements for the success of Semantic Web services. Furthermore, they use formal semantics to connect machine and human terminologies.

Web services: are computational entities that provide some value in a certain domain. They are described from three different aspects: non-functional properties, functionality and behavior.

Goals: describe aspects related to user desires with respect to the requested functionality, i.e. they specify the objectives of a client when consulting a WS. Thus they are an individual top-level entity in WSMO.

Mediators: describe elements that handle interoperability problems between different elements, for example two different ontologies or services. Mediators can be used to resolve incompatibilities appearing between different terminologies (data level), to communicate between services (protocol level), and to combine Web services and goals (process level).

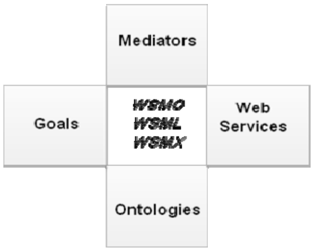

Figure 1. Four top-level elements of WSMO.

Besides these main elements, *Non-Functional* properties are used in the definition of WSMO elements that can be used by all its modeling elements. Furthermore, there is a formal language to describe ontologies and Semantic Web services called WSML (Web Service Modeling Language). To introduce aspects of Semantic Web services in WSMO, the Meta-Object Facility (MOF) [21] specification is used. In addition, WSMX (Web Service Modeling eXecution environment) is the reference implementation of WSMO, which is an execution environment for business application integration. [22].

3. Comparative evaluation

In this section we compare the above WSC approaches with respect to the following criteria. We claim that, any approach to WSC should satisfy these set of criteria. We consider an approach as a "*good"* quality approach, if the approach can provide all aspects of the criteria. If an approach can provide part of what the criteria expects, is considered as an *"average"* quality approach based on that criteria. If an approach does not provide what the criteria expects, is considered as a "*low"* quality approach regarding that criteria. The result can be seen in Table 1.

3.1. QoS

Currently, the *Quality of Service* (QoS) is one of the critical issues in the WSC area. When referring to QoS, nonfunctional properties such as performance, cost, or reliability are intended. Since a composed service uses other services to form itself, its quality depends on the WSs it uses. To be accepted by its customers, a business should try to provide good quality regarding the customers' requirements to a composed WS.

QoS aspects are considered at the time of WS candidates selection for a composition. By defining aggregation formulas for several QoS aspects which are applied to simple composition patterns, the whole workflow pattern of a composed service can be collapsed stepwise, and each time the most suitable collection of simple services is selected. As QoS information assigned with each basic service, *performance*, *reliability*, and *availability* were chosen.

- *Performance:* This represents how fast a Web service request can be completed. According to [23], performance can be measured in terms of throughput, latency, execution time, and transaction time. The response time of a Web service can also be a measure of the performance. High-quality Web services should provide higher throughput, lower latency, lower execution time, faster transaction time and faster response time.
- *Reliability:* This represents the ability of a Web service to perform its functions (that is, to maintain its Web service quality). It can be measured by the number of failures of the Web Service in a certain time interval.
- *Availability:* the probability that a WS is available at any given time, measured as the percentage of time a WS is available over an extended period of time.

The management of QoS when composing WSs requires a careful consideration of the QoS criteria of the constituent WSs. To enable the specification and monitoring of QoS aspects like performance, financial, reliability, and availability, various approaches have been developed. An excellent research for considering QoS aspects in WSC can be found in [24]. Most of workflow based approaches like EFlow neglect specification of nonfunctional QoS properties such as *security*, *dependability*, or *performance*. In AI planning approaches like Situation Calculus, a planning operator cannot represent such information. However, HTN-DL by using ontology that allows flexible matchmaking, tries to tackle this problem. In addition, BPEL4WS does not directly support the specification of most QoS measures. However, in OWL-S, QoS measures such as availability are specified as service parameters in the WS description definition, but the specification of metrics and guarantees is missing. Moreover, there is no way to specify functional relations between metrics and therefore quality-aware WS discovery is not feasible [12].Finally, QoS (Nonfunctional properties) are applicable to all the definitions of WSMO elements such as *Ontologies*, *Web services*, *Goals*, and *Mediators*. Which QoS properties apply to which WSMO element is specified in the description of each WSMO elements. Therefore, this approach has a good QoS.

3.2. Automatic composition

Many composition approaches aim to automate composition, which promises faster application development and safer reuse, and facilitates user interaction with complex service sets. With automated composition, the end user or application developer specifies a goal (a business goal expressed in a description language or mathematical notation) and an "intelligent" composition engine selects adequate services and offers the composition transparently to the user. The main problems are in how to identify candidate services, compose them, and verify how closely they match a request [25]. Generally, we cannot assign any of the above approaches as an automated approach. Although, most of these approaches like HTN-DL, OWL-S and WSMO can be assigned as a semi automated approach.

3.3. Composition scalability

This represents the ability of the WS to process multiple requests in a certain time interval. Composing two WSs is not the same as composing ten or more WSs. In a real-world scenario, end users will typically want to interact with many WSs while enterprise applications will invoke chains of possibly several hundred services. Thus, one of the important issues is how the proposed approaches scale with the number of WSs involved. It can be measured by the number of requests resolved in a certain time interval.

The HTN-DL has a tolerable scalability. This is due to the fact that DL reasoner Pellet used in HTN-DL is optimized to handle large number of instances. In BPEL4WS, since XML files have increased a lot, WSC is a bit tiresome. BPEL4WS composition can be modularized, because this approach is recursive. But, BPEL4WS has no standard graphical notation. Some orchestration servers offer graphical representation for descriptions, such as UML, but they don't map one-to-one to complex constructs of BPEL4WS. Finally, OWL-S and WSMO have similar issues. The Web component approach achieves good scalability with class definitions, but requires additional time for mapping and synchronization between class definitions and XML.

3.4. Correctness

Verifying correctness depends on the WS and composition specifications. The composition of WSs may lead to large and complex systems of parallel executing WSs. An important aspect of such systems is the correctness of their behavior. Situation Calculus and HTN-DL, because of their solid mathematical basis for compositions generated from the resulting planning domain, are good in the terms of correctness verifiability. All other approaches offer no direct support for the verification of WSC at design time. For example, BPEL is a Turing complete language dealing more with implementation than specification, and thus it's difficult to provide a formalism to verify the correctness of BPEL4WS flows [26].

Table 1. Comparing Web service composition approaches

	Criteria				**Overall Result**
Approaches	**QoS**	**(Semi) Automatic**	**Scalability**	**Correctness**	
EFlow	Low		Low	Low	**Low**
PPM	Low		Low	Low	**Low**
Situation Calculus	Low		Good	Good	**Average**
HTN-DL	Average	√	Good	Good	**Good**
BPEL4WS	Average		Average	Low	**Average**
OWL-S	Good	√	Good	Low	**Good**
WSMO	Good	√	Good	Low	**Good**

4. Conclusion

This paper has aimed to provide an overview and compare recent progress in WSC. We classify these approaches to four categories. But we cannot claim that this classification is exhaustive. In each category, we give the introduction and comparison of selected approaches. The workflow-based approaches are usually used in the situation where the request has already defined the process model, but automatic program is required to find the atomic services to complete the requirement. The AI-planning based approaches deal with WSs as planning operators and use a causal planner to generate WSC. The syntactic-based approaches concentrate on two main approaches, namely: orchestration and choreography. Choreography languages are still in an introductory phase of definition. In Semantic-based approaches, ontologies are used as data models throughout these types of approaches, meaning that all resource descriptions and all data interchanged during service usage are based on ontologies. The main problems with most of these approaches to WSC are the verification of correctness of WSC and the analysis of QoS aspects.

5. Acknowledgement

This research is supported by the Ministry of Science & Technology and Innovation (MOSTI), Malaysia and Universiti Teknologi Malaysia (UTM).

6. References

[1] Web Services Glossary. (2004). [Online]. Available: http://www.w3.org/TR/ws-gloss/

[2] New to SOA and Web services. [Online]. Available: http://www.ibm.com/developerworks/webservices/

[3] WSDL v1.1. (2001). [Online]. Available: http://www.w3.org/TR/wsdl

[4] SOAP v1.2. (2007). [Online]. Available: http://www.w3.org/TR/soap12-part1/

[5] F. Casati, M. Sayal, and M.C. Shan, "Developing E-Services For Composing Eservices", *In Proceedings of 13th International Conference On Advanced Information Systems Engineering (Caise)*, Springer Verlag, Interlaken, Switzerland, June 2001.

[6] T. Andrews et.al., BPEL v1.1.(2007).[Online].Available: http://www.ibm.com/developerworks/library/specification/ws-bpel/

[7] A. Ankolekar, DAML-S: "Web Service Description for the Semantic Web". *In Proceedings of ISWC'02*, ser. LNCS, vol. 2342, Springer Verlag, 2002, pp. 348–363.

[8] WSMO working group. [Online]. Available: http://www.wsmo.org

[9] J. Rao and X. SU, "A survey of automated web service composition methods". *In Proceedings of SWSWPC,* LNCS 3387, Springer, 2005, pp. 43-54.

[10] F. Casati, S. Ilnicki, and L. Jin, "Adaptive and dynamic service composition in EFlow". *In Proceedings of 12th International Conference on Advanced Information Systems Engineering (CAiSE)*, Springer Verlag, Stockholm, Sweden, June 2000.

[11] H. Schuster et.al, "Modeling and composing service-based and reference process-based multi-enterprise processes", *In Proceeding of 12th International Conference on Advanced Information Systems Engineering (CAiSE)*, Springer Verlag, Stockholm, Sweden, June 2000.

[12] M. Beek, A. Bucchiarone, and S. Gnesi, "Web Service Composition Approaches : From Industrial Standards to Formal Methods". *In Proceedings of Int'l Conf. On Internet and Web Application and Services (ICIW'07),* IEEE, 2007.

[13] N. Kavantzas, D. Burdett, and G. Ritzinger, WSCDL v1.0. (2004). [Online]. Available: http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/

[14] W.M.P. Van der Aalst, "Don't Go with the Flow: Web Services Composition Standards Exposed", *Intelligent Systems*, IEEE, 18(l), 2003, pp. 72-76.

[15] D.J. Mandel and S.A. Mcllraith, "Adapting BPEL4WS for the Semantic Web Bottom-up Approach to Web Services Interoperation", *In Second International Semantic Web Conference (ISWC)*, Sanibel Island, Florida, 2003.

[16] Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web*. ScientificAmerican, 2001, pp. 34–43.

[17] Fensel, D. et.al, *Enabling Semantic Web Services*, Springer, Berlin, 2007.

[18] *OWL-S: Semantic Markup for Web Services,* W3C Member Submission, (22 November 2004). [Online]. Available: http://www.w3.org/Submission/OWL-S/.

[19] S. Narayanan and S. McIlraith, "Simulation, verification and automated composition of web services", *In Proceedings of the 11th International World Wide Web Conference (WWW2002)*, Honolulu, Hawaii, 2002.

[20] Battle, S. *Semantic Web Services Framework (SWSF)*.

[21] Object Management Group Inc. (OMG), Meta Object Facility (MOF) Specification v1.4, 2002.

[22] WSMX working group. [Online]. Available: http://www.wsmx.org

[23] Rajesh, S. and D. Arulazi, *Quality of service for Web services – demystification, limitations, and best practices.*

[24] D. ThiBen and P. Wesnarat, "Considering QoS Aspects in Web Service Composition", *In Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06)*, 2006.

[25] N. Milanovic and M. Malek, "Current solution for Web service composition", *IEEE*, 2004.

[26] X. Fu, T. Bultan, and J. Su, "Analysis of Interacting BPEL Web Services". *In Proceedings of WWW'04*. ACM Press, 2004, pp. 621–630.

[27] R. E. Fikes and N. J. Nilsson. "Strips: A new approach to the application of theorem proving to problem solving." Readings in Planning, Kaufmann, San Mateo, CA, 1990.

[28] S. Oh, D. Lee, and S. Kumara, "A Comparative Illustration of AI Planning-based Web Services Composition", *In Proceedings of the 2005 ACM SIGecom Exchanges*, Vol. 5, No.5, December 2005.

[29] S. McIlraith and T. C. Son. "Adapting Golog for composition of Semantic Web services", *In Proceedings of the 8th International Conference on Knowledge Representation and Reasoning(KR2002)*, Toulouse, France, April 2002.

[30] S. McIlraith, T. C. Son, and H. Zeng. "Semantic Web services", *IEEE Intelligent Systems*, 16(2), March/April 2001.pp. 46–53.

[31] S. Narayanan and S. McIlraith, "Simulation, verification and automated composition of Web service", *In Proceedings of the 11th International World Wide Web Conference*, ACM, Honolulu, Hawaii, USA, May 2002. Presentation available at http://www2002.org/presentations/narayanan.pdf.

[32] E. Sirin. "Combining Description Logic Reasoning with AI Planning for Composition of Web Services", PhD thesis, University of Maryland, 2006.