

Ford Motor Company

VOLVO



mazda



LINCOLN



Mercury



JAGUAR

ASTON MARTIN

LAND-ROVER



Ford Motor Company Case Study

(Today's view)

Version: 1.0

28 September 2001

Legal Disclaimer

This document contains information that is confidential and proprietary to the Business Internet Consortium and Ford Motor Company. The BUSINESS INTERNET CONSORTIUM name and design logo is a trademark of the Business Internet Consortium. All other trademarks and brands are the property of their respective owners.

Copyright

©2001 Business Internet Consortium and Ford Motor Company. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

Contents

Version History	v
------------------------------	----------

Preface	vii
----------------	------------

1 Architecture Conceptual Model	1
--	----------

2 Architecture Layers – Current Instantiation	3
--	----------

2.1 Backend Integration.....	3
------------------------------	---

2.2 Service-oriented Architecture	3
---	---

2.3 Network Transport.....	3
----------------------------	---

2.4 Core XML Standards	3
------------------------------	---

2.5 Messaging	4
---------------------	---

2.5.1 Packaging	4
-----------------------	---

2.5.2 Reliable Messaging.....	5
-------------------------------	---

2.6 Dictionary / Repository.....	5
----------------------------------	---

2.7 Business Content Format Definition	5
--	---

2.8 Universal Business Content	5
--------------------------------------	---

2.9 Specialized Business Content	6
--	---

2.10 Business Content Instance.....	6
-------------------------------------	---

2.11 Directory & Registry.....	7
--------------------------------	---

2.12 Service Description	7
--------------------------------	---

2.13 Process Description.....	7
-------------------------------	---

2.14 Universal Business Processes	8
---	---

2.15 Specialized Business Processes.....	8
--	---

2.16 Business Process Instance	9
--------------------------------------	---

2.17 Trading Partner Agreement.....	9
-------------------------------------	---

2.18 Security	9
---------------	---

2.19 Management.....	9
----------------------	---

Appendix A References.....	10
-----------------------------------	-----------

Appendix B Glossary.....	11
---------------------------------	-----------

Figures

Figure 1. B2B Integration Conceptual Model	1
--	---

Figure 2. Structure of Ford Motor Company's XML Message Header.....	5
---	---

Figure 3. Diagram Used for Requirements Definition7
Figure 4. Example of Process Description Used to Configure Middleware8

Tables

Version History

Version 0.1	29 August 2001	Tim Thomasma drafted based on Ford Motor Company internal documents
Version 0.2	21 September 2001	Revision based on input from Dawn Xu and discussion at BIC teleconference
Version 1.0	28 September 2001	Final version

Preface

Purpose of the Document

This document uses current Ford Motor Company experience with XML messaging to demonstrate how the B2B Conceptual Model could be used to analyze current solutions. It shows the relevance of the model to real-world implementation. It also shows what works today in sections of the model, and which other pieces are still missing and less defined.

Intended Audience

e-Business architects and business managers responsible for strategy and to implement B2B solutions; B2B standard bodies (W3C, OASIS, OAGI, etc.); B2B vendors and solution providers; Members of other BIC workgroups.

Prerequisites

Readers of this document should have read: *High-Level Conceptual Model for B2B Integration*.

Scope of the Document

High-level architecture discussion, does not contain implementation details.

Structure of This Document

Follows the layers described in the High-Level B2B Conceptual Model.

Acknowledgements

Many thanks to Dawn Xu for her review and input based on experience with XML messaging projects all around the enterprise.

1 Architecture Conceptual Model

Figure 1 is a conceptual B2B model. A detailed description of this model is available in a separate document (*High Level Conceptual Model for B2B Integration*).

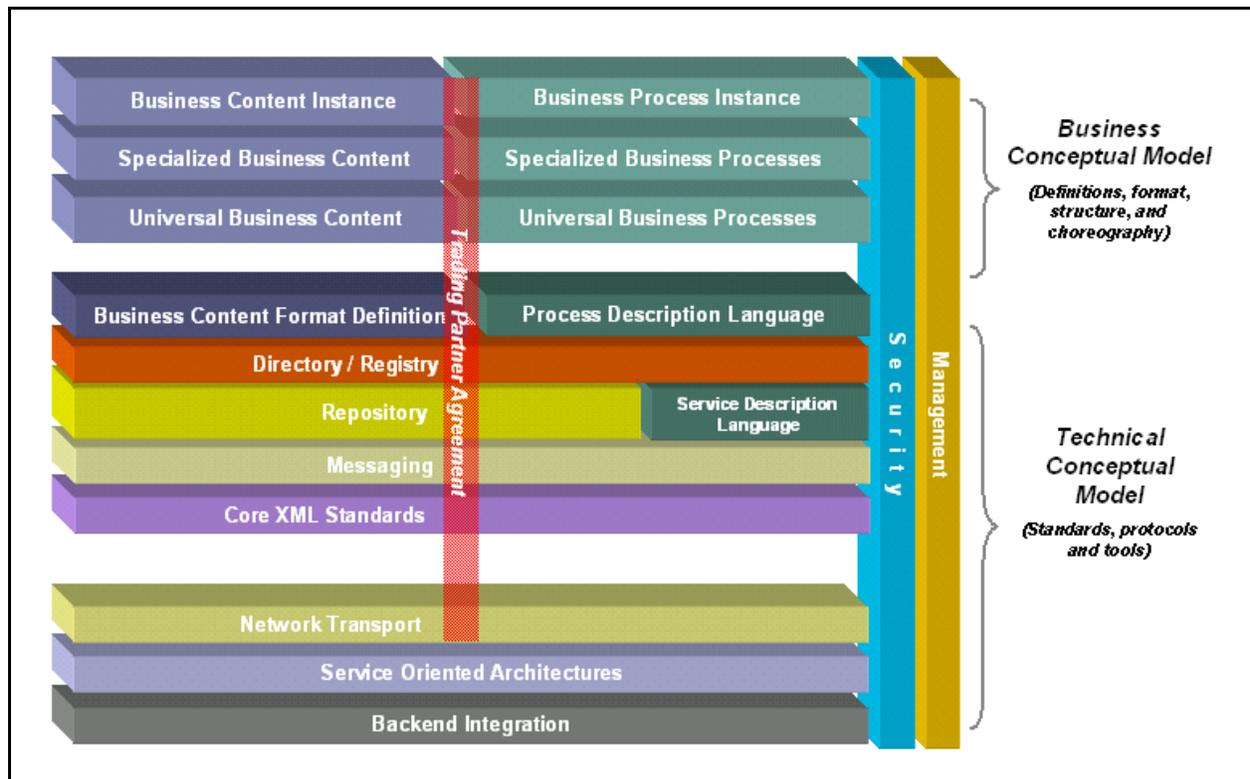


Figure 1. B2B Integration Conceptual Model

The bottom two layers (Service-oriented Architecture layer and Backend integration layer) are in the scope of a much more complete Application Architecture reference model that we have developed at Ford Motor Company. They are not in the scope of our understanding of what standards or agreements are needed in order to make possible applications integration and B2B enabled by XML messaging.

For completeness, we include information on what we do with backend integration and service oriented architecture, even though we consider it out of scope.

Caution: Please read the Conceptual Model document for descriptions of each layer before proceeding with the following sections.

2 Architecture Layers – Current Instantiation

This section references Figure 1. The following subsections describe each layer of instantiation from the bottom to the top and from left to right.

2.1 Backend Integration

We expect applications to provide open standard interfaces. We use the OAGIS compliant interfaces that are provided in some of the business application software packages that we have purchased. We find that leveraging open standard interfaces implemented natively by business application software vendors in their products is the low cost, high value approach for A2A and B2B integration.

In order to enable the applications we've built ourselves to participate in XML messaging collaborations, we build new interfaces on them to emit and receive the standard XML documents we use. It is relatively easy to write the programs to do this. Now that the latest versions of the database products we use come with tools for reading and writing information from XML documents into relational tables, this task is even easier.

2.2 Service-oriented Architecture

We use SOAP to build request/reply interfaces in some of the applications we develop.

We don't find service oriented architectures to be necessary to integrate applications or to implement B2B using XML messaging, nor do we see a need for trading partners to agree to use any particular architecture in building their applications.

We find service oriented architectures attractive for other reasons when we build new applications, not because we think they're necessary for B2B collaborative applications that use XML messaging.

2.3 Network Transport

Generally, we use HTTP over TCP/IP for message transport. In some applications we use SMTP to transfer business content.

2.4 Core XML Standards

We maintain both DTD and XDR versions of the XML documents we place on our internal Ford Motor Company repository. In some cases we may require a human-readable message guideline to capture the complete message structure, as well as additional validation logic and constraints, external to the DTD and XDR. We believe it will still be necessary to do this even with XSD, although we expect to be able to leverage the increased richness of the XSD technology.

2.5 Messaging

Ford Motor Company is just now in the process of establishing internal standards for message packaging and reliable messaging.

2.5.1 Packaging

Ford is adopting OAGIS BOD as business content standard, and ebXML Message packaging as communication protocol that is structured in compliance with the SOAP 1.1 Messages with Attachments [SOAPPATTACH]. We replicate some content from the OAGI BOD Control Area into the ebXML header.

("Carrying ebXML headers in SOAP Messages does not mean that ebXML overrides existing semantics of SOAP, but rather that the semantics of ebXML over SOAP maps directly onto SOAP semantic." - "STAR XML Infrastructure Specification Draft version 0.5 ")

The ebXML and OAGI specifications leave room for interpretation. We have made the effort to map our standard Ford header to headers used in various applications we've purchased and headers used by Covisint.

Figure 2 below shows the structure of the Ford Motor Company header. Note: Namespaces are removed to accommodate a temporary limitation of the XML messaging middleware product we're using.

We believe this new header will be useful. Currently, the only portion that we use is not much different from information that the BOD itself carries in its Control Area.

We have only recently produced this standard Ford Motor Company header. Previously, we have used SOAP headers or simply used the OAGI BOD without additional envelope.

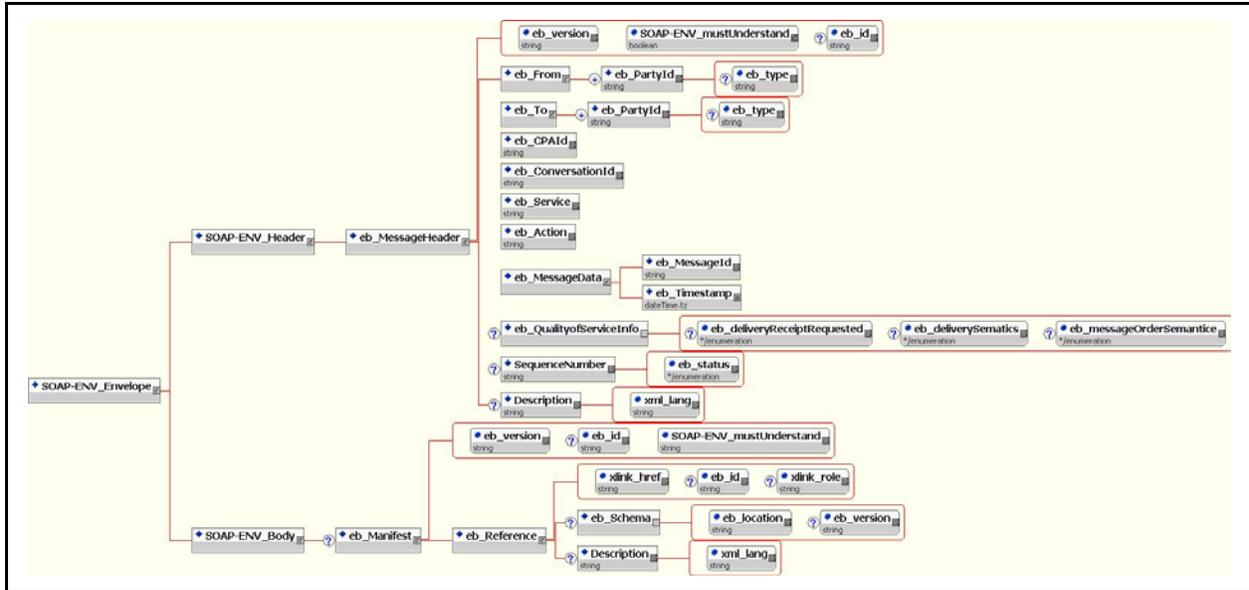


Figure 2. Structure of Ford Motor Company's XML Message Header

2.5.2 Reliable Messaging

In our initial implementations we queue the messages in front of the HTTP Post routine. We do retries in case we receive error codes in the acknowledgements of the HTTP posts. We also maintain transaction logs on both sides (HTTP post and HTTP listener).

We are now implementing the ebXML Reliable Messaging specification, using default settings agreed in the Standards and Technology for Automotive Retailing XML project.

2.6 Dictionary / Repository

We have an internal web site where we place XML DTDs, XDR Schemas, examples, mappings and data dictionaries. At present this serves our purposes. As the content in the repository grows, we have plans to use a tool for maintaining the repository.

2.7 Business Content Format Definition

We use OAGI's business content format definition structure. When we build our own document definitions, we follow patterns established in OAGIS.

2.8 Universal Business Content

OAGIS.

2.9 Specialized Business Content

We communicate additional Ford usage information beyond the OAGIS definitions in mapping spreadsheets and other documents we post at our repository web site and communicate with trading partners. For example, the work-in-process status codes we send in messages to our outbound logistics provider are documented in requirements specifications we have agreed to with them.

We feel that a large amount of the content we have produced ourselves and placed in our repository is really universal in nature, and we are bringing this to the OAGI to add to their industry-horizontal specification.

As we approach specialized problems we typically follow these steps:

- Map the definitions of our terminology to the appropriate field definitions in an OAGIS BOD.
- For our data that does not fit into the BOD, look in the OAGIS dictionary for matching terms that are not in the BOD we are using.
- For the rest of the data elements, extend the OAGIS dictionary by adding our terms. We distinguish between Ford-specific terms and new cross-industry terms we should submit for inclusion in OAGIS.
- Place our added terms, if any, into the userarea elements of the BOD in order to produce an extension.
- Version the extension.
- Create sample messages to reflect trading partner usage.
- Convert the extended schema and examples to ones that our XML messaging middleware can process.
- Log them into the XML repository web site.
- Bring our suggestions to OAGI for BOD revisions and new BODs.

We do not want to do intermediate conversions between formats within our middleware.

The BOD design needs to support versioning of the standard and of various levels of extensions.

2.10 Business Content Instance

We generally include sample instances of all our XML documents in our repository. Our applications and our XML messaging middleware is programmed to manage the XML instances.

2.11 Directory & Registry

We use the support our XML messaging middleware provides for recording endpoint definitions (sources and destinations of the XML messages that are sent).

2.12 Service Description

This is included in the requirements specifications of the various applications that are involved in the e-collaborations that we build.

2.13 Process Description

We use OAGI's scenario diagrams to locate Business Object Documents that apply to our projects. During requirements definition, we use simplified UML sequence diagrams that we annotate. An example is shown in Figure 3.

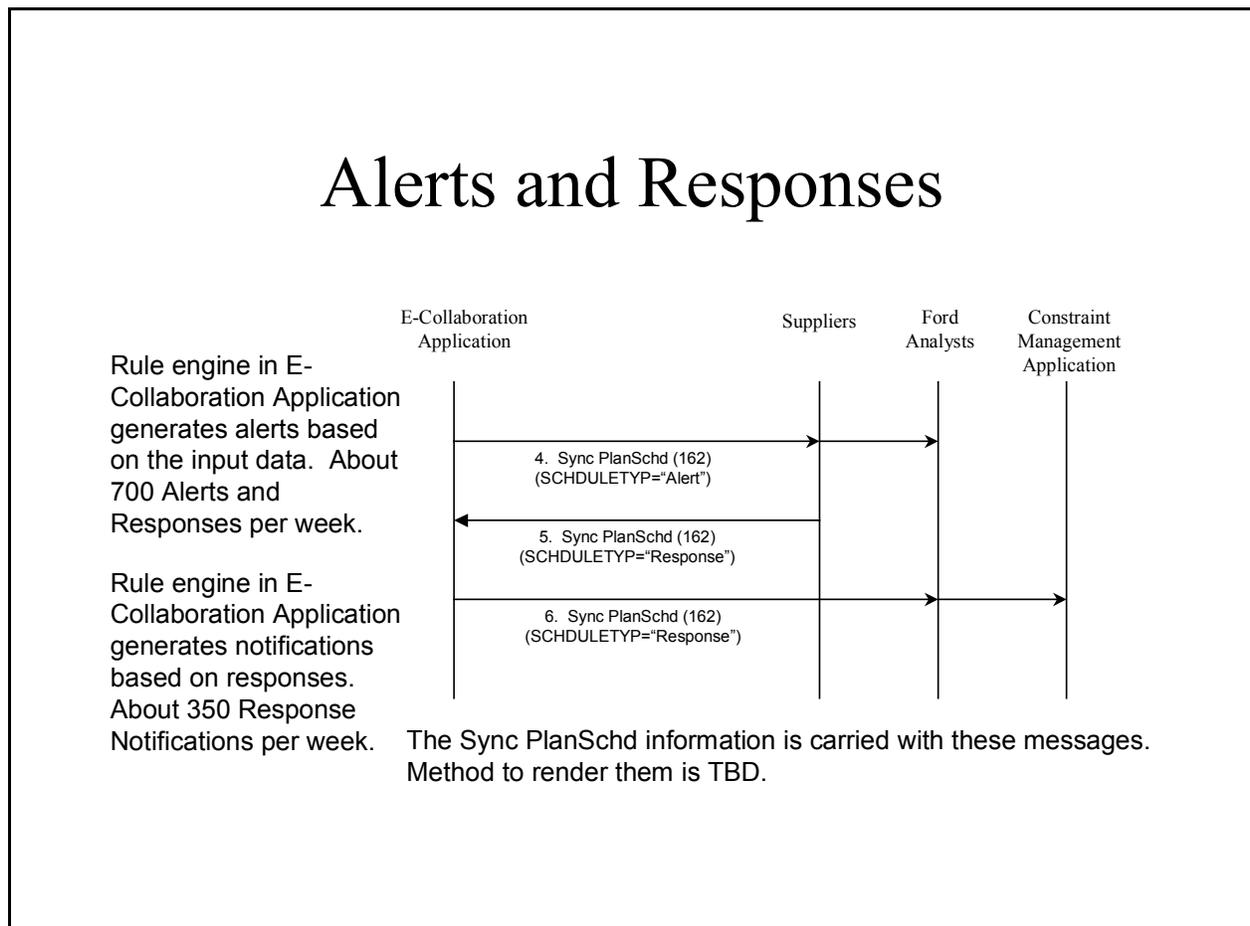


Figure 3. Diagram Used for Requirements Definition

We describe processes to our middleware, using a diagramming notation that the tool provides. It's quite easy for experts familiar with the tool to produce this diagram from the diagram and other instructions used in requirements definition. Figure 4 shows an example. The tool then generates XLANG XML from this diagram to support process execution at run-time.

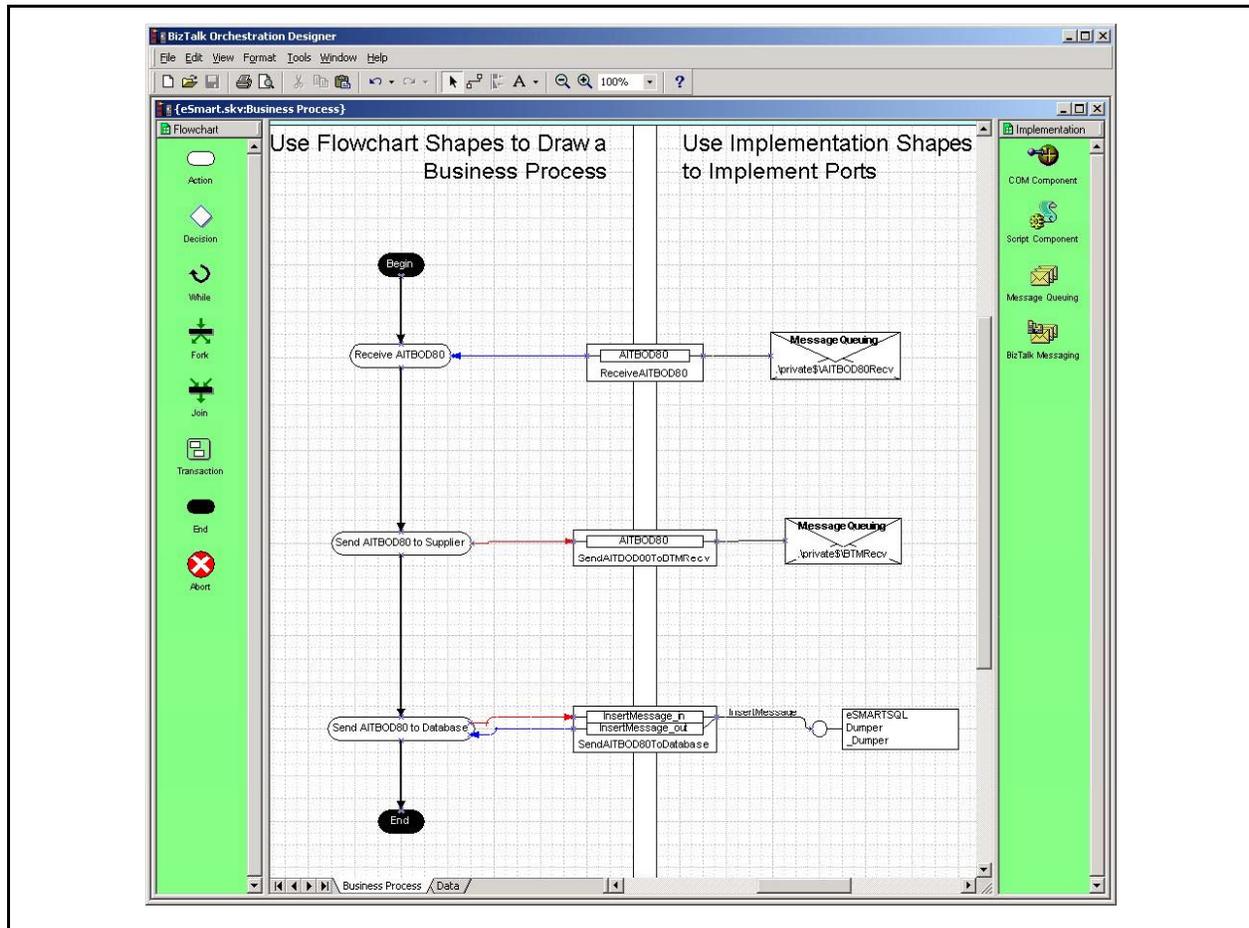


Figure 4. Example of Process Description Used to Configure Middleware

2.14 Universal Business Processes

In the course of requirements definition, we refer to the library of OAGI Scenarios. We may also look at RosettaNet Partner Interface Processes and models produced in UN/CEFACT ebXML activities, as these become available.

2.15 Specialized Business Processes

We record definitions of the business processes that we actually implement in the requirements specifications and technical documentation of our systems. The business

process information is communicated to all the trading partners involved in the business process the systems support.

2.16 Business Process Instance

The business process instances are implemented in XLANG in our middleware, in other workflow middleware tools, and principally in the business logic that the applications implement.

2.17 Trading Partner Agreement

The agreements we make with trading partners are recorded in the systems requirements specifications that we jointly develop.

2.18 Security

We favor securing the message, rather than relying on a secure channel. So far, we have not implemented this.

Current implementations use IPsec security that the automotive industry's extranet (ANX) provides. In the near future we will deploy applications that use SSL for encryption and XML Signature for authentication and non-repudiation as part of our involvement with the Standards for Automotive Retailing XML project.

We rely on management tools and manual procedures to detect and respond to attempts to breach security. These measures are reviewed at the time applications are launched and they are periodically reviewed by an internal systems auditing group.

2.19 Management

Procedures to manage the applications, middleware, and infrastructure involved in the solution are a key part of any B2B or integration solution. Notification and escalation procedures must be in place and agreed by all parties for handling discrepancies, systems failures, disaster recovery, etc. We develop these with our trading partners and record them in the requirements specifications of our systems.

There are specifications in ebXML that cover some of these issues, which we plan to implement in future solutions.

APPENDIX A REFERENCES

- All specifications, technical reports and other information are available for free download from the ebXML, OAGI and W3C web sites:
 - www.ebXML.org
 - www.openapplications.org
 - www.w3c.org

APPENDIX B GLOSSARY

ANX: The extranet service provider used in North America by DaimlerChrysler, Ford, GM and their suppliers

BOD: Business Object Document, OAGI's term for an XML document used in an XML messaging collaboration between applications.

Covisint: an automotive industry e-Marketplace

OAGI: Open Applications Group, Inc.

OAGIS: Open Application Group Interoperability Specification

OASIS: Organization for the Advancement of Structured Information Systems, a co-sponsor of ebXML

SOAP: Simple Object Access Protocol

UML: Unified Modeling Language

UN/CEFACT: United Nations activity that co-sponsored ebXML

W3C: World Wide Web Consortium

XLANG: are both process execution languages, actually both are based on the same formalism: π -calculus