A Classification Framework for Approaches and Methodologies to make Web Services Compositions Reliable

Muhammad F. Kaleem

Technical University Hamburg-Harburg
m.kaleem@tuhh.de

Abstract: Individual web services can be composed together to form composite services representing business process workflows. The value of such workflows is directly influenced by the reliability of the composite services. There is considerable research concerned with reliability of web services compositions. There is, however, no clear definition of reliability that establishes its scope in the context of a composite service. We propose a definition of composite service reliability in this paper that takes into account different aspects affecting reliability of a composite service. We also put this definition to use as the basis of a framework that can be used for classification of approaches and associated methodologies for making composite services reliable. The framework can be used to identify which aspect of composite service reliability is addressed by a particular approach. We will also reference some selected methodologies to classify them according to the aspect of reliability they address. A definition of composite service reliability and its use to classify methodologies for composite service reliability as described in this paper will prove useful for comparing and evaluating methodologies for making composite services reliable, and will also have a bearing on the quality of service aspects of architectural styles and methodologies of software solutions based on web services compositions.

Introduction

Web services represent autonomous services with clear service definitions. Interaction with web services is possible through their service definitions, which can be made available in WSDL [1]. Individual service definitions may represent limited business functionality. However, it is possible to compose functionality offered by different individual services, likely from different service providers, into a composite service representing a complete business process. A number of standards exist for composition of autonomous, individual web services into a composite service representing a workflow [2-5].

Given that a composite service will most likely comprise of a number of autonomous web services, the reliability of the workflow represented by the composite service becomes significant. Reliability of composite services is the subject of much research. There is, however, no clear definition of reliability that establishes its scope in the context of a composite service. Also, there is diversity of research in

this area, but different research approaches for ensuring reliability of composite services cater to a definition of reliability peculiar to that approach.
In this paper we present a definition that identifies main areas of composite service reliability, and lists the reliability aspects related to these areas. We will use this definition as the basis of a framework that can be used to classify different approaches that may be used to address the reliability aspects identified in the definition. We will also reference some selected methodologies complementing these approaches, gleaned from research and industry proposals, and classify them using the framework according to the aspect of composite service they address.

Definition and its use as a classification framework

We present a definition of composite service reliability that is split into two parts, so as to identify two main areas of composite service reliability. We also list the reliability aspects related to these areas. We will then provide more details about these aspects in the next sections, when we use the definition as a framework for classification of methodologies for composite service reliability. To this effect, we will first describe the approaches that may be taken to address each aspect of reliability, and then reference some selected solutions that provide methodologies complementing these approaches. We reference solutions from research and industry, and in this way classify the methodology according to the particular aspect of composite service reliability it addresses. With the help of the definition and using it as a basis of a classification framework, it is therefore possible to classify a solution according to the aspect of composite service reliability it addresses.

The two main areas of composite service reliability, and the reliability aspects related to these, are:

1. Reliability of composition
   – *that the composite service is correctly specified*

   This aspect requires that the notation for specifying a web services composition is correct, and that the composite service specification, when put into execution, translates to correct process flow which it represents.

   - *Approaches*

     A number of standards exist for composition of web services, as we mentioned previously. These standards describe how individual web services can be composed together to form an executable business process. For the composite service to reliably represent a business process, it is important that the composite service specification is correct. However, correct notation for specifying the composite service alone is not sufficient to guarantee reliability. It is also important that the specification representing the web services composition translates into an error-free process flow when the composite service executes. It is important to address both of these points.

   - *Possible Solutions*

     Specifying a web services composition according to a composition standard can be a complex task. This can, however, be facilitated by tools typically provided by implementers of web services composition standards. An additional advantage

such tools can provide is checking whether the specification notation is correct, and thereby identify potential sources of error. As an example, we may mention the commercial implementation [6] of the web services composition standard [2], which provides graphical tools to visually compose a composite service definition.

Once the composite service has been specified, it can be checked whether the specification will translate into correct process flow on execution. A possible solution for this activity is described in [7], which uses model checking techniques to verify the workflow specification created according to a particular web services composition standard. Another technique for verification of composite e-services is described in [8]. Even though the work described in [8] is not directly related to web services compositions, the methodology described can be useful for further work into composite service reliability.

– *that the composite service consists of functional interface definitions*

This aspect deals with the requirement that all individual web services making up the composite service are available and functional according to their interface definitions, which may have been obtained from an online registry or resource.

- *Approaches*

A composite service may be composed dynamically, with requisite web services being added as they are needed. The information about a service's interface may come from a registry, or any other online resource. However, it is possible that the requisite service itself does not exist any more. This aspect of reliability is also relevant if we take into account the fact that a composite service may exist for a long period of time, during which time a service may go offline, or the service provider may not offer it anymore. The composite service should be able to handle such a situation. The approach should be to address this aspect at composition time, so that the composite service consists of functional individual services, and errors during the execution of the composite service due to unavailability of individual services are prevented. However, the problem represented by this aspect is relevant during composite service execution as well, when an individual service may go down (due to software or hardware failure, for example), and an approach for addressing this aspect of reliability has to take this factor into account as well.

- *Possible Solutions*

Implementers of web services composition standards could provide tools for checking whether all individual web services making up the composite service are functional. Similarly, the implementations of the standards could provide a test framework that checks the availability of the web services implementation at composition time. The advantage of doing this at composition time is to avoid potential sources of error during composite service execution. However, even during composite service execution, the breakdown of an individual service will present a problem. This can be dealt with methodologies explained later when we go over the aspects related to execution of the composite service.

– *that the composite service specification is conformant to the specification of individual web services*

This aspect requires conformance between the process flow requirements represented by the composite service and the constraints expected by the component web services.

- *Approaches*

A composite service could comprise of individual web services from different service providers, and each individual service may have its own business service policy that drives its interaction with its users. This implies that used individually, the service may be able to impose constraints on its usage. As part of a larger composition, the constraints expected by the individual services may conflict with the business service policy of the business process represented by the composite service. It is important to avoid this for the reliability of the web services composition. Similarly, but seen from a different viewpoint, individual web services should conform to the business policy represented by the composite service so that that this aspect of reliability is addressed.

- *Possible Solutions*

The web services composition standards can allow specifying constraints on the usage patterns of individual web services that are commensurate with their business service policies, so that there are no inconsistencies between the composite service and the individual services it is composed of. For example, the composition standard [2] has the notion of abstract and executable processes, where abstract processes may be used to capture behavioural aspects of services. It is also possible to address this aspect of composite service reliability at a level other than that of the composition standard. [9] suggests an experimental methodology that is relevant in this regard. However, similar methodologies will have to be refined and enhanced in scope before they can be applied to making composite services reliable. It is also possible to allow a web service to specify its policies and characteristics with respect to the interface it exposes to the outside world, so that these policies are taken into account during the web service composition process. We are working on a framework that allows a web service to specify its policies with respect to its participation in a composite service using the set of standards [10-12]. The framework also allows a web service composition provider and the web service provider to negotiate participation in the composite service through an enrolment process. The enrolment process also helps cover the reliability aspect mentioned in the previous section, related to functional interface definitions.

– *that the composite service can handle interface definition changes within individual services gracefully*

This reliability aspect deals with the ability of the composite service to handle interface changes within individual services without producing unexpected errors.

- *Approaches*

As was mentioned previously, a composite service may be in operation for a long period of time. Given the autonomous nature of service providers, it is possible that a service provider may change the interface of a service that is part of the composite service. This should not cause the breakdown of the business process represented by the composite web service, or lead to unexpected errors. Therefore ability to deal with such interface definition changes, called graceful handling in the definition, is important for the reliability of the composite service. Possible approaches could be extensions in the web services composition standards to allow for graceful handling of interface changes within individual services in the

composite service, or providing a layer of functionality on top of the composition layer, which takes care of interface changes.

- *Possible Solutions*

It is possible to build a layer of functionality on top of the composition standards implementation to address this issue, as described in [13]. This approach uses a conversational model of interaction between web services for graceful handling of interface changes in individual web services. Another solution could be facilitated by negotiated enrolment of the individual web service in the composite service (as mentioned in the previous section), where, for example, the web service may be bound by contract not to change its service interface.

2. Reliability of execution
   – *that the composite service execution is consistent with the business process flow it represents, and there are appropriate fault-handling mechanisms in place*

This aspect covers the reliability issues that are relevant when there is mutual interaction between a composite service (also within services of which the composite service is composed of) and external users and services as part of a business process flow scenario.

- *Approaches*

Fault-handling mechanisms can deal with errors that may occur during the execution of the composite service in an appropriate manner. The appropriate manner would depend on the kind of the error and its effect on the composite service execution. For a simple error, the fault-handling approach may just be to log the error, but for an error that can lead to data corruption, for example, a fault-handling mechanism may have to restore the state of the data to the one before the error, and may have to take compensatory actions to achieve this. A general approach to handle such requirements is to use transactional mechanisms. Furthermore, since interactions between participants in a web services composition are expected to represent long-running transactions, where application of traditional rollback mechanisms is not possible, compensatory actions to reverse the effect of an activity that has already completed play an important role in fault-handling for composite services.

- *Possible Solutions*

Due to the loosely-coupled interaction between web services, and the autonomy of service providers, requirements for transaction management in a web services environment are different from traditional transaction management requirements. Therefore traditional transactional mechanisms cannot be applied directly to web services. There are standards available [14, 15] that can be used to enable transactions over web services. These standards are the result of industry efforts. From among these, [14] is closely related to a web services composition standard [2], and therefore provides a ready model for using transactional behaviour to address reliability aspects during composite service execution. However, the standards for transactions over web services are quite new, and there is still some way to go for easy applicability of these standards to practical situations. An interesting comparison of these standards is presented in [16].

There are research proposals [17, 18] catering to this area of reliability as well. [17] proposes a framework for building transactional compositions of web

services. With respect to the definition of composite service reliability that we presented, the scope of this framework relates to the reliability of composite service execution. [18] seeks the same objective as [17], that being building reliable web services compositions on top of autonomous web services, however the approach it takes is different. The framework proposed by [18] consists of a multi-layered architecture and a transaction model for building reliable composite services, and in general covers a wider spectrum of reliability issues related to compositions of web services.

Using compensation to reverse the effect of completed activities has a bearing on reliability of execution, and hence the overall reliability of the composite service. Compensation is, however, a complex task, and requires coordination between all participants in a composite service, in the form of correct specification of compensatory activities in the composite service on part of the composite service provider, and the provision of compensatory activities on part of the web service provider.

– *that the messages flowing between web services are reliably delivered*

- *Approaches*

Reliable delivery of messages flowing between web services is an important factor that influences correct execution of the composite service, thereby having a bearing on the overall reliability of the composite service. The approach for reliable message delivery should be to ensure that messages from a sender should reach the intended recipient despite any software and hardware failures along the way.

- *Possible Solutions*

There have been a number of efforts into reliability of message delivery between web services. We will not list these here, except for a recent industry proposal in the form of a set of web services standards [19] that provide for reliable message delivery. Among these standards is [20], which specifies a model that provides the guarantee that messages sent by the initial sender will be delivered to the ultimate receiver.

Conclusion

In this paper we presented a definition of composite service reliability that modularises reliability concerns into two main areas and identifies different aspects affecting reliability related to these areas. This definition is incipient, and as part of our research, we plan a systematic evaluation of this definition, so as to substantiate and further refine it, as well as to verify and possibly enhance its scope. Similarly, we plan to evaluate the classification framework presented in this paper as well, so as to enhance it, and verify its usefulness. We also described some possible approaches to address these aspects of composite service reliability, and then referenced selected works that propose methodologies complementing the approaches. This was done to demonstrate how the definition may be used as a classification framework, and it was shown how a particular methodology could be classified according to the aspect of reliability it addresses.

We believe a comprehensive definition of composite service reliability, and the classification of approaches and methodologies according to the aspect of composite service reliability they address is important for further research into this subject. A definition and classification framework can help with the comparison and evaluation of different methodologies for composite service reliability, based on the aspect(s) of reliability they address. Given that there are at present different standards for web services composition, and different approaches for ensuring reliability of composite services, such comparison and evaluation is essential.

The selected methodologies that we presented deal with reliability of composite services. However, through the use of the definition and classification framework presented in this paper, it can be seen that these address particular aspects of composite service reliability. The definition and classification framework as presented in this paper should help establish the context in which research work on composite service reliability is performed.

Another important use of the definition of composite service reliability relates to the view of web services as components, and composite services as orchestrations of components. Since components and their composition touches upon the issue of re-use, a comprehensive definition of composite service reliability can help identify issues that could affect reliability when reusing web services components.

References

1. *Web Services Description language (WSDL).* http://www.w3.org/TR/wsdl
2. *Business Process Execution Language for Web Services, version 1.1*. http://dev2dev.bea.com/technologies/webservices/BPEL4WS.jsp
3. *Business Process Modelling Language*. http://www.bpmi.org/bpml-spec.esp
4. *Web Service Choreography Interface (WSCI).* http://www.w3.org/TR/wsci/
5. *XML Process Definition Language*. http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf
6. Collaxa Inc. *Collaxa BPEL Orchestration Server*. http://www.collaxa.com
7. Shin Nakajima. *Model-Checking Verification for Reliable Web Service*. OOPSLA 2002 Workshop on Object-Oriented Web Services. 2002.
8. Xiang Fu, Tevfik Bultan, and Jianwen Su. *Formal Verification of E-Services and Workflows*. Workshop on "Web Services, e-Business, and the Semantic Web (WES): Foundations, Models, Architecture, Engineering and Applications". 2002.
9. Giacomo Piccinelli, Anthony Finkelstein, and Christian Nentwich. *Web Services Need Consistency*. OOPSLA 2002 Workshop on Object-Oriented Web Services. 2002.
10. *Web Services Policy Framework (WSPolicy), version 1.1*. http://www-106.ibm.com/developerworks/library/ws-polfram/
11. *Web Services Policy Assertions Language (WS-PolicyAssertions), version 1.1*. http://www-106.ibm.com/developerworks/library/ws-polas/
12. *Web Services Policy Attachment (WSPolicyAttachment), version 1.1*. http://www-106.ibm.com/developerworks/library/ws-polatt/

13. Santhosh Kumaran and Prabir Nandi. *Conversation Support for Web Services*. Accessed on: 15 June 2003. http://www-106.ibm.com/developerworks/webservices/library/ws-conver/
14. *Web Services Transaction (WS-Transaction).* http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/
15. *OASIS Business Transaction Protocol (BTP).* http:/ www.oasis-open.org/committees/business-transactions/
16. Roger Sessions. *Shootout At The Transaction Corral; BTP Versus WS-T.* Accessed on: 15 June 2003. http://www.objectwatch.com/issue_41.htm
17. Thomas Mikalsen, Stefan Tai, and Isabelle Rouvello. *Transactional Attitudes:Reliable Composition of Autonomous Web Services*. International Conference on Dependable Systems and Networks (DSN 2002). 2002.
18. Paulo F. Pires, Marta Mattoso, and Mário Benevides. *Building Reliable Web Services Compositions*. NET.Object Days Conference (WS-RDS'02). 2002.
19. IBM and Microsoft. *Reliable Message Delivery in a Web Services World: A Proposed Architecture and Roadmap*. Accessed on: 15 June 2003. ftp://www6.software.ibm.com/software/developer/library/ws-rm-exec-summary.pdf
20. *Web Services Reliable Messaging Protocol (WS-ReliableMessaging).* http://dev2dev.bea.com/technologies/webservices/ws-reliablemessaging.jsp