壹種網路服務對話
A Conversation of Web Services

楊慶祥
C S Yang

崑山科技大學電子工程系
Department of Electronic Engineering, Kun Shan University of Technology

摘要

XML 網路服務近年來快速地發展，而且有潛力凌駕舊式的中間軟體及舊有的服務。在其所遭遇的各式各樣問題中，最重要的就是要求網路服務能交互操作，且有能力穿越各種邊界作對話。本文利用 WS-Coordination、WS-AtomicTransaction、WS-BusinessActivity 及 WS-I profile 協定，提出一個對話的例子。

關鍵字：XML, SOAP, WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, WS-I.

Abstract

XML web services have rapidly grown in recent years, and have the potential to override old middle-wares and services in the future. Among all problems they encountered, the most important one is to make them interoperable and able to do conversations across all kinds of boundaries. This article presents one example of conversation utilizing the WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, and WS-I profile protocols.

Keywords :XML, SOAP, WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, WS-I

1. Introduction

There are lots of services in the world with different languages, tools, platforms, etc. One kind of services has emerged with the potential of being independent of the platforms, languages, and tools. And they are the web services.

Since the Internet was developed, the web services have appeared. The early web services depended on the middle-wares and were heavily relied on the platforms, languages, and tools they used.

New generations of web services utilize the techniques of XML[1], SOAP[2], and others, and are able to communicate across different languages, platforms, organizations, enterprises, and all kinds of services. This is called the interoperability of the web services.

XML, like HTML, is a potentially promising language for interoperability. In addition to the newly developed schema XSD[3], XML can have a more flexible format of contents than HTML can have. And the SOAP is an XML content with its own schema.

XML web services take advantage of the flexibility of SOAP. By sending and receiving the SOAP messages, they
can communicate across all kinds of boundaries.

On the other hand, interoperability still has many problems to be overcome. Developers of web services are still struggling their ways to solve the problems of interoperability.

First, the web services need something to be used to describe their functionality. WSDL[4], among others, was proposed for this purpose. Its functions limit the usages of the web services. For example, it cannot be used to do co-occurrence or concurrent works, which most businesses and enterprises do frequently. One solution is to use XPATH[5] to replace some functions of middle-wares. More ambitious solutions are BPEL4WS[6], WSCL[7], BTP[8], etc. However, there are three protocols, WS-Coordination[9], WS-AtomicTransaction[10], and WS-BusinessActivity[11], seem to me to be in the good way. In addition, W3C proposed the WS_I[12] profile for the guidelines to be interoperable. The WS_I organization only proposed this profile for SOAP-HTTP communications. But the SOAP messages can be transported not only by the HTTP protocol, but also by other protocols such as TCP and SMTP.

2. The Conversation Models

2.1 A Coordination Service

The coordinator of a Coordination service is shown in Figure 1. The Activation service is required to initiate a CreateCoordinationContext, and the receiver responds a CreateCoordinationContextResponse. The Registration service registers Identifier and ProtocolIdentifier and the others according to the individual needs.

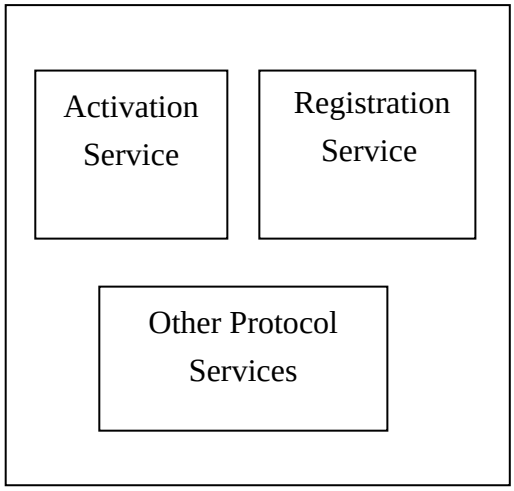

Figure 1. A coordinator representing the Coordination service.

2.2 An Atomic-Transaction Service

Figure 2 shows the coordinator of the Atomic-Transaction service. The Completion service demands two kinds of commits, one is the volatile commit, another is the durable commit.

An example use of WS-Policy assertion is

<wsp:SpecVersion
wsp:URI="http://schemas.xmlsoap.org/ws/2003/09/wsat"
wsp:Usage="wsp:Required"/>

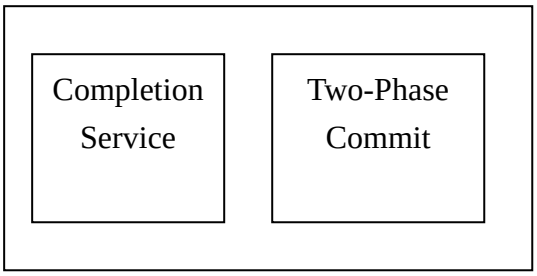

Figure 2. A coordinator representing the Atomic-Transaction service.

2.3 A Business-Activity Service

Figure 3. A coordinator representing the Business-Activity service.

The coordinator of a Business-Activity service is shown in Figure 3. The Business Agreement With Participant Completion protocol accepts:

Close: The participant indicates close.

Cancel: The participant indicates cancel.

Compensate: The participant indicates compensate.

Faulted: The participant acknowledges fault.

Exited: The participant acknowledges exit.

The Business Agreement With Coordinator Completion protocol accepts:

Complete: The sender indicates complete.

Both protocols accept:

GetStatus: The receiver returns the state message.

Status: The target service returns the state message.

The coordinator accepts:

Completed: The sender acknowledges complete.

Fault: The sender indicates fault.

Compensated: The sender accepts compensate.

Closed: The sender acknowledges close.

Canceled: The sender acknowledges cancel.

Exit: The sender indicates exit.

An example use of WS-Policy assertion is

<wsp:SpecVersion
wsp:URI="http://schemas.xmlsoap.org/ws/2004/01/wsba"
wsp:Usage="wsp:Required"/>

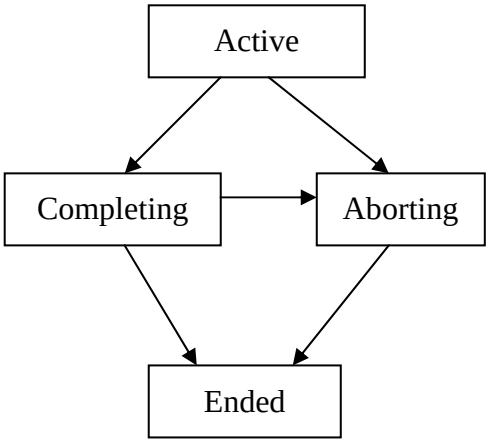

Figure 4. State diagram of the completion.

2.4 The completion state diagram

The completion is explained in figure 4. After activating the conversation, one may continue to complete or to abort. Also, under completing, one may choose to abort or to finish the conversation.

2.5 The 2-phase commit state diagram

The 2-phase commit is explained in figure 5. One can step through the processes of "activate", "preparing", and "prepared" with the choice of aborting. If one goes into "commit", he can only finish the conversation without aborting.

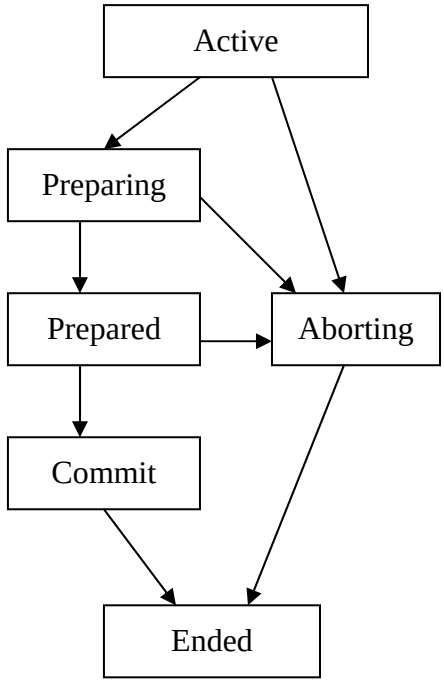

Figure 5. State diagram of the 2-phase commit.

2.6 Other objects

In addition to the objects presented here, many other objects constitute important parts of the web services. These are security objects, addressing objects, and so on. They are not the concerns of this article and are omitted here.

3. Implementation

A web service named WSS1 and a client of WSS1 have been developed using Microsoft Visual C++ .NET 7.1 in the Microsoft .net platform. The implemented protocols consist of WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, and WS-I. The transportation protocol is HTTP/1.1.

The conversation states consist of:

Activate: The initiator participates the activity by creating "CreateCoordinationContext" and registering the identity and the protocol identity. The receiver responds by emitting "CreateCoordinationContextResponse", and indicates "Fault" or "Activate".

Complete: The initiator with state "Activate" demands "Complete". The receiver responds "Fault" or "Complete".

Commit: The initiator with state "Complete" demands "Commit". The receiver responds "Fault" or "Commit".

Cancel: The initiator demands "Cancel". The receiver responds "Compensate" or "Close".

Compensate: The activity was canceled halfway, and the responder demands compensation.

Close: The activity was finished successfully.

Fault: Some mistakes happened and the initiator can retry or cancel.

The web view of the service is shown in figure 6. The client is a window form application and is shown in figure 7. The state diagram of this conversation is shown in figure 8.

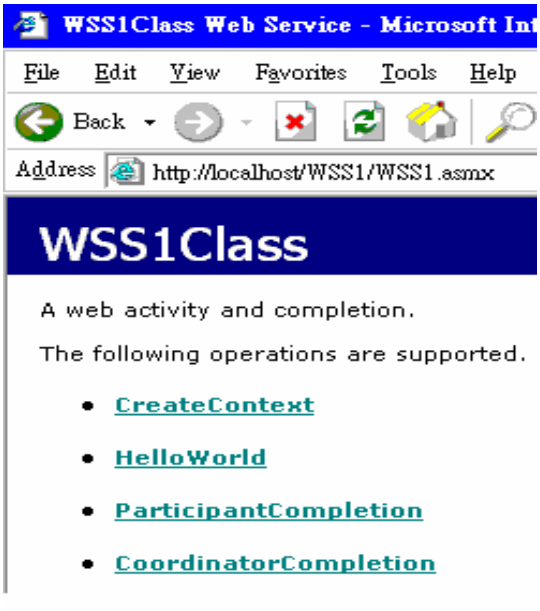

Figure 6. The WSS1 XML web service view.

Figure 7. The client of the WSS1 web service.

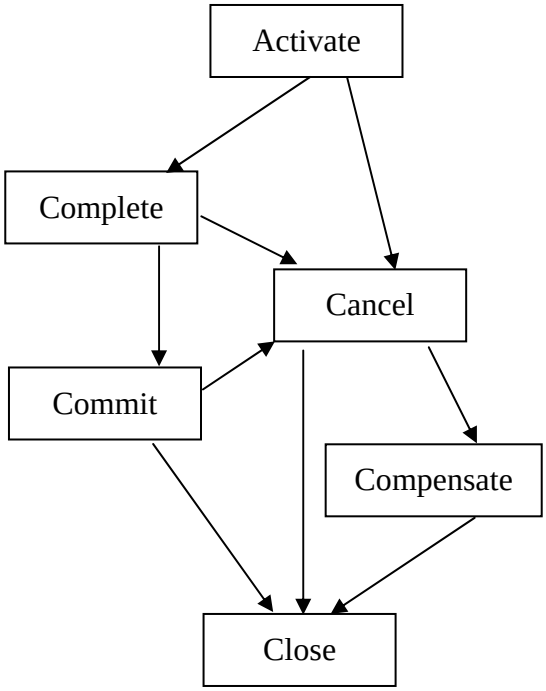

Figure 8. State diagram of WSS1.

4. Conclusions

(1) The protocols WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, and WS-I are able to cover most states of the conversation.

(2) The object representations of the SOAP messages have some ambiguities over complex messages and are not unique.

(3) The SOAP messages for some complex objects are incomplete, especially in the WSDL descriptions of the attributes of objects.

(4) SOAP messages with large data are vulnerable to transportation faults.

(5) Complex conversations need more states than WS-AtomicTransaction and WS-BusinessActivity can provide with.

(6) BPEL4WS is one kind of languages for web services and is lack of functionalities for web conversations.

(7) WSCL is also for web services. It is not appropriate for web conversations because web conversations need more than web services do. For example, the abstraction for the models of web conversations is still an active subject of researches.

(8) This article is a small step toward tackling the problems of web conversations. Although WS-AtomicTransaction and WS-BusinessActivity are not perfect for web conversations, they may be the step stone for a finer web conversation. Equipped with the power of the object-oriented SOAP and the flexibility of XML, this article has actually done more than WS-AtomicTransaction and WS-BusinessActivity said.

5. References

1. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", 06 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006.
2. W3C note, "SOAP: Simple Object Access Protocol", 08 May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508.
3. H. Thompson, et al, "XML Schema Part 1: Structures", May 2001,

http://www.w3.org/TR/xmlschema-1/.
P. Biron, et al, "XML Schema Part 2: Datatypes", May 2001, http://www.w3.org/TR/xmlschema-1/.

4. E. Christensen, et al, "Web Services Description Language (WSDL) 1.1", March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315/.
5. W3C recommendation 16 1999, "XML Path Language (XPah) 1.0", http://www.microsoft.com/fwlink/?LinkId=4873.
6. Business Process Execution Language for Web Services, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel1/.
7. Web Services Conversation Language, http://www.w3.org/TR/wscl10.
8. Business Transaction Protocol, http://www.oasis-open.org/commitees/business-transactions/.
9. WS-Coordination, http://msdn.microsoft.com/ws/2003/09/wscoor/.
10. WS-AtomicTransaction, http://msdn.microsoft.com/ws/2003/09/wsat/.
11. WS-BusinessActivity, http://msdn.microsoft.com/ws/2004/01/wsba/.
12. WS-I, http://www.ws-i.org/.