



UNIVERSITY OF
CALGARY

**Tutorial Report for
SENG 609.22- Agent-based Software Engineering**

Course Instructor: Dr. Behrouz H. Far

XML and Agent Communication

Jingqiu Shao

Fall 2002

XML and Agent Communication

Jingqiu Shao

Department of Electrical and Computer Engineering,
University of Calgary,
2500, University Drive, N.W.,
Calgary, Alberta, Canada, T2N 1N4

Email: shao@enel.ucalgary.ca

Abstract

Developers of software agents should be mindful of the need to support some form of agent communication. By designing agents to perform specialized tasks, and to work cooperatively with other agents, system complexity can be reduced. Sharing a common form of communication is vital to the success of agent cooperation, and there are many solutions available to bridge the communication gap. XML is a good solution to agent communication. In this paper, XML is briefly introduced as well as growing need for agent communication. I also give some explanation why XML is fit for agent communication. Finally two application examples are given to show how XML is applied in agent communication.

1. Introduction

The industry is currently facing a lot of challenges. Business models need to be adapted, taking into account the liberalization of the market, the advent of new types of players and business interactions. In this very dynamic and technology-driven environment, control and automation of all those interactions has become more important than ever. Also interoperability issues between heterogeneous networks, different service platforms, or different terminal types need to be solved. Fast delivery of new services to the customer is not sufficient anymore and service providers need to think about personalization and user mobility. Software agents using XML languages can offer an attractive way of solving many of above issues.

2. Overview

2.1 what is XML?

Nearly everyone is talking about XML currently, but few people really understand what's behind the buzzword. XML stands for Extensible Markup Language.

Extensible

Extensible means "easy to extend without breaking." It is currently very rare to find a program that can read files saved by a later version of itself. For example, you can't open

a Word 2000 file in Word 6.0. Documents stored in an XML format have a sort of built-in immunity to incompatibly issues like this.

Markup

Have you ever looked at the source of an HTML or Web page? If so, you have seen “tags,” such as <TITLE> hello </TITLE>, that describe the Web page. XML mirrors HTML in this way by providing a clear text syntax that describes the Web page and the data associated with the page. Because XML deals with both page presentation and data interchange, XML is viewed as more powerful and flexible than HTML.

Language

XML is not a programming language like C++ or Java. Programming languages contain detailed lists of instructions for a computer to perform. By contrast, XML is a standard that allows the creation of different markup language for different purposes. XML isn’t really a language, but rather a metalanguage, or a language that defines the rules by which other languages are governed.

DTDs and Schemas

XML includes built-in support for Document Type Definitions, also called DTDs, which allow for a more formal description of a specific flavor of XML. Because of inherent limitations of DTDs, there is great interest in a more powerful technique called XML Schemas. An XML Schema defines the specific elements and attributes that are valid within an XML document. Additionally, schemas provide rules to ensure that the internal structure of the XML to which they apply remains predictable and consistent. The World Wide Web Consortium is working on a standard for XML schemas. In the meantime, other initiatives such as Microsoft’s BizTalk are filling the gap by providing their own flavor of XML schemas.

XSLT

XSLT, or Extensible Stylesheet language Transformations, are tools for converting one type of XML to another via specially formatted stylesheets (which are themselves XML documents). XSLT processor programs use stylesheets to perform the conversions.

XML vs. SGML and HTML

An XML document is essentially a specially formatted description of a person, place, or thing. In this respect, it resembles both SGML (Standard Generalized Markup Language) and HTML (Hypertext markup Language). However, XML is far less complex and easier to work with than SGML, while also being more powerful and flexible than HTML.

An Example for XML

```
<customerList>
  <customer>
    <name>Robert</name>
    <age>31</age>
    <address>5 Grier PL NE Calgary</address>
    <telephone>210 5674</telephone>
```

```
</customer>
<customer>
  <name>Gary</name>
  <age>45</age>
  <address>150 halkstone NW Calgary</address>
  <telephone>267 7659</telephone>
</customer>
</customerList>
```

2.2 Growing Need for Agent Communication

Agents are generally designed with a specific purpose in mind. They do one or perhaps several tasks very well, but aren't often designed as a jack-of-all-trades. If agents must perform more tasks, we can either increase their complexity (which increases the development effort), or we can make them work co-operatively.

For cooperation between agents to succeed, effective communication is required. We can view a collection of agents that work together co-operatively as a small society, and for any society to function coherently we need a common language and communication medium. Societies can do things that no individual agent can, but diversity introduces heterogeneity. Autonomy encourages disregard for other agents' internal structure while communicating agents need only care about understanding a "common language".

This language and communication medium is critical for cooperation between agents. Imagine how ineffective software agents would be if they worked in isolation - unable to interact with their peers. Certainly we write proprietary protocols and languages, or piggyback requests over other protocols like HTTP - but this adds time and complexity to an agent's development. Unless your communication mechanism is reusable in future projects, this can be a wasteful option, and can lead to systems that aren't interoperable with future systems.

2.3 Why XML is fit for Agent Communication

First of all, the web is definitely a very attractive 'place-to-be' for making real business of agent technology today. The incredible success of HTML has led the WWW consortium to the development of XML [W3C, 1998], a language for data representation that is likely to become a standard for data interoperability in the Internet, due to the advantages it can provide. In fact, XML represents data in a familiar (HTML-like) tagged form, and explicitly separates the treatment of data from its representation. This achieves both the well-appreciated feature of human-readability and the platform-independence required for the Internet. In addition, XML can be made capable of representing whatever kind of data and entity one is likely to find in the Internet: documents, services, and objects, as well as agents. These characteristics let us think that interoperability in the Internet will be information-oriented and based on XML, rather than service-oriented as may happen in architectures based, for instance, on CORBA [OMG, 1997]. Combined with style-sheet languages such as XSL, agent messages can be

represented in common Web pages. Also, its linking capabilities allow splitting up services into constituting components, potentially offered by different providers. Moreover, a wide variety of XML supporting tools exist to allow a fast application development.

3. Application

3.1 XML for E-Commerce Multi-Agent Cooperation

E-Commerce applications operate in a dynamic and distributed environment, dealing with a large number of heterogeneous information sources with evolving contents and autonomous tasks for information search, fusion, extraction and processing, without centralized control. E-Commerce typically involves the following activities: identifying requirements; brokering products; brokering vendors; negotiating deals; or making purchase and payment transactions. We suppose all these activities to be conducted by software agents.

The dynamic nature of E-Commerce requires multi-agent cooperation to be based on dynamic ontology. By dynamic ontology we mean that the concepts, rules and facts underlying agent interaction, are different from domain to domain, and vary from time to time. In order to automate agent cooperation, it is necessary to provide a standard format for encoding messages with meaningful structure and semantics, as well as domain ontology that agents can readily exchange and interpret. This format should be common for agent communication as well as for E-Commerce data exchange in general. We use XML for the above purpose.

Dynamic agents send and receive information through XML encoded messages. We use a KQML/FIPA ACL-like format, encoded in XML. In fact, an XML document is an information container for reusable and customizable components, which can be used by any receiving agent. This is the foundation for document-driven agent cooperation. By making Web accessible to agents with XML, the need for customer interfaces for each consumer and supplier will be eliminated. Agents may use XML format to explain their BDI (beliefs, desires and intentions), explaining new performatives by existing, mutually understood ones. Based on the commonly agreed tags, agents may use different style DTDs to fit the taste of the business units they mediate. Further, a dynamic agent can carry an XML front-end to a database for data exchange, where both queries and answers are XML encoded.

The power of XML, the role of XML in E-Commerce, and even the use of XML for agent communication, have been recognized. Although XML is well structured for encoding semantically meaningful information, it must be based on ontology. As ontology varies from domain to domain, and dynamic for dynamically formed domains, the more significant issue is to exchange the semantics of domain models, and interpret messages differently in different problem domains.

Generally speaking, domain ontology provides a set of concepts, or meta-data, that can be queried, advertised and used to control the behavior of agent cooperation. These concepts can be marked using XML tags, and then a set of commonly agreed tags, underlie message interpretation. The structures and the semantics of the documents used in a particular problem domain are represented by the corresponding DTDs and interpreters.

We use different languages, all in XML format, for different problem domains, such as product ordering, market analysis, etc. Accordingly, we use an individual interpreter for each language. Dynamic agents can exchange those DTDs together with documents, and exchange those DTDs together with documents, and exchange those interpreters as programming objects, in order to understand each other in communication.

This XML approach allows us to provide a unified agent communication mechanism for supporting E-Commerce.

3.2 XML for Supporting Agent Communication of A Distributed Active Data Archive System

In this case study, SARA is an active digital library of multi-spectral remote sensing images of the earth, and provides Web-based on-line access to such images. Active data represents data that is dynamically generated by a scientific experiment, or may be obtained from a sensor or monitoring instrument. Scientists making use of the SARA archive often require integrated access to information combining retrieval, computation, and visualization of multiple images. A large scientific collaboration may generate many queries and the resulting analysis of images can lead to large quantities of data, some of which must be integrated with data from other system such as a Geographic Information System for a given area/region. The data is maintained in different kinds of file system. These requirements have generated an urgent need for a multi-agent system, which comprises both intelligent and mobile agent, to manage and analyze distributed multi-agency remote sensing data. In this system we use XML to encode system structure as metadata and user requests.

When a user launches a query, an XML document is created. Every specific XML specification is based on a separate DTD that defines the name of tags, their structure and content model. While the XML specification contains the structured information, the DTD defines the semantics of that structure, effectively defining the meaning of the XML document. An agent can generate the processing programs representing the XML elements of interest according to the DTD with a parser, and then travel across the Internet to retrieve related information. For example, corresponding to tag TRACK in the XML document, a Java method getTrack can be generated to interpret the meaning of the tag. The resulting data can be returned to the user with an XSL style sheet.

We also use XML as an application-specific transport protocol, to enable agents within the system to communicate with each other. Autonomous agents cooperate by sending messages and using concepts from the SARA ontology – where the ontology describes terms and concepts (such as a Track, a Latitude/Longitude coordinate etc) and their

inter-relationships. We are defining a message, which embeds such ontology, to provide a mechanism for agents to exchange requests and message interpreters. Agents send and receive information through XML encoded messages. Based on pre-defined tags, agents may use different style DTDs to fit different mediation. Moreover, a mobile agent can carry an XML front-end to a remote data source for data exchange, where both queries and answers are XML encoded.

4. Conclusion

Agent technology is becoming more prevalent as the availability of network access, and the demand for the end-uses of agents, becomes greater. Electronic commerce and the Web are playing no small part, posing a challenge for developers to design agents that can enrich the user experience, or make behind-the-scenes work cheaper, faster, or more effective. Agents are being called on to assist users, to assist employees, or even to assist other agents. Therefore, Agent- to-agent communication is key to realizing the potential of the agent paradigm, just as the development of human language was key to the development of human intelligence and societies. Most, but not all, would agree that communication is a requirement for cooperation.

There are many agent frameworks, languages and protocols available today, specifically designed for inter-agent communication. Some solutions are language-specific, while others are more general in nature. Some solutions are designed for simplicity and ease of use, and others offer more expressive communication. Each has its pros and cons, and is suited to different types of agents. Before choosing any agent communication mechanism, be sure to evaluate your needs carefully.

For developers who are getting up to speed with software agents, you may find that the sheer range of solutions is daunting. There are so many choices available, and each solution offers different properties. The following is a checklist of agent communication properties that agent developers should look for.

- Ease of Use
- Support for Text and Binary Data Exchange
- Support for Message Passing/Queries as well as Data Exchange
- Implementations Available for Target Platform

XML meets all the criteria described above as medium of agent communication. XML provides a way of structuring data in human readable form and lends itself well to expressing ontologies. The DTD allows people to define the structural relationships among agreed upon terms. The resulting XML documents are then readable by humans and usable, without modification, by machines. Because of this convenience, a XML is increasing used to agent communication.

References

- [1] [W3C, 1998] WWW Consortium, extensible Markup Language (XML) 1.0, W3C Recommendation, <http://www.w3.org/TR/REC-xml>, 1998
- [2] [OMG, 1997] OMG, “CORBA 2.1 specifications”, 1997, <http://www.omg.org>
- [3] Bart Bauwens, “ XML-based Agent Communication: VPN Provisioning as a Case Study”, <http://www.infoloom.com/gcaconfs/WEB/granada99/bau.HTM>
- [4] Qiming Chen, Meichun Hsu, Umeshwar Dayal, and Martin Griss, “ Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation”, http://www.hpl.hp.com/org/stl/dmsd/publications/qchen_AA2000_short.pdf
- [5] Yanyan Yang, Omer F. Rana, and David W. Walker, “ Towards an XML and agent-based framework for the distributed management and analysis of active data archives”, <http://www.cs.cf.ac.uk/User/David.W.Walker/papers/iawtic2001.pdf>
- [6] Tutorial for “Connect Agent for XML 1.0”, <http://www.cs.cf.ac.uk/User/David.W.Walker/papers/iawtic2001.pdf>