A Business-to-Business Interoperability Testbed: An Overview

Boonserm (Serm) Kulvatunyou
Manufacturing Systems
Integration Division
National Institute of Standards &
Technology Gaithersburg, MD 20899-
8260, U.S.A
1-301-975-6775
serm@nist.gov

Nened Ivezic
Manufacturing Systems
Integration Division
National Institute of Standards &
Technology Gaithersburg, MD 20899-
8260, U.S.A
1-301-975-3536
nivezic@nist.gov

Monica Martin
Sun Microsystems, Inc.
1 Network Drive,
Burlington, MA 01803
1-208-585-5946
monica.martin@sun.com

Albert T. Jones
Manufacturing Systems
Integration Division
National Institute of Standards &
Technology Gaithersburg, MD 20899-
8260, U.S.A
1-301-975-3554
jonesa@cme.nist.gov

ABSTRACT

In this paper, we describe a business-to-business (B2B) testbed co-sponsored by the Open Applications Group, Inc. (OAGI) and the National Institute of Standard and Technology (NIST) to advance enterprise e-commerce standards. We describe the business and technical objectives and initial activities within the B2B Testbed. We summarize our initial lessons learned to form the requirements that drive the next generation testbed development. We also give an overview of a promising testing framework architecture in which to drive the testbed developments. We outline the future plans for the testbed development.

Categories and Subject Descriptors

E-Business Interoperability

General Terms

E-Business Management

Keywords

Keywords are your own designated keywords.

1. INTRODUCTION

B2B testing is instrumental for a successful implementation and deployment of inter-enterprise e-commerce standards. B2B testbeds are enabling instruments to (1) assure and assess quality of emerging standards, (2) promote standards adoption in the vendor products and by users, (3) validate conformance of the products to standards, and (4) validate interoperability between the vendor products and their implementations.

A software implementation is said to **conform** to a standard if it satisfies requirements set forth within the standard, as expressed by a certain collection of test cases and requirements. Software may conform to only a subset of requirements within the standard constituting conformance levels (or profiles) [8].

Interoperability, on the other hand, refers to two or more software systems that can function with each other to achieve a set of requirements. Software systems that conform to the same standard may not interoperate because of the differences in their respective environments and technical requirements. Typically, differing business needs drive technical requirements that can cross conformance levels and create interoperability challenges.

Related to conformance and interoperability is **compliance**. However, compliance is a less formal notion and it is typically used for stating that a product fulfills certain standard requirements as expressed by a standards organization-approved testing program. For example, a vendor may quote its product as ‘RosettaNet-compliant’ to indicate that it passed the RosettaNet-approved testing program.

Different standards organizations and industry consortia continue to invest in individual interoperability **demonstration, pilot, and proof-of-concept** efforts to showcase utility of these emerging standards and technologies [11], [4]. With the surge in the development of B2B standards and adopting technologies, the need for demonstration and testing to accelerate adoption of these standards has increased.

In the rest of the paper, we give an overview of the OAGI/NIST B2B Interoperability Testbed, its objectives, and the initial activities. We summarize lessons learned from these initial activities as a collection of requirements that drive the next generation testbed development. Then, we provide an overview of one promising testing architecture for developing the next generation testbed. We briefly reference other related international efforts in support of interoperability testing and we outline the future plans for the testbed development.

2. OAGI/NIST B2B Interoperability Testbed Overview

2.1 Objectives

The purpose of the OAGI/NIST B2B Interoperability Testbed is to establish an on-going effort to mobilize software vendors, users, standards organizations, and other stake-holding parties to enhance the capability for on-demand demonstration and testing of conformance and interoperability involving enterprise applications in a B2B setting.

For a software vendor, participation in the testbed offers an opportunity for marketplace recognition of the vendor B2B interoperability readiness. The testbed also offers a venue for vendors to maintain their product interoperability during minor software releases. Customers/users are increasingly aware of the costs and risks that immature and rapidly changing B2B standards bring to their companies. The testbed resources help its users to answer key questions such as 'What is an optimal way to leverage a new B2B standard?' and 'Does standard X adequately support my business requirements and business processes?' Using the testbed, standard developers receive feedback from vendors, partners, suppliers and customers on the standards issues.

Our initial experiences indicate that effective and efficient interoperability testing approaches and facilities can help significantly reduce cost and cycle time associated with B2B integration activities.

2.2 An Example of Supported Activity

STAR/XML is an effort within the Standards for Technology in Automotive Retail (STAR) consortium to define standard XML messages for dealer-to-OEM (Original Equipment Manufacturer) business transactions [13]. The STAR/XML initiative uses Electronic Business using eXtensible Markup Language Business Process Specification Schema (ebXML BPSS) specifications to represent scenarios of collaborations between OEMs and dealers/retailers [15].

To support the STAR/XML information exchanges, software vendors have included supports for BPSS in their products. Both the users (i.e., OEMs and retailers) and the vendors desire to assess the functionality and interoperability aspects of these BPSS implementations. Users are asking, "Did we define the collaborations properly?" Vendors are concerned "Do our implementations work when using real integration scenarios?" Both users and vendors pose the question "Will the different vendor products interoperate when using the standards?"

The OAGI/NIST B2B Interoperability Testbed offered a cost-effective environment to pilot an interoperability testing effort with STAR/XML users and software vendors to start addressing the above questions. The goal of the STAR/XML BPSS interoperability pilot was to explore the following scenarios: (1) Identify a real business collaboration between OEMs and retailers; (2) analyze and model the collaboration using a modeling tool; (3) define the XML documents and BPSS schema required by the collaboration; (4) execute the collaboration using products from different vendors; and (5) assess that the participating vendor products can interoperate.

We used sample deliverables from the STAR/XML consortium using Business Object Documents (BOD) [12] conforming to Open Application Group Integration Specifications (OAGIS), ebXML Messaging Service Handler (MSH), and BPSS as the B2B execution framework. The result of the interoperability pilot is summarized here:

- We identified the Process Parts Order business process definition and defined the business scenario using OAGIS BODs and ebXML BPSS to specify the characteristics of the collaboration.
- The business process was modeled using Mega International Inc.'s modeling tool, which also generated the specified BPSS instance [7].
- The BPSS instance was loaded into software products from Fujitsu and Sybase to validate that the schema, as generated by a third party modeling tool, is recognized and validated by their BPSS execution engines [6], [14].
- The Process Parts Order binary collaboration scenario was implemented using the two BP engines (BPSS Implementations) exchanging ebXML MSH-compliant messages over the HTTP (Hypertext Transfer Protocol) transport [5].
- The testbed tools were used to demonstrate that each of the BPSS engines properly executes the intent of the BPSS instance within the scenario.
- The testbed pilot was able to make evident a number of interoperability issues, provide a venue to resolve those issues, and demonstrate that the participating software products interoperate for the given scenario.

2.3 Tools

A major focus during the initial testbed activities was to show how a Web-based interoperability demonstration and testing infrastructure could satisfy needs of the customers and software vendors. Specific needs for such Web-based demonstration, conformance, and interoperability testing resulted in development and adoption of a collection of testbed tools:

- The Reflector is a testing tool that supports both disconnected and connected testing scenarios while allowing for the transactions to be routed to the specified end points, reflected to the originator, and stored in a permanent transaction log [1].
- The Business Process Monitor enables monitoring and conformance checking for choreographed transactions between business partners. The monitor currently supports ebXML BPSS. The tool provides a Web-based graphical user interface to monitor in real time the business interactions based on the ebXML BPSS specification.
- The Collaborative Content Checking tool enables specification and execution of content constraints that define

valid syntax, structure, or semantics of the business messages. This facility allows standard developers, users, and implementers to precisely specify, extend, and test for conformance with, semantics of a common data dictionary (lexicon).

- The Graphical Semantic Constraint Construction tool supports the Collaborative Content Checking facility. Manual encoding of syntactically valid content constraints is hard and tedious. This tool assists the user by offering an intuitive interface to construct the constraint specifications. The tool, which is motivated by natural language processing approaches [17], uses a set of classifications based on relatively low-level constraint semantics (such as cardinality, uniqueness, etc.) to guide the user.
- The Virtual Trading Partner aims to provide a reference implementation for a trading partner based on the ebXML BPSS specification. The user can utilize this tool to interact with the candidate system in a stepwise manner through a series of collaboration states. The tool generates a finite state automata from a BPSS instance and uses it for internal consumption.

3. Testing Requirements for B2B Integration

As the B2B testbed enters a new development phase, we summarize our lessons learned during the initial phases of development as the requirements that will be driving the new testbed advances. In this section, we give an overview of the requirements we consider imperative to making the testbed effective and relevant to the stakeholders.

3.1 Technology Perspective

We have identified the need to support both the horizontal and vertical testing among the components of integrated systems.

3.1.1 Horizontal Testing

From a technology perspective, horizontal testing involves obtaining assurance that components having the **same** set of functionalities conform to a relevant standard and/or interoperate. Figure 1 illustrates three cases of horizontal testing (illustrated with solid lines) that may be present during B2B integration. In the first case, message handlers (MSHs) are tested for interoperability with respect to the communication functionality. In the second case, business process engines (BPEs) are tested for interoperability with respect to workflow or collaboration handling functionalities. In the last case, complete business systems (including applications) are tested for interoperability with respect to application integration functionalities. As illustrated in the latter two cases, the dotted lines between the internal components indicate that internal operations of each candidate system are treated as black boxes.

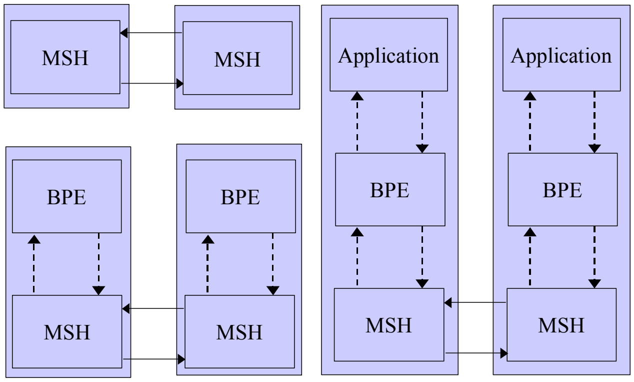


Figure 1. Horizontal Testing – the technology perspective.

3.1.2 Vertical Testing

Vertical testing involves providing assurance that two components with **different** function sets conform to relevant standards and can interoperate. Software vendors may develop products that provide limited functions and, necessarily, depend on the products that support these functions (e.g., application providers rely on middleware vendors). In some cases, users seek the best of breed solution and combine products from different vendors. These trends require software components (of different functionalities yet supportive of each other) from different vendors to interoperate.

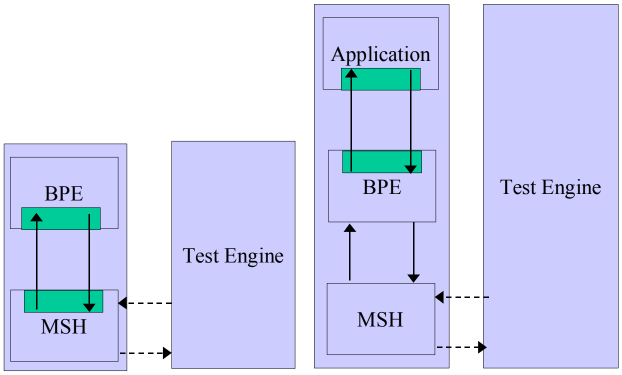


Figure 2. Vertical Testing – the technology perspective.

Figure 2 illustrates two possible cases of vertical testing (illustrated with solid lines) that may occur during B2B integration. In the first case, a workflow engine (i.e., BPE) needs to interoperate with a message handler component. In the other case, we need to ensure that a workflow engine can plug into a backend application to provide business services. In both cases, the dotted horizontal lines indicate that the external operations of each candidate system are treated as black boxes. Since for any two vertical components to interoperate, they must be able to provide functions required by one other, it is possible to use a similar testing facility for both horizontal and vertical testing needs (i.e., functional requirement tests).

3.2 Industry Perspective

From the industry perspective, the testing activities may be viewed as being done within an industry vertical or across industries. The two subsections below discuss this in further detail.

3.2.1 Vertical Testing

Each industry usually forms its own consortium, recognizes its own set of requirements, and promotes interoperability within that industry. In other cases, consortia are formed to cover respective industry value chains. For example, the automotive industry has the Automotive Industry Action Group (AIAG), the chemical industry has the Chemical Industry Data Exchange (CIDX), and the electronic component and high-tech industry has the RosettaNet consortium. However, within the automotive industry itself, for instance, there exist a consortium of car dealers and auto-manufacturers (STandard for Automotive Retailers (STAR)) and a consortium of automotive after-market manufacturers and retailers (Automotive Aftermarket Industry Association (AAIA)). These segmented consortia also promote their own requirements (although these may be influenced by the more general concerns of the industry).

The primary motivation for vertical interoperability testing is to reduce the business integration costs throughout the supply or value chain. For example, a car dealer who sells cars from multiple manufacturers needs to use a different software system to communicate with each manufacturer. In addition, similar situations exist for suppliers dealing with multiple manufacturers. When communication changes need to be introduced, each system needs to be updated. Interoperability testing, in this case, helps reduce the risk of introducing incompatible changes into vertically integrated systems and supports continuous improvement of these inter-dependent systems.

3.2.2 Horizontal Testing

As described in the previous section, each industry typically comes up with its own set of interoperability requirements. These requirements are being introduced at different levels including the technologies, business processes, and business semantics. When different industries need to integrate their business systems, interoperability issues appear at all levels of business system stacks due to the differentiated requirements.

For example, when the automotive manufacturer wants to integrate their procurement system with electronic OEM for electrical parts or chemical manufacturers for paints, horizontal interoperability issues arise. Furthermore, when each industry vertical changes their business process or information requirements in their system, these changes cannot and are not automatically propagated. After changes are applied, new interoperability assurance is needed. Without effective and efficient interoperability testing approach, significant costs are incurred that discourage continuous improvement.

3.3 B2B Interoperability Testbed Focus Areas

This section describes the requirements identified through our experience on e-business architecture and testbed tools development.

3.3.1 Infrastructure

This focus area involves obtaining assurance that systems can communicate in a secure and reliable manner. Several optional functionalities defined in the infrastructure specification (such as ebXML Messaging Service) are potential problems for achieving interoperability. In addition, the availability of multiple algorithms for security, authentication, and confidentiality for multiple transport protocols impact interoperability at the infrastructure levels.

3.3.2 Business Document or Content Semantics

This focus area seeks to achieve semantic interoperability of the content exchanged between applications and partners.

As content standards have been increasingly built with flexibility to support users in various industry sectors, formal semantics and structure requirements have been placed into separate layers of specifications and some are delayed until the standard implementation. In addition, most popular schema languages do not provide sufficient expressiveness to support accurate and precise semantic expression. An extensively sophisticated schema specification will be too complex for implementers to effectively use. A simplistic schema specification, on the other hand, will be too loose and will allow for imprecise specifications and interoperability problems that are hard to resolve. Nevertheless, the separation of lexicons, structures, and semantics of a content specification into layers positively affect the standard adoption and, hence, realization of a common data dictionary. Examples of such content specification are OAGIS 8.0 [12] and ebXML Core Component [3]. This focus area of the B2B interoperability testbed seeks to complement a standard structure by providing a facility for the standard developers, standard customers, and implementers to precisely specify, extend, and test for conformance with, semantics from the common data dictionary (lexicons).

3.3.3 Business Process Specification

Typically, after businesses have successfully formalized their structure and semantics of business documents, a business process specification (BPS) or a business process model is used to formalize the business scenarios (i.e., capturing artifacts), which utilize those business documents. The Business Process Specification can be used to manage (or automate) business integration as follows:

- Align business states throughout periods of collaboration, regardless of length.
- Specify message choreography including validation of signals and actions, and process and security controls.
- Specify success, failure, exception, and timeout and retry conditions of business trading, as well as determine recording requirements.
- Capture legal/security requirements such as non-repudiation.

These functionalities indicate that BPS plays a crucial role in communicating business and integration requirements as well as governing business (legal) commitments. Hence, potential interoperability challenges exist in utilizing the business process.

The specification that governs the BPS grammar is, for instance, ebXML Business Process Specification Schema (BPSS). This specification only specifies a grammar for modeling business process collaboration requirements. There is a significant potential that business process engines from different vendors interpret and implement BPSS parameters differently. Several interoperability challenges exist in the business process specification focus area:

- BPS (instance of BPSS) typically developed by business expert needs to be tested at run time to ensure that it satisfies business and technological requirements (a business expert usually does not have sufficient knowledge of technological requirements).
- During the transitions and conditions that occur at the time of executing a business transaction and/or collaboration, either trading partner may not be certain of the current collaboration state. Due to different interpretation of timeout, retry, and legal binding, the transaction may be in an unknown state.
- There exist multiple levels of commitment controlling the proceeding of the B2B collaboration. The hierarchy of different state machines and the interaction between them has not been fully defined nor how it affects the business process collaboration.
- The business processes change, often times without proper redress of the business process or notification to the actors involved. This creates interoperability challenges.

3.3.4 End-to-End Integration

The testing activities in each of the previous three focus areas are directed toward achieving conformant and interoperable functionalities as specified by the target standard specification. The End-to-End (E2E) integration testing aims at ensuring that business systems can achieve the goal of business integration; consequently, it focuses on the business semantics of information exchanged in the collaboration.

Figure 3 shows a simple E2E test case of an information integration situation where a candidate system (on the left hand side) imports invoice information. The testbed may test this by issuing a business document "Sync Invoice" expecting the candidate application to process this document and store invoice information into its data store. The testbed then could attempt to verify the application processing by sending another request using the "Get Invoice" business document to retrieve the same invoice. The testbed expects the candidate system to respond with the "Show Invoice" business document, from which the testbed compares the invoice information with the original invoice information in order to verify the success of this information integration. Typically, test cases written for E2E integration testing need to be narrowed down by specifying context in order to bind the intended semantics and allow for a meaningful test case. As shown in the figure, the testing scenario focuses on invoice containing hazardous material.

4. ebXML IIC Specifications

In this section, we discuss a promising test framework architecture that we are currently investigating as a basis for advancing the testbed capabilities. The requirements that have been summarized in the previous section are steering this investigation.

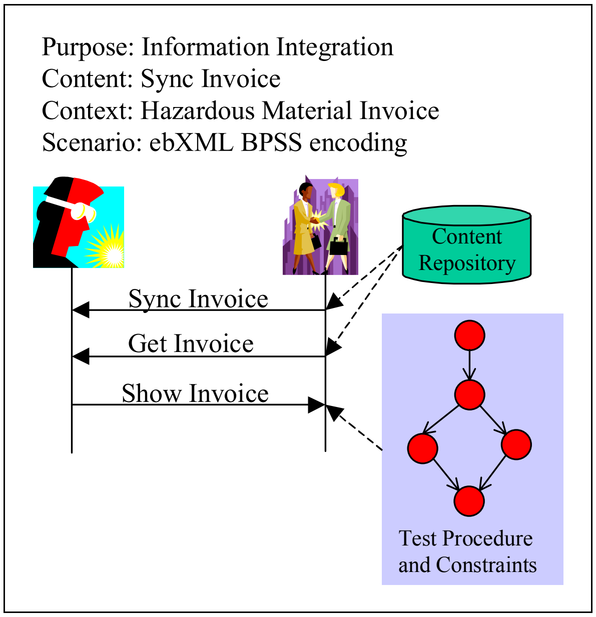
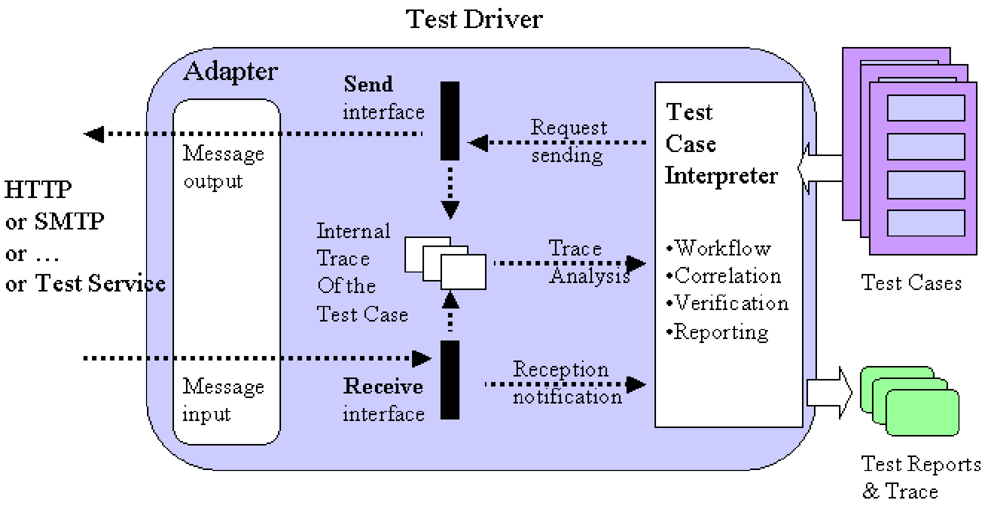

Purpose: Information Integration
Content: Sync Invoice
Context: Hazardous Material Invoice
Scenario: ebXML BPSS encoding
Content Repository
Sync Invoice
Get Invoice
Show Invoice
Test Procedure and Constraints

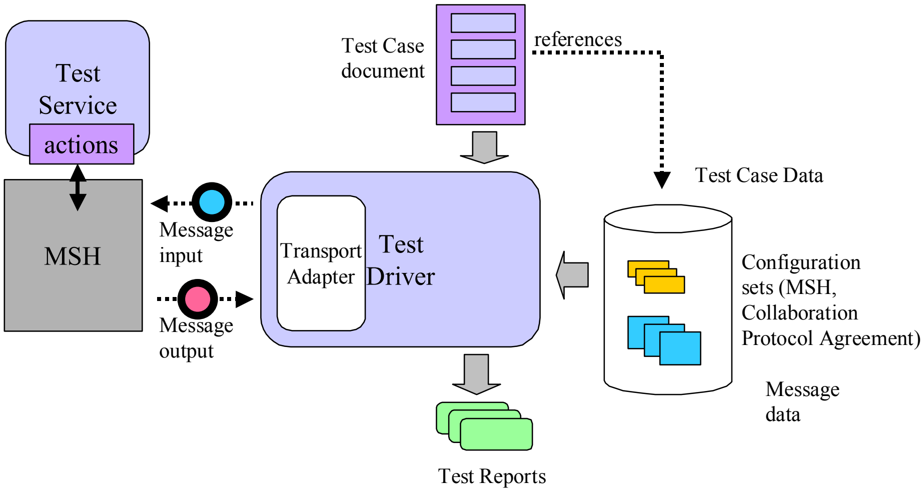
Figure 3. An Example End-to-End Integration Testing Scenario.

The ebXML Interoperability, Implementation, and Conformance (IIC) Technical Specification defines a new test framework for automating conformance and interoperability testing. The ebXML IIC specification is being developed under the Organization for the Advancement of Structured Information Standards (OASIS) consortium. The goal for the ebXML IIC specification is to define a test framework that is capable of supporting all testing needs that the ebXML specifications may need, from the messaging specifications to content specifications. Its current focus is to enable the ebXML MSH testing. With this focus, the test framework can support horizontal technology and industry testing requirements at the infrastructure level.

Presently, the ebXML IIC specification includes test framework and procedures to automate the test and verification of conformance and interoperability requirements. In addition, a number of interoperability and conformance test requirements and test cases for ebXML MSH and Registry and Repository have been created. The Test Framework includes distributed software components. Some of them will be local to the testbed; others will be downloaded by e-business partners. For conformance tests, the testbed can be used as a service. For interoperability tests, the testbed can be used as a hub, providing monitoring and routing functions or a service to implement the tests. The Test Framework also includes automated testing functionality where the testbed is used to control end-point(s).

The IIC test framework is the only interoperability initiative among others (such as the Web Services Interoperability Organization [16]) that defines a complete set of specifications for test materials and architecture necessary to perform the interoperability and conformance tests. The framework has strength its extensibility and high-configurability to cover all stacks of ebXML specifications. This strength is partially inherited from the ebXML specifications themselves, because ebXML has started out top down defining all necessary stacks for B2B collaboration. IIC also has a plan to embrace the





interoperability of the Web Services standards by enhancing its test framework specifications.

4.1 Overview of IIC Test Framework Architecture

This section provides an overview of components and functionalities of the IIC Test Framework. Figure 4 illustrates the **Test Driver** as designed by the IIC. The Test Driver is the brain of the Test Framework. It is responsible for driving and verifying the test, and validating the results.

The Test Driver can be used in both conformance and interoperability testing. When the Test Driver is used for conformance testing, it utilizes its own message handling functionality. When the Test Driver is used for interoperability testing, it controls candidate systems and monitors and verifies interactions through **Test Services**. The Test Service simulates higher-level components in the e-business stack. Test Service receives **Service Actions** (commands) from the Test Driver and performs its functionality as specified in the IIC Test Framework specification. The Test Service also notifies the Test Driver of incoming messages used for verification. Figure 5 and Figure 6 illustrate where Test Services and Test Driver fit in the test environment (also called **Test Harness**). Figure 5 shows the two components working for conformance testing. Figure 6 shows the case of interoperability testing. There are two possible test harnesses in the interoperability testing– local or remote test service. The Test Service receives the Service Action through an embedded element in the message header (i.e., in the action element according to the ebXML Message Specification) sent from the Test Driver.

4.2 Overview of IIC Test Material Artifacts

Another important artifact in the IIC Test Framework specification is the test material. At the abstract level, test materials include test requirements, test suites, test cases, test profiles, and message data. Several test metadata are also worth mentioning although they are not used for test execution. Test metadata indicate robustness of the test materials. This section describes test materials and test metadata in further detail. They are important concepts for conformance and interoperability testing.

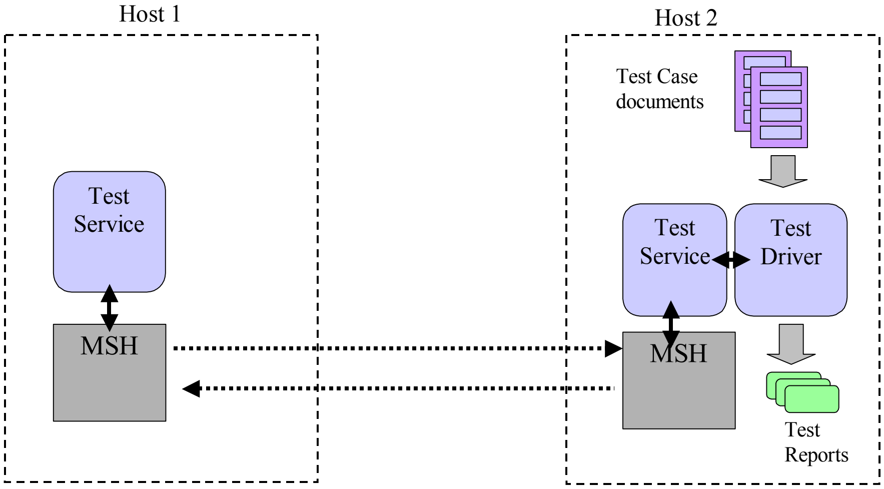
Figure 7 illustrates conceptual relationships between key elements in the test materials (note that this diagram is neither complete nor normative, it is only used to illustrate these concepts).

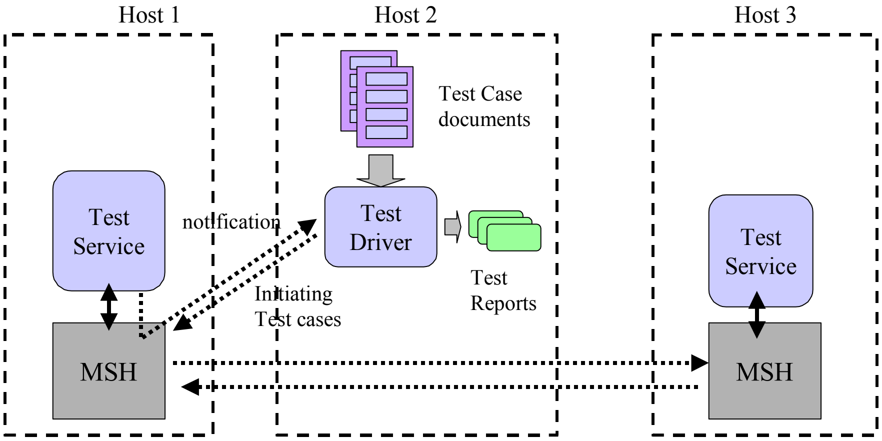
Each **test requirement** aggregates test assertions, which provide formal representation of functional requirements stated in the target standard. Each **test assertion** is verifiable by computer to be true or false.

Each **test case** indicates how a test requirement should be executed through a number of **test steps**. Each test step contains **action** and materials including the **message data** and **configuration data** necessary for test setup and test requirement verification. The **Test suite** is a container that groups together related test cases, while a **test case instance** is the term referring to an execution of a test case.

Test profile is in fact an entry point to the Test Driver. Test profile groups test requirements together to form, for example, conformance levels or interoperability groups (e.g., interoperability requirements for an industry vertical). The Test Driver reads a test profile and its associated test requirements, and then searches for and executes related test cases (that implement those test requirements).

The **specification coverage** relationship between the test requirement and **specification document** indicates levels in which the test requirement can formally express functional requirements set forth in the target standard specification. Values of the coverage level can be *partial*, *full*, or *none*. This limitation may be due to limited capability of the test framework such as expressiveness underlying the formal language used. However, the limitation may be caused by the functional requirement in the target standard. If the latter is the case, it can be an indicator that such a functional requirement is not realizable because it is not





verifiable.

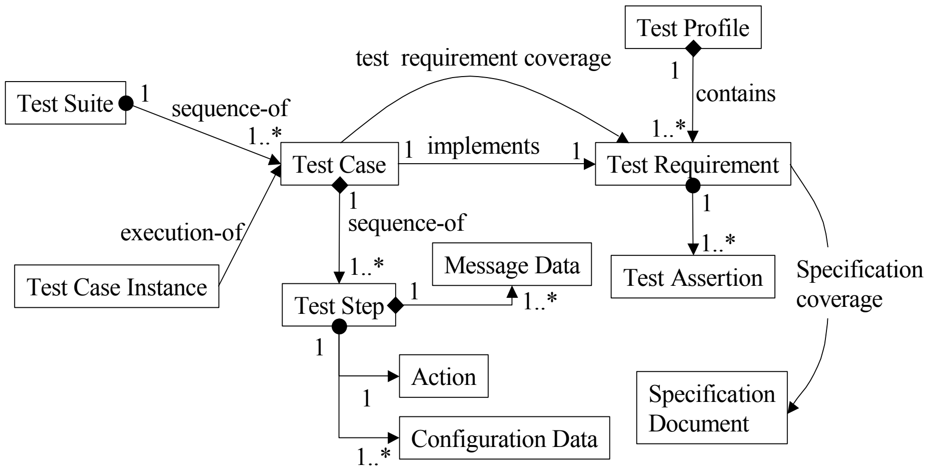
The **test requirement coverage** relationship between the test case and the test requirement indicates the level in which the test case can implement the associated test requirement. Similarly, values of the coverage level can be *partial*, *full*, or *none*. Technical difficulties or limitations that can be associated with the test framework (particularly test harness) or an uncontrollable environment can prevent full test requirement verification. This coverage value reflects these constraints.

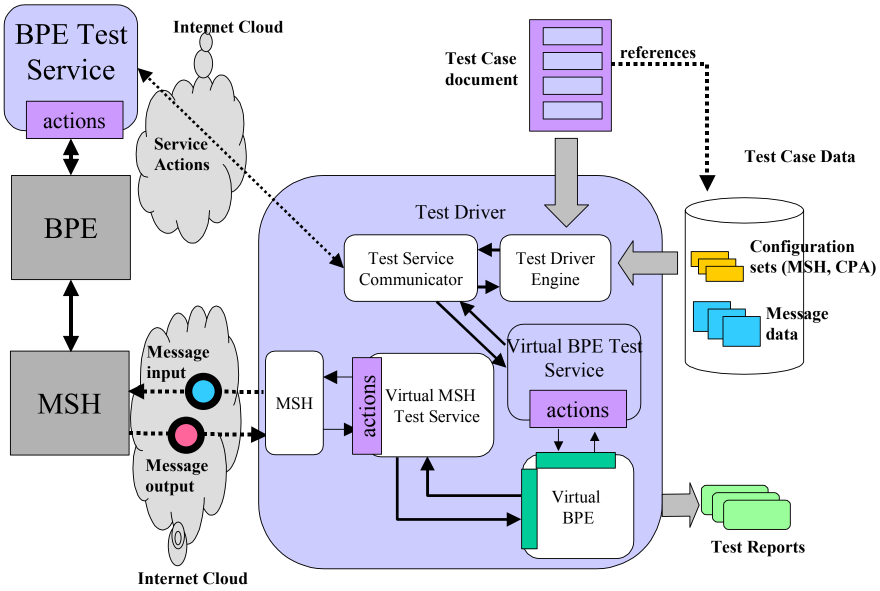
The concepts described in this section are essential and can be scaled to any interoperability testing need described in Section 3.

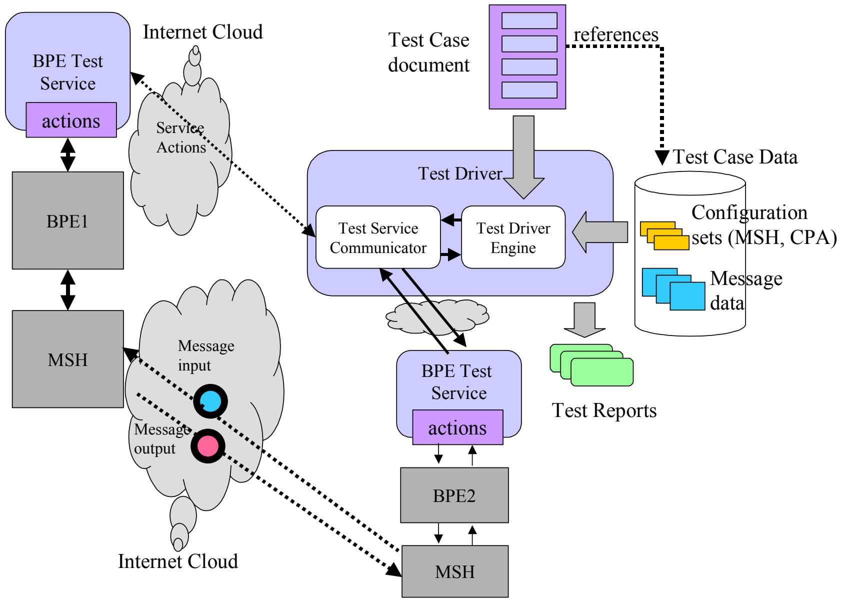
4.3 Overview of ebXML IIC Test Framework Extension

This section provides an overview of an initial investigation to extend the IIC test framework, particularly to accommodate testing of the business process engine (e.g., implementations of the ebXML BPSS). Figure 8 and Figure 9 illustrate possible extensions to support a business process engine test for conformance and interoperability. In these two cases, the BPE Test Service simulates higher-level component receiving controlled command as well as reporting status/messages from and to the Test Driver, respectively. For example, the Test Driver needs to control the candidate system to start, end, or cancel collaboration. Collaboration status and messages are also reported back to the Test Driver for verifications.

Unlike the original IIC test framework, where the Test Driver controls a candidate system only through the same channel used by business messages, the Test Driver in Figure 8 and Figure 9 can communicate directly with the Test Service. These capabilities can be deemed as required for two reasons. The first







reason is more specific to ebXML, that the 'action' field in the ebXML header used to pass on the Service action for MSH testing is now used by the BPE. Hence, the BPE will attempt to consume that action before it reaches the BPE test service. The other reason is more general but follows from the first reason: relying on a specific message header protocol to pass on the service commands will limit the scalability of the test framework to handle multiple standards.

Figure 8 also illustrates details of the Test Driver internals. The essential point of this illustration is that the Test Driver internals have the same structure as its externals; that is, it uses Test Services to connect with virtual (or reference) components in the conformance testing. This decouples the Test Driver from specific component implementation. This separation enables easy switching from conformance to interoperability testing. From Figure 8 to Figure 9, the Test Service Communicator only changes its endpoints. There is virtually no additional code or change required for the Test Driver to handle conformance and interoperability testing.

We continue to extend the IIC test framework and cover all of the B2B testbed focus areas. Another next step is to enhance the test material in support of all focus areas. For example, a test case currently includes only linear sequential test steps initially; however, to support Business Process Specification and E2E testing, conditional and looping test steps are needed.

5. Conclusion

The OAG/NIST B2B Interoperability testbed is a unique initiative to drive the advancement of interoperability testing and the state of the art in B2B integration. The testbed provides a neutral environment for collaboration where industry partners meet, identify problems, and find solutions. On the other hand, it provides a venue for researchers to collaborate with the industry partners and identify pragmatic research topics and prototypical scenarios to advance the integration technology. During the initial phase of testbed development, we have worked with industry partners and gathered lessons learned that are reported within this paper. We formalized them into the interoperability testing requirements. Initial tools have been developed to support those requirements on an incremental basis. We have started a top-down approach to explore a common testing framework with the aim of satisfying all the testing requirements using the ebXML IIC testing framework.

6. Disclaimer

Certain commercial software products are identified in this paper. These products were used only for demonstration purposes. This use does not imply approval or endorsement by NIST, nor does it imply that these products are necessarily the best available for the purpose.

7. REFERENCES

[1] Accordare Web Site. The Reflector. http://www.accordare.com

[2] Conger., S., and Loch, K.D. (eds.). Ethics and computer use. Commun. ACM 38, 12 (entire issue).

[3] DISA UN/CEFACT Web Site. EbXML core component specification version 1.9. http://webster.disa.org/cefact-groups/tmg/downloads/CCWG/for_review/CCTS_V_1pt90.zip.

[4] EbXML Proof-of-Concept Working Group Web Site. Proof-of-concept working group. <http://www.ebxml.org/project_teams/poc/vienna/>.

[5] Fielding, R., Getty, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T. (June 1999). Hypertext Transfer Protocol -- HTTP/1.1 specification, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc2616.txt

[6] Fujitsu Web Site. Interstage application server. http://www.fsw.fujitsu.com/INTERSTAGE/index.html

[7] Mega International Web Site. Business process modeling. http://www.mega.com/us/product/megaprocess/.

[8] OASIS Conformance TC conformance resource. Glossary of conformance terminology. http://www.oasis-open.org/committees/ioc/glossary.htm

[9] OASIS ebXML Implementation, Interoperability, and Conformance Technical Committee Web Site. EbXML IIC test framework version 1.0. http://www.oasis-open.org/committees/documents.php?wg_abbrev=ebxml-iic.

[10] OASIS ebXML Messaging Services Specification Technical Committee Web Site. EbXML messaging service specification version 2.0. http://www.oasis-open.org/committees/documents.php?wg_abbrev=ebxml-msg

[11] Open Application Groups News Page. Ford, Lucent and Lockheed Martin lead the OAGI vendor challenge. http://www.openapplications.org/news/000403.htm

[12] Open Application Groups Web Site. Open Application Group Integration Specification (OAGIS) version 8.0. http://www.openapplications.org/downloads

[13] STAR (Standards for Technology in Automotive Retail). Making the case for IT standards in retail automotive. STAR publication, 2003.

[14] Sybase Web Site. Enterprise application server. http://www.sybase.com/easerver.

[15] UN/CEFACT Web Site. EbXML business process specification schema version 1.01. http://www.ebxml.org/specs/ebBPSS.pdf

[16] Web Services Interoperability Organization. http://www.ws-i.org

[17] Winograd, T. (1972). Understanding Natural Language. New York: Academic Press.