

# Realizing Interoperable Infrastructure for Electronic Business Systems and Local Government Systems

Kazunori Iwasa<sup>1</sup>, Masahiko Narita<sup>2</sup>, Makiko Shimamura<sup>1</sup>

<sup>1</sup>Strategy and Planning Division, Software Unit, Fujitsu Limited  
3-9-18 Shinyokohama, Kohoku-ku, Yokohama, 222-0033, Japan  
E-mail: {kiwasa, maki.shimamura}@jp.fujitsu.com

<sup>2</sup>Department of Information Systems Architecture  
Advanced Institute of Industrial Technology  
1-10-40 Higashi Ohi, Shinagawa-ku, Tokyo, 140-0011, Japan  
E-mail: narita-masahiko@aait.ac.jp

**Abstract.** Web Services and related open standard technologies are key technologies for system infrastructure in industry systems and local government systems. There are many projects that have adopted these technologies in the local government systems and also in B2B systems in various industries. We found that one of the key requirements to promote these standards in industries is pull messaging for client/server systems, since many small and medium enterprises can't afford server systems. To achieve this, we have standardized Pull messaging specification as a part of ebXML (Electronic Business using eXtensible Markup Language) Messaging Services Specification Version3.0, with inputs of industries requirements. We also have developed Conformance test tool and additional test assertions for ebMS3.0 to promote interoperable implementation for ebMS3.0, and executed interoperability test. In this experience, we have found interoperability issues in open source. We will investigate this issue further, and continue giving feedbacks to implementations and prototype. We believe this experience regarding Pull messaging technology will be valuable for local government systems in the future, when it is going to expand its infrastructure to connect with medium and small entities.

---

*Please use the following format when citing this chapter:*

Iwasa, K., Narita, M., Shimamura, M., 2008, in IFIP International Federation for Information Processing, Volume 286: Towards Sustainable Society on Ubiquitous Networks, eds. Oya, M., Uda, R., Yasunobu, C., (Boston: Springer), pp. 341–352.

## 1 Introduction

Web Service and relating open standard technologies are going to be one of the most important key technology for system infrastructures recently, because of 1) the openness of the key technologies, e.g., XML, Internet, SOAP and other Web Services related Specifications, 2) increasing a number of products and services that support Web Services and relating technologies, and 3) user demands to openness, flexibility of business data format, system architecture, and trading partners. Many industries are going to adopt Web Services technologies and its relating technologies in their system infrastructure when they standardize the industry standard of electronic business trading among enterprises in the industry. It is also adopted in the public service system infrastructure of local government in Japan. For this reason, interoperability of these infrastructures is critical to develop a system, since such systems should be able to communicate with a lot of other different systems, developed by various vendors. This paper describes concrete example of Web Services adoption in industries and local government in Japan, issues and lessons learned from standardization and interoperability verification for a standard messaging protocol, and the future activities to promote these technologies in industries and local government further.

## 2 Web Services in the Local Government Systems

There is a Japanese government project to adopt Web Services in the local government systems. Ministry of Internal Affairs and Communications, and the Association for Promotion of Public Local Information and Communication jointly sponsor this project. This project was initiated in 2005 to develop a Public Local Information Platform that will be the base of the nationwide local government information systems.

### 2.1 Overview of the Public Local Information Platform

The Public Local Information Platform is a set of specification to define common platform to provide integrated/communicated local information services from local government and/or private companies. Traditionally, Japanese local government has different systems for each department: electronic application, personnel and payment, pension, public welfare, tax, and others. It has been common to exchange data manually or by paper between different departments. To enable data exchange and operational interaction between systems in a different department, they are going to connect each application on a common system communication platform that adopt Web Services and relating open standard technologies. This system enables to share services and operations. For instance, when you move, it is required to submit various documents to the local government. It includes: Notification of change of address, address change application for the National Health Insurance and the National Pension Plan, application for changing location of light automobile, application for child-

care allowance, taxation certificate, and others. It is possible to provide one-stop service solution if each system that supports the above application: the basic resident register system, light automotive tax system, child-care allowance system, nursing care and public welfare system, electronic application system and others are communicated on the local information platform.

## **2.2 Goal of the Local Information Platform and its Requirements**

The goal of the Local Information Platform includes: 1) increasing convenience of government services by providing communicated services within a local government, inter-local government, and between local government and private sector, and creating a new business opportunity with this change, 2) decreasing a system cost by adopting an open standard technologies like Web Services, which enables more choices of software/service provider, and 3) reusing the existing local government systems. Therefore the standardization of the services, a platform that is adopting open standard technologies, and interoperability is the requirements for this platform.

## **2.3 Web Services as a Platform of Local Government Systems**

The Local Information Platform specification is going to define common vocabulary, standard business documents and business process to cover wide range of local government services including: Basic Resident Register, National Health Insurance, Child-care Allowance, Seal Registration, Property Tax, Resident Tax, Corporate Tax, National Pension Plan, Elder Care, Care Insurance, School Attendance, Census Registration, Personnel and Compensation, Document Management, and others. [4]

Local Information Platform also defines to use standardized Web Services technologies, since the system should be able to communicate with different systems from different vendors. For instances, it defines to use a standard communication protocols including reliable messaging and security, since those functions - guaranteed delivery, duplicate elimination message ordering and message security are critical to develop real business systems. It also defines specification for business process execution and management, authentication for service interaction, and system monitoring.

The Local Information Platform is currently for server systems for large entities. And it may be a future requirement to support smaller entities that can't afford server systems. In that case, client/server systems with pull messaging, that client asked a message to receive in the server, will be required in the future.

### 3 Web Services in the Industry

In contrast with the previous section, this section describes how the Web Services and B2B messaging standards are used in the vertical industry. It also describes how we have developed a standard specification to meet industry requirements.

#### 3.1 Industry Adoption for B2B Standard Messaging Protocol

- Retail Industry: Japanese retail industry has defined their standard protocol - "Japan Chain-stores Association (JCA) Protocol". And it is widely used in the industry from 1980s. It is pull messaging that client asks for a message to receive. Adoption rate between retail stores and wholesale/manufacture is more than 90%. DSRI (the Distribution Systems Research Institute) has defined Pull Messaging extension for "SOAP-RPC" to migrate it to Web Services technologies. They have executed POC in 2004. Japanese retail industry also decided to adopt ebMS2.0 for standard messaging protocol in the industry recently. [1]
- Manufacturing Industry: Japan Electronics and Information Technology Industries Association (JEITA) ECALGA project has standardized an industry standard specification for parts procurement system from 2003. It is adopting ebMS2.0 and many systems are in-services. JEITA also defined messaging model for client / server model to develop JEITA client recently as described in the Fig. 3.1.

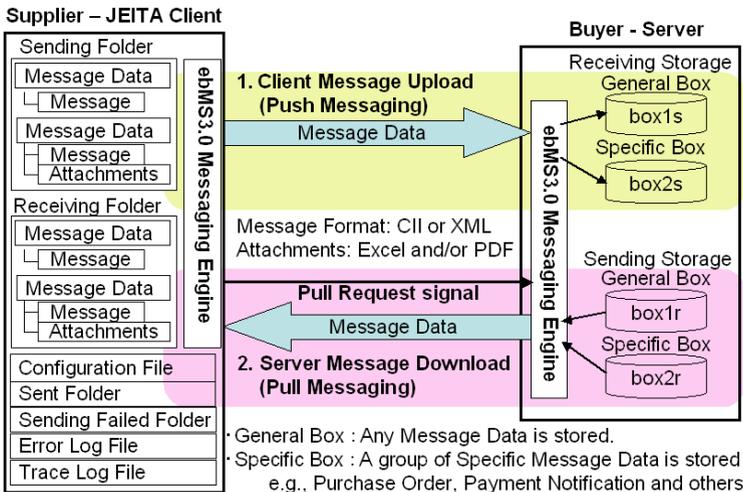


Fig. 3.1. JEITA Client and its messaging model \*

\* The fig. 3.1 is translated to English and merged from two figures that were originally created by JEITA member in Japanese [6].

- Common XML/EDI Practice Promotion Council (COXEC) is promoting EDI services with ASP model. They have defined "Common XML/EDI Framework" in December 2005.

### **3.2 Challenge to Increase B2B Adoption Rate in SMEs**

The above three industries are promoting the industry standard specification. However one of the big obstacles for them to increase adoption rate of B2B messaging standard is the low adoption rate of the B2B standard in the Small and Medium Enterprises (SMEs). There was ISO standard for B2B messaging protocol i.e., ebXML Message Services 2.0. However it was standardized mainly for server-to-server messaging. It requires the receiving PC stand by to receive an incoming message. There was a big demand of B2B messaging standard for client/server model to promote B2B standard in SMEs.

### **3.3 Requirements to Promote B2B in SMEs**

There are some requirements to promote B2B messaging standard in SMEs:

- The system should be easy to adopt. Most SMEs are not adopting real time processing. Currently, they are not required to process data in real time. They are not required to connect the Internet 24X7. They need light-weight system to allow them exchange data on the Internet easily.
  - A single system should be able to use for all trading partner. Although legacy Web-EDI is affordable, it requires to use different Web EDI systems for each trading partner. Sometimes it requires to pay multiple system fee or connection fee for multiple Web-EDI systems. It requires to input data manually. It is expensive in terms of human resources.
  - The system should be easy to use. SMEs have no enough resources for system maintenances or operation. It should be easy to exchange messages like FAX.
  - Messaging data in client system should be reused in other systems. Legacy Web EDI systems don't allow to reuse data, since all data is stored in server.
- [3]

### **3.4 Resolution: Standardizing Pull Messaging Protocol for Client/Server System**

To meet these requirements, we have standardized Pull messaging specification for Client-Server messaging model. The figure 3.2 shows how the Pull messaging works.

In this figure, the Responding MSH on the left side is going to send a message to the Initiating MSH on the right side. However Responding MSH doesn't initiate a message connection, but put the message in the local storage. The Initiating MSH will send a Pull Signal Message regularly to get a message. If there is a message for the Initiating MSH on the Responding MSH, then the message is delivered to the Initiating MSH on the response of the underlying protocol e.g., HTTP Response, if the underlying protocol is HTTP. This model is applicable for client / server model. The Initiating MSH is the client in this case. This is useful when the client has no global IP address, or client can't accept an incoming connection for the restriction of the firewall.

It realizes affordable adoption of B2B messaging infrastructure and easy maintenance. The Pull messaging specification was submitted to OASIS ebXML Messaging Services TC, and it was standardized as a part of ebXML Messaging Services specification Version 3.0 (ebMS3.0). This feature is one of the major enhancements of ebMS3.0 from ebMS2.0, since ebMS2.0 supports server-to-server messaging only. Pull messaging is legacy EDI friendly and it doesn't require changing the system architecture, although it is not real time messaging. The client system doesn't have to wake up 24X7. It doesn't require global IP address costing you a lot. Therefore Pull messaging meets requirements from SMEs and it

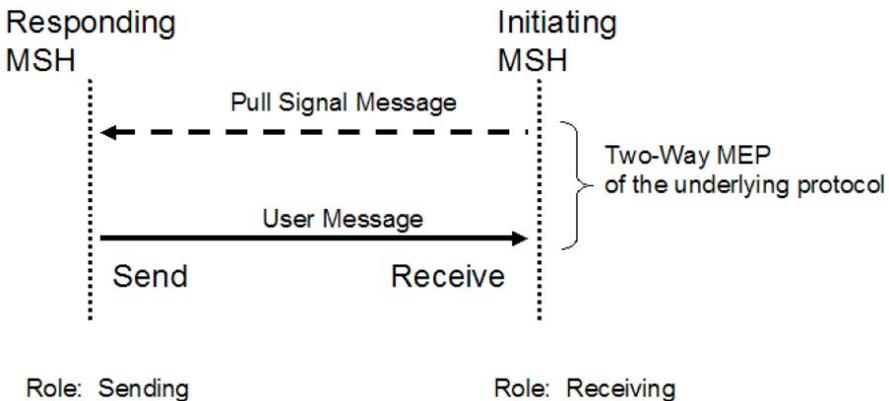


Fig. 3.2. Pull Messaging Model [7]

realizes affordable adoption cost and easy maintenance for B2B system. To standardize the Pull messaging specification, we also have gotten requirements from JEITA, COXEC, and other industries. The requirements includes:

- A simple authentication for client. The ebMS2.0 requires certificate for authentication. But the bar is too high for some SMEs. Therefore ebMS3.0 added simpler authentication - ID and Password authentication for SMEs.
- Prioritized / Categorized messaging. This feature let client choose a category of messages to receive.
- Reliable messaging. It requires resending a lost message and recovering after system down. It should not be depend on the application.

There requirements are submitted to OASIS ebXML Messaging Services TC, and standardized as a part of ebMS3.0 specification. With this achievement, JEITA and COXEC have adopted the ebMS3.0 with Pull Messaging for their client / server messaging model, as described in the Figure 3.1.

## **4 Verification and Promotion of the Specification**

In the previous two sections, it was described some examples of Web Services in the local government systems and some vertical industries. The common requirements to the technologies they have chosen are: 1) it should be a standard technology, and 2) assurance of interoperability of implementation, since a system needs to communicate with other systems developed by other vendors.

Currently, local government system is not targeted for small and medium organizations that can't maintain server. On the other hand, vertical industries are trying to promote electronic trading to SMEs, since many SMEs have not yet adopted electronic trading. In the previous section, it was described that Pull messaging standard specification was standardized as a part of OASIS ebMS3.0 specification to meet industry requirement that wish to promote B2B in SMEs. However standardizing a specification is not the end of resolution, since it is more important to make various implementations interoperable when they have implemented the standard specification. This section describes how we are ensuring conformance and interoperability of the standard specification we have standardized.

### **4.1 Conformance Test Tool for Web Services is Developed**

There is a conformance test tool for Web Services technologies that we have developed to promote interoperable Web Services implementation. We have tested reliable messaging specification and implementation with this test tool and executed interoperability test for local government system application before.

### **4.2 Supporting ebMS3.0 with the Conformance Test tool**

To help implementers of ebMS3.0 test their implementations, and promote interoperable implementation, we have developed ebMS3.0 test assertions for the confor

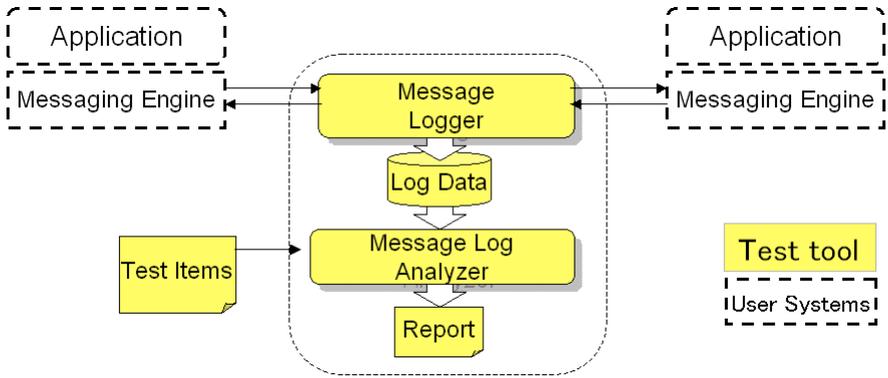


Fig. 4.1. Architecture of Conformance Test Tool

mance test tool. It supports all elements and attributes of the ebMS3.0 specification with 111 test assertions. The architecture of this conformance test tool is as follows:

### 4.3 Interoperability Verification for ebMS3.0

We have executed interoperability test among three implementations on March 2008. Two of them are clients and one of them is a server. Each implementation have been developed as follows:

Table 4.1. Implementation overview

	Client A	Client B	Server C
Java	JDK1.5.0_14	JDK1.5.0_11	JDK1.6.0.14
HTTP	HTTP1.0	HTTP1.0	HTTP1.1
SOAP engine*	Apache Axis1.3	Apache Axis1.3	Apache Axis2

\* The SOAP engine that each implementation adopted for generating, sending and receiving SOAP messages.

Please note each of these three implementations has adopted open source implementation -apache Axis, but the version is not same between server implementation and clients.

For this interoperability test, we have defined the following two major test scenarios. These test scenarios are very simple but include typical client server messaging scenarios that will be used widely in various industries.

**Table 4.2.** Test Scenarios

Test cases	Test description
SC1: Client sends a message to Server - Push messaging	
SC1-1	One-way Reliable Messaging with one payload
SC1-2	One-way Reliable Messaging with three payloads
SC2: Client receive a message from Server - Pull messaging	
SC2-1	One-way Pull Reliable Messaging with no message
SC2-2	One-way Pull Reliable Messaging with one payload
SC2-3	One-way Pull Reliable Messaging with three payloads

#### 4.4 The Current Status of the Test Results

Test Results for Client A and Server C is follows:

The UserMessage in the table includes all elements and attributes under the ebMS3.0 UserMessage element that includes business message.

The SignalMessage is the table includes all elements and attributes under the

**Table 4.3.** Test Results for Client A and Server C

Test category	Test Items	SC1-X				SC2-X			
		P*	F*	W*	N*	P*	F*	W*	N*
Messaging	6	6	0	0	0	6	0	0	0
UserMessage	69	56	0	0	13	0	0	0	69
SignalMessage	36	0	0	0	36	14	0	0	22

\* P:Passed, F:Failed, W:Warning, N:Not tested

ebMS3.0 SignalMessage element that include signal message like PullRequest - indication to pulling a message.

Messaging in the table is indicate the Messaging element in ebMS3.0 that is the most outer element of the ebMS3.0 message.

Test Results for Client B and Server C is follows:

**Table 4.4.** Test Results for Client B and Server C

Test category	Test Items	SC1-X				SC2-X			
		P*	F*	W*	N*	P*	F*	W*	N*
Messaging	6	6	0	0	0	6	0	0	0
UserMessage	69	48	0	0	21	0	0	0	69
SignalMessage	36	0	0	0	36	14	0	0	22

\* P:Passed, F:Failed, W:Warning, N:Not tested

Please note that the above test results showed a portion of tests we have successfully tested. There are some interoperable issues for this test. For instance, it was failed to complete sending SC1-x messages, since there was error occurred to save a received message in the server. The reason may be the incompatibility between Axis1.3 and Axis2, but need further investigation. However the message sent was captured by the testing tool and analyzed. The above test results showed the result of the analyzed message, although it was not saved in the server correctly.

## 4.5 Observation of the Test Results

From the test result above, we found that format of all message tested were correct. In detail, there is no test item that was failed or warned in Table 4.3 and Table 4.4. There are six test items for "Messaging" test category in Table 4.3. And it showed us that the tested message have passed all six test items. For "UserMessage" Test categories, 56 out of 69 test items were successfully passed the test. There are 13 test items that were not tested, since the tested message is not including such elements or attributes. Thus, about 65% of the test items were tested in this test, and all of them were successfully passed. We can conclude the test messages generated in this test were compliant with ebMS3.0 specification.

There are also interoperability issues in this test. The error was occurred during the process of Axis2 in the server. Axis2 tried to save attachments in the temporally file, but for some reason, it couldn't save the second MIME part and the after. One of the possible reasons is incompatibility of MIME implementation between Axis1.3 and Axis2. This issue should be further investigated. Currently it is the middle of migration from Axis1.x to Axis2, so both implementations exist in the market. We need to be careful about interoperability issues caused from such situation.

In summary, we have developed test assertions for the conformance test tool to test pull messaging in ebMS3.0. We tried to test existing implementations and prototype for ebMS3.0 with this testing tool. At this point, we have gained experience for incompatibility of the open source. We will continue the investigation of the issues and we will give feedbacks to implementation and prototype.

## 5. Conclusion and Future Activities

As described in section2, Local government is adopting Web Services technologies as a key infrastructure in the Local Information Platform. In industries, there are attempt to promote B2B standard technologies to Small and Medium Enterprises. One of the key requirements to promote B2B systems to SMEs is Pull messaging for Client/Server systems. It is one possibility that the Local Information Platform is also require Pull messaging to promote the platform more widely. For instance, it may expand the platform to connect with small systems, i.g., it may want to allow: 1) certified public tax accountant office to connect to the platform with a client system, 2)small health clinic to connect some systems on the platform in the future, and oth-

ers. In such cases, it may be possible to reuse the technologies the industry have developed, or get some hints from such experiences. We also have developed conformance test tool for Web Services and additional test assertions for ebMS3.0. We have tried interoperability test with existing ebMS3.0 implementations and prototype. We have gained knowledge about incompatibility of open source. The test tool also is applicable to other projects in the future, since it is publicly available.

It is important the industries validate and promote interoperable standard technology for pull messaging. We would like to contribute for this direction by promoting interoperability testing in Japan, Asia or in the world.

We believe this experience regarding Pull messaging technology will be valuable for local government systems in the future, when it is going to expand its infrastructure to connect with medium and small entities. In this case, the experiences and knowledge we get with this activity in industries can be applied in local government systems in the future.

## Acknowledgements

These activities are part of the “Digital Information Appliance Interoperability Technology Project (Development of Remote Management Platform for Information Appliances)” sponsored by NEDO (New Energy and Industrial Technology Development Organization) in Japan.

## References

1. Masahiko Narita, et al., Developing a Messaging Specification for EDI on Internet for SME and Its Standardization, IEICE, Vol.J89-D, No.10, pp.2237-2245, 2006
2. Masahiko Narita, et al., Verifying the Reliability of Web Services Interactions for the Robot Communication Platform, Tokyo Metropolitan University, Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol.12, No.1 pp77-84, January 2008
3. Kazunori Iwasa, et al., Promoting e-Business and Web Services Standards to SMEs in Japan, and Asia, OASIS, 2007
4. Masahiko Narita, et al., Verification for the Interoperability of Web Services for Municipality Systems, Advanced Institute of Industrial Technology, Bulletin of Advanced Institute of Industrial Technology, October 2007
5. Small and Medium Enterprise Agency, Ministry of Economy, Trade and Industry, White Paper on Small and Medium Enterprises in Japan, Japan Small Business Research Institute, 2007  
[http://www.chusho.meti.go.jp/pamflet/hakusyo/h19/download/2007hakusho\\_eng.pdf](http://www.chusho.meti.go.jp/pamflet/hakusyo/h19/download/2007hakusho_eng.pdf)
6. Kazunori Iwasa, et al., ebXML Messaging Services Ver3.0 Interoperability test specification, ECOM, 2008

7. Pete Winzel, et al., OASIS ebXML Messaging Services Version3.0: Part 1, Core Features, OASIS Standard, OASIS, 2007
8. Information Sciences Institute, University of Southern California, Internet Protocol, DARPA Internet Program Protocol Specification, IETF, RFC791, 1981
9. Information Sciences Institute, University of Southern California, Transmission Control Protocol, IETF, RFC793, 1981
10. R. Fielding, et al., Hypertext Transfer Protocol -- HTTP/1.1, IETF, RFC2616, 1999
11. Tim Bray, et al., Extensible Markup Language (XML) 1.0 (Fourth Edition) W3C Recommendation, W3C, 2006
12. Tim Bray, et al., Namespaces in XML 1.0 (Second Edition) W3C Recommendation, W3C, 2006
13. Henry S. Thompson, et al., XML Schema Part 1: Structures Second Edition W3C Recommendation, W3C, 2004
14. Don Box, et al., Simple Object Access Protocol (SOAP) 1.1 W3C Note, W3C, 2000
15. Nilo Mitra, et al., SOAP Version 1.2 Part 0: Primer (Second Edition) W3C Recommendation, W3C, 2007
16. Martin Gudgin, et al., SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) W3C Recommendation, W3C, 2007
17. Martin Gudgin, et al., SOAP Version 1.2 Part 2: Adjuncts (Second Edition) W3C Recommendation, W3C, 2007
18. Kazunori Iwasa, et al., WS-Reliability1.1 OASIS Standard, OASIS, 2004
19. Doug Davis, et al., WS-Reliable Messaging Version1.1 OASIS Standard incorporating Approved Errata, OASIS, 2008