UML based Modeling of Web Service Composition- A Survey

Irum Rauf
International Islamic University, Islamabad, Pakistan
irumrauf@hotmail.com

Muhammad Zohaib Z Iqbal
International Islamic University, Islamabad, Pakistan
zohaib.zafar@iiu.edu.pk

Zafar. I. Malik
Ministry of Education, Government of Pakistan, Islamabad, Pakistan
zafarimalik@hotmail.com

Abstract

Web Service Composition is an emerging trend in the field of Service Oriented Architecture, where a new web service is developed using existing web services. Developing a composite web service is a complex task. There are a number of reported techniques for modeling web service compositions to specify the exact requirements, identify errors, and eliminate the conflicts in the development of a composite web service at design level. In this paper, we study and classify various UML based approaches for modeling web service compositions. We also present a detailed analysis of all such approaches.

Keywords

Web Service Composition, UML, UML Profile, MDA

1. Introduction

Web Services have revolutionarized the way our systems interact. Web Services are defined in Web Service Architecture document by W3C [13] as software systems that interact over a network from one machine to another. Their descriptions are given in a machine readable-format. Other software systems can discover them over the network using internet standard protocols. The way other systems interact with the web service is prescribed in its description using SOAP messages which are conveyed using HTTP and other web related standards [13]. Wide range of web services have been developed and published over the internet. These can be discovered and used by the consumer as required. The presence of large number of Web Services gave rise to the composition of web services.

A composite web service may be composed of elementary or other composite services. With web service composition, developers can create, run, adapt, and maintain services that are dependent on other services [23]. A composite web service contains functionality that is implemented by several other services. Several services that are developed under different circumstances by different vendors and with different specifications can be used to form a composite web service. Thus, developing a composite web service is considered as a difficult task. Composition of web services has gained considerable momentum and much R& D efforts are being put into this field.

Different type of languages have been proposed and established for describing service composition from different viewpoints [23]. The composite web services are generated using BPEL4WS [26], WSFL [24], XLANG [25]. Business Process Execution Language for Web Services (BPEL4WS) is an XML based notation for providing semantics and describing automated business processes that communicate through web services.

Previously, surveys have been conducted in this area providing a solid background for future researches. Rao and Su [27] present a survey on automated web-service composition methods. These composition methods are classified under 'Workflow technique' and 'AI Planning'. In [28], Hull and Su identify tools and techniques available for design and analysis of composite web services.

Dustdar and Schreiner [16] present a survey on web service composition. They present different composition strategies namely Static vs. Dynamic service composition, Model- Driven service composition, Declarative service composition, Automated vs. Manual service composition and Context based Web Service Discovery and Composition. In this paper, Dustdar and Schreiner have talked about model-based composition but that study does not adequately cover modeling techniques of UML.

We have focused our work on providing comprehensive survey of approaches that involve UML as modeling language. With the advent of MDA, UML has also gained more acceptances at industrial level as MDA talks about designing web services at a more abstract level to that in a technology specific manner [29]. Modeling at a higher level of abstraction results in longevity of the business process, besides it is easier to find errors

978-0-7695-3302-5/08 $25.00 © 2008 IEEE
DOI 10.1109/SERA.2008.39

and debug an application at design level rather than at code level.

UML has emerged as a standard language for specifying, visualizing, constructing and documenting of a software system [21]. We have covered in detail various UML based web service composition techniques. These techniques are classified in three categories. These classifications are explained in Section 2 along with the respective approaches. Section 3 provides analysis of these approaches and the comparison table. Section 4 concludes the paper.

2. UML based modeling of WS Composition: A Classification

UML provides different diagrams to specify a software model. In [22], thirteen different diagrams are presented to specify various aspects of a software model. These diagrams are organized in two broad categories namely: Structure Diagrams and Behavioral Diagrams.

Our work revolves around UML based modeling of web service compositions so we have classified our work based on these UML categories:

- Structure based WSC Modeling
- Behavior based WSC Modeling
- Hybrid WSC modeling

Structure based WSC Modeling

Structure Diagrams in UML represent static view of software. These involve Class, Component, Object, Composite Structure, Deployment, and Package Diagrams [22].

Web Service Composition is more of a behavior-oriented process. We were unable to find any approach in the existing literature that covers only the static part of the web service composition. Authors have worked on the modeling of the static aspect of web service composition, but that is done along with behavioral modeling. All such approaches are covered in section 2.3 that describes Hybrid WSC modeling.

Behavior based WSC Modeling

In UML, behavior diagrams specify behavioral concerns of software. These involve Activity, Use Case, State Machine, Sequence, Collaboration, Interaction Overview, and Timing Diagrams [22].

Behavior-based modeling of web-service composition covers those approaches in which only behavioral aspects of a Web Service are considered.

Castro et. al [1] proposed a service composition model that can be mapped to languages like BPEL and can be integrated with different aspects of Web Information System (WIS). The proposed MIDAS framework provides a methodology to model both the hypertext and behavioral aspects of WIS. Here, a platform independent model is developed for behavioral aspects of WIS. Services are taken as modeling elements. Starting from user requirements, services are identified and service compositions are modeled. ‘Conceptual User Services’ are special kind of use-cases that represent services. The functionality needed by them is shown in ‘User Services’. Conceptual user services are further composed of ‘basic user service’ and ‘composite user service’. ‘Service Process Model’ is an activity diagram that represents process to complete a ‘Conceptual User Service’. The whole system’s view is represented by ‘Service Composition Model’. ‘Business Collaborations’ are represented by ‘Swim Lanes’ in activity diagram. These models are then mapped to different web service technologies using different tools. The authors discuss a case study of Medical Image Management to show the application of their approach. The technique is automated however it does not address any service composition patterns and neither it caters any data and control conflicts that may arise during composition. The technique also does not provide any feedback mechanism.

Gronmo and Solheim[3] propose some extensions to compose a web service with UML. Authors identify some web services workflow patterns and provides extension in UML to model these patterns. The patterns identified are Web Service Call, Loop, Data Transformation and Alternate Services. The proposed solutions contain UML activity diagrams along with required extensions. The paper addresses the service composition patterns, but does not provide any feedback mechanism for refinement.

Benatallah et. al [5][6] focus on composition of web services in a peer-2-peer environment. Self-Serv is based on two major concepts: Composite Service and Service Container. Composite Services combine multiple component services. Operation of composite services is represented with the help of State Charts. A state can be either a basic or a compound state. Service Container is a service that contains several other services- either elementary or composite. These services contain common capabilities and are substitutable. A state coordinator is created for each state that takes information from routing tables and receives notifications from other state coordinators. It is also responsible for invoking services once all the pre-conditions are met. and notifying other state-coordinators once the execution is complete. The types of messages exchanged are 'control-flow notifications', in which the messages are exchanged between the coordinators, and 'service invocation messages', in which the messages are exchanged between the coordinator and the component services. The work is explained by taking Travel Planning Service as a Case study. The paper provides an important technique for web service compositions using statecharts. However, it does not address service composition patterns and does not cater static aspects of the composition.

Pathak et al [9] propose a framework for modeling web service composition. Specifications are provided using UML state machines. These UML state machines can be annotated with appropriate domain ontologies and non-functional requirements. They are then translated to Finite State Automata (FSA). Service providers publish their services using OWL-S and WSDL. These are also translated to FSA and analyzed based on user feedback. End-user can refine their specifications using the feedback. This is an iterative process. In an extension to this work [10], Phathak et al discuss in detail service composition in MoSCoE using Symbolic Transition System. However, this approach does not cover modeling in UML at a broader level. The approach only discuses writing specifications in UML state machines that are later translated into STS.

UML profile for OWL-S is presented by Timm and Gannod in [11]. Developers can model their OWL-S descriptions in UML. The XMI specifications of the UML models are translated into OWL-S using XSLT transformations. It is specified for atomic processes. This work is extended by Barrett and Pahl in [12] for web service composition. Interfaces are modeled as activity diagrams from UML models and distribution patterns are applied on them. Semantic matching is done and model is executed. TOPMAN is presented as a tool for this approach. This approach caters semantics of web service compositions but is dependent on a profile. It also does not cater feedback mechanism.

Hybrid WSC modeling

The section covers the approaches that cater both structural and behavioral aspects of web service composition in their model.

In the approach discussed by Skogan et. al [2] initially a preliminary model of the new (composite) web service is developed with two sub models; one with class diagrams that model the interface of the web service and the other using activity diagrams that model the composition operations. Web services are identified by textual description and names. The WSDL descriptions of the candidate services are extracted and are transformed to UML. UML Profiles are used to tag necessary information like Input/Output messages and operation signatures. Control and data flows can also be properly represented. Transformation from this UML profile model to XML document is performed. The XML document is used as input to the execution engine. Such transformations can be performed by various transformation tools. Finally, the interface of new web service is modeled and its corresponding WSDL is generated. The methodology has been applied on gas dispersion emergency case as a case study.

Numerous approaches have been discussed in literature that model BPEL4WS with UML, since it is a widely accepted language for web service composition. Gardner et al [7] [14] provide UML 1.4 profile for BPEL 1.0 covering most of the aspects of Web Service Composition. Ambühler [15] extends this profile and propose UML profile for UML 2.0. He also presents the implementation of the profile in Rational Software Modeler and transformation rules are implemented using transformation engine as an Eclipse plug-in.

Thone et. al [8] discuss and evaluate BPEL4WS. They propose UML-WSC profile which is mainly based on activity diagrams. UML activity diagrams represent dynamic part of the composition, whereas, static part is represented by UML class diagrams. States are stereotyped as service states, transform states and object flow states. 'Object flow states' represent flow of XML messages, 'Transform states' represent structural transformation on messages and 'Service states' represent call operations. Stereotyped UML class diagrams are used to describe operations and parameters of available web services.

Gronmo and Jaeger [18] describe four phases for development of web service composition. These phases include modeling, discovery, selection and deployment of composite web service. UML activity models show control and data flow whereas ontology concepts and QoS characteristics are represented by class models. UML profiles are used to capture domain ontologies. Services are discovered on matchmaking of semantic descriptions and are ranked based on QoS characteristics. Finally a concrete composition model is generated that contains different descriptions of it. The paper also presents an overview of automated transformations involved in the methodology.

Orriens et al in [17] introduce a model driven approach for development and management of dynamic web service composition. It provides a phased approach starting from abstract definition, scheduling, construction and execution. First the required information is modeled as classes in Information Model. OCL is used to specify business rules. In scheduling phase, the abstract composition is made more concrete by correlating messages and structuring activities. These are made unambiguous in Construction phase by associating activities with roles and selecting concrete services. . Finally, in execution phase executable format of structured service composition is derived.

B´ezivin et al in [19] talks about MDA approach for development of Web Services and their compositions. Business processes are shown by activity diagrams and static structures by UML class diagrams. Authors have shown mapping from UML to BPEL4WS, WSDL and Java Platform.

In [4], Xu et. al use aspects for web service composition using UML. Web service composition is modeled using UML. . The static part of the composition i.e., interfaces with their operations' signatures and message types are modeled using class diagrams. The standard UML elements of activity diagram are extended to show the control flow and data flow of the composition. Aspects are used to separate crosscutting concerns and they provide 'pluggable' behaviors. These are also modeled using UML. UML profile is used to model aspects. This UML profile shows aspects, pointcuts, web service interfaces and binding relationships. This paper is a first effort towards modeling of web service compositions using aspects. The paper uses UML profile, addresses control and data flow and provides modeling at meta-model level. However, it does not provide any tool support and does not address web service composition patterns.

John and Gerald [20] use UML profile to specify semantic web service compositions using model-driven approach. In this approach, the software developer specifies the structure of the composite web service using UML class diagrams and the compositions are shown using UML Activity diagrams. The pre- and post-conditions are specified using OCL. UML specifications are transformed to OWL-S specifications. A grounding tool is used to specify input/output messages and operations in a WSDL file from OWL-S specifications. This approach is supported by a tool and is well presented with a case-study. The approach lacks any feedback mechanism to developer and use web service composition patterns, which address common composition behaviors.

3. Comparison Of Approaches

In our study, we have discussed various approaches that employ UML as modeling tool in Web Service composition. The classification proposed is useful in analyzing similarities and relations between different approaches.

Based on our discussion of the approaches, a comparison table can now be established. The attributes used are explained below:

a- *Artifacts Used*
This attribute tells about the UML artifacts that are employed in modeling the composition.

b- *Categorization*
It tells about the classification under which the approach falls.

c- *Use of Profile*
It tells that in order to cater various requirements of the domain, has the author proposed/used any UML Profile.

d- *Feedback/Refinement*
Web Service Composition is not a trivial task. Different web services which are developed by different organizations/ developers and under different circumstances are combined to form a new composite web services. Many problems may arise in this process. This attribute is important as user feedback can help in finding and eliminating the conflicts in the composition of a new web service and provides control to developer over the composition of required composite service.

e- *Use of Ontology*
Work on developing semantic web is gaining momentum. Some approaches have used ontologies to provide semantics in developing composite web services.

f- *Addresses Service composition patterns:*
Various Patterns have been identified in SOA world. These patterns describe problems which occur again and again in an environment. This attribute specifies whether the particular approach caters service composition patterns in its model or not.

g- *Addresses Control conflicts*
This parameter shows whether the control conflicts that arise during composition of a web service are addressed at modeling level or not.

h- *Addresses Data Conflicts*
Data Conflicts can also be addressed at model level in order to avoid the errors that may come at later stage i.e. at execution level. It is better

to address such conflicts at model level since discovering and fixing the error at a later stage is expensive in terms of time and cost.

i- *Modeling at Meta-Model level:*
Meta-Model level is at a higher level of abstraction. Some approaches cater different domain specific modeling requirements at a meta-model level. This attribute shows that the modeling of composite of web services is addressed at meta-model level and hence semantics of the modeling are also catered.

j- *Automatable:*
Automation of an approach tells whether a tool can be developed for it so that minimum human interaction is needed. Some techniques are automatable but do not have tool support available yet.

k- *Tool Support*
This attribute represents maturity of the technique and serves as its proof of concept.

l- *Case Study*
It provides a practical example about the application of the approach. A case study strengthens the work as it highlights the practical nature of the work.

Based on these attributes, Table 1 summarizes the analysis of various approaches discussed above.

4. Conclusion

Web service composition has made a remarkable impact on the way systems interact and sharing information. However, this composition resulted in an increase in complexity, since the existing component services are developed under different environments and by different developers. Our study focused on the techniques that model web-service composition to address the issue of tackling these complexities at a design level. Efforts have been done to model the web-service compositions using domain specific languages and new modeling notations. Such modeling methods are hard to learn and use, and have a very limited tool support available. On the other hand, UML is an easy to understand modeling language that has a proper meta-model, extension mechanisms defined, and that caters designing issues from different point of views.

We have studied various UML based approaches for web service composition. These approaches have been classified based on the type of UML artifacts used. Based on our survey, we conclude that a good UML based modeling approach for Web Service Composition should cater structural and behavior aspects of the composition. The field, however, lacks a standard UML modeling profile. There is a need to define a modeling approach that covers all the aspects needed to provide a valid composite service.

References

[1] Castro, V., Marcos, E. and Sanz, M. L. Service Composition Modeling: A Case Study, *Seventh Mexican International Conference on Computer Science (ENC'06)*

[2] Skogan, D., Grønmo, R. and Solheim, I. Web Service Composition in UML, presented at *The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, California, USA., 2004.*

[3] Grønmo, R. and Solheim, I. Towards Modeling Web Service Composition in UML, presented at *The 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI-2004), Porto, Portugal, 2004.*

[4] Xu, Y., Tang, S., Xu, Tang, Z. Towards Aspect Oriented Web Service Composition with UML*, 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007).*

[5] Benatallah, B., Sheng, Q. Z. and Dumas, M. The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing 7(1): 40-48 (2003).*

[6] Benatallah, B., Dumas, M., Sheng, Q. Z. and Ngu, A. H. H. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services, *Proc. Int'l Conf. Data Eng. (ICDE)*, *IEEE Press, 2002, pp. 297-308.*

[7] Gardner, T. UML Modeling of Automated Business Processes with a Mapping to BPEL4WS, presented at *17th European Conference on Object-Oriented Programming (ECOOP), Darmstadt, Germany, 2003.*

[8] **T**höne, S., Depke, R. and Engels, G. Process-Oriented, Flexible Composition of Web Services with UML, presented at *Int. Workshop on Conceptual Modeling Approaches for e-Business: A Web Service Perspective (eCOMO 2002), Tampere, Finland, 2002.*

[9] Pathak, J., Basu, S., Lutz, R. and Honavar, V. MoSCoE: A Framework for Modeling Web Service Composition and Execution. In *IEEE 22nd Intl. Conference on Data Engineering Ph.D. Workshop*, page x143. IEEE CS Press, 2006.

[10] Pathak, J., Basu, S., Lutz, R. and Honavar, V.: Parallel Web Service Composition in MoSCoE: A Choreography-based Approach. In *4th IEEE European Conference onWeb Services (2006)*

[11] Timm, J. and Gannod, G. A Model-Driven Approach for Specifying Semantic Web Services. In *3rd Intl. Conference on Web Services*, *pages 313–320. IEEE press, 2005.*

[12] Barrett, R. and Pahl C. Semi-Automatic Distribution Pattern Modeling of Web Service Compositions using Semantics, In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*

[13] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D. *Web Services Architecture,* W3C Working Group Note 11, February, W3C Technical Reports and Publications, Available at http://www.w3.org/TR/ws-arch/.

[14] Amsden, J., Gardner, T., Griffin, C. and Iyenger, S. Draft UML 1.4 Profile for Automated Business Processes with a mapping to the BPEL 1.0. IBM alphaWorks (2003). Available at http://dwdemos.alphaworks.ibm.com/wstk/common/wstkdoc/services/demos/ uml2bpel/README.htm

[15] UML 2.0 Profile for WS-BPEL with Mapping to WS-BPEL, Available at http://elib.uni-stuttgart.de/opus/volltexte/2005/2487/index.html

[16] Dustdar, S. and Schreiner, W. A Survey on Web Services Composition. *International Journal on Web and Grid Services, 1(1):1–30, 2005.*

[17]. Orriens, B., Yang, J. and Papazoglou, M. P. *Model Driven Service Composition*, ICSOC, LNCS 2910, Springer-Verlag Berlin Heidelberg, pp.75–90.

[18] Grønmo, R. and Jaeger, M. Model-driven semantic web service composition. In *Proc. 12th Asia-Pacific Software Engineering Conference (APSEC 2005*), Taipei, Taiwan, 2005.

[19] B´ezivin, J., Hammoudi, S., Hammoudi, S., Lopes, D. and Jouault, F. Applying MDA Approach for Web Service Platform. *The 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC'04)*, *2004.*

[20] John, T.E.M, Gerald .C.G., Specifying Semantic Web Service Compositions using UML and OCL, *IEEE International Conference on Web Services (ICWS 2007)*

[21] Formal UML Specifications. The Unified Modeling Language (UML), Available at: http://www.uml.org. 2003

[22] UML 2.0 Superstructure Specification, Object Management Group, Framingham, Massachusetts, October 2004.

[23] Benatallah, B., Dijkman, R. M., Dumas, M. and Maamar, Z. *Service Composition: Concepts, Techniques, Tools and Trends*, Eds. Service-Oriented Software Engineering: Challenges and Practices, chapter 3, pages pp. 48-66. IDEA Group-2005.

[24] Web Services Flow Language (WSFL 1.0). Available at: http://www306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[25] XLANG. Web Services for Business Process Design. Available at: http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm, 2004.

[26] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I. and Weerawarana, S. Business Process Execution Language for Web Services, Version 1.*1*.Specifcation, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, 2003.

[27] Rao, J. and Su, X. A survey of automated web service composition methods. In: *Proc. of the 1st Int. Workshop on Semantic Web Services and Web Process Composition, SWSWPC2004, LNCS, San Diego, USA. (2004)*

[28] Hull, R and Su, J. Tools for Composite Web Services: A Short Overview. *SIGMOD Record*, *34(2):86–95, 2005.*

[29] Frankel, D. and Parodi, J. *Using Model-Driven Architecture to Develop Web Services*. Technical Report, IONA Technologies PLC, April 2002

Table 1 - Comparison Table Of UML based Modeling Approaches for WSC

Technique	Categorization	Artifact	Use of Profile	Feedback / refinement	Addresses Service composition patterns	Use of Ontology	Addresses Control Conflicts	Addresses Data Conflicts	Modeling at Meta-model level	Automatable	Tool Support	Case Study
Castro et.al	Behavior Based	Use Case, Activity	Yes	No	No	No	No	No	Yes	Yes	Yes	Medical Image Management
Grønmo & Solheim	Behavior Based	Activity	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	Gas explosion
Benatallah et.al	Behavior Based	State Chart	No	No	No	No	Yes	Yes	No	Yes	Yes	Travel Planning Service
Pathak et.al	Behavior Based	State Chart	No	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Online Doctor's Appointment
Timm & Gannod, Barrett & Pahl	Behavior Based	Activity	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	CongoBuy
Skogan et.al	Hybrid	Class, Activity	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	Gas explosion
John and Gerald	Hybrid	Class, Activity	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Find Cheaper Bool
Gardner, Amsden et.al.	Hybrid	Class, Activity, Package, Interaction	Yes	No	No	No	Yes	Yes	No	Yes	Yes	Loan Approval
Thöne et.al	Hybrid	Class, Activity	Yes	No	No	No	Yes	Yes	No	Yes	Yes	E-Business
Grønmo & Jaeger	Hybrid	Class, Activity	Yes	No	No	Yes	Yes	Yes	No	Yes	Partial	ExpressCongo Buy
Xu et. al	Hybrid	Class, Activity	Yes	No	No	No	Yes	No	Yes	Yes	No	Health Department Service
Bézivin et.al	Hybrid	Activity, Class	No	No	No	No	No	No	Yes	Yes	Yes	No
Orriens et.al	Hybrid	Activity, Class	Yes	No	Yes	No	No	Yes	yes	Yes	No	Travel Plan Flow