A Workflow-based Electronic Marketplace on the Web *

Asuman Dogac, Ilker Durusoy, Sena Arpinar, Nesime Tatbul,
Pinar Koksal, Ibrahim Cingil and Nazife Dimililer
Software Research and Development Center
Dept. of Computer Eng.
Middle East Technical University
06531, Ankara, Turkey
asuman@srdc.metu.edu.tr

Abstract

In this paper, we describe an architecture for an open marketplace exploiting the workflow technology and the currently emerging data exchage and metadata representation standards on the Web.

In this market architecture electronic commerce is realized through the adaptable workflow templates provided by the marketplace to its users. Having workflow templates for electronic commerce processes results in a component-based architecture where components can be agents (both buying and selling) as well as existing applications invoked by the workflows. Other advantages provided by the workflow technology are forward recovery, detalied logging of the processes through workflow history manager and being able to specify data and control flow among the workflow components.

In the architecture proposed, the resources expose their metadata using Resource Description Framework (RDF) to be accessed by the resource discovery agents and their content through Extensible Markup Language (XML) to be accessed by the selling agents by using Document Object Model (DOM). The common set of Document Type Definitions (DTDs) are used to eliminate the need for an ontology.

The marketplace contains an Intelligent Directory Service (IDS) which makes it possible for agents to find out about each other through a match making mechanism. References to the related Document Type Definitions (DTDs) are stored in IDS. The IDS also contains the template workflows for buying and selling processes.

* This work is partially being supported by the Middle East Technical University, Graduate School of Natural and Applied Sciences, Project Number: AFP-97-07.02.08 and by the Scientific and Technical Research Council of Turkey, Project Number: 197E038

1 Introduction

In this paper, we describe an architecture for an open marketplace exploiting the workflow technology and the currently emerging data exchage and semantic representation standards on the Web.

We have chosen the distribution infrastructure of the marketplace to be Web plus CORBA [6]. The current Web-native distributed object communication proposals like WebBroker [23] do not seem to be mature enough yet and lack the services that CORBA provides. In our implementation the CORBA methods on the server are invoked through XML. In this way a CORBA compliant ORB is not necessary at the client side.

We plan to realize the communication of agents in the system through a KQML implementation based on CORBA. As discussed in [1], CORBA provides both the transport layer and the necessary services in this respect.

In the architecture proposed, the Intelligent Directory Service (IDS) contains the template workflows for buying and selling processes, as well as a match maker mechanism, Document Type Definitions (DTDs) and a dictionary of synonyms. The match maker mechanism lets agents find out about each other. Using the common set of DTDs eliminate the need for an ontology. The dictionary of synonyms is used by the buying processes to help the customer to specify the item s/he wishes to purchase.

In this architecture, resource discovery agents of IDS working in the background discover resources. If a resource is willing to join the marketplace, the IDS sends a selling workflow template to the seller's site. The resource creates a selling process by automatically customizing the template using the information obtained from the resource through a user friendly graphical interface. The selling agent involved is registered with the match maker. However if the resource has a selling agent already defined, that agent is registered. The match maker informs the related buying agents already in the marketplace about this newly created selling

agent.

When a customer wants to buy a service or an item from the marketplace, s/he is provided by a buying workflow template. The first activity invoked registers the buyer to the IDS. The next activity checks the related DTDs in IDS with the item name specified by the buyer. Since the buyer may not know the right term (used in DTD) to use for the item, an intelligent dictionary of synonyms is used. For example, consider a computer shop using a computer DTD in describing its service. If a customer wants to buy a CPU and uses the term "Processor" and if "CPU" is the term used in DTD, then dictionary of synonyms is used to match the word "Processor" with "CPU". The names and types of the properties obtained from the related DTD by the activity are passed to the buying agent. Obtaining the names and types of properties from DTDs is necessary since the buyer may not know in advance all the properties of the item.

The buying workflow activates a buying agent which presents the user a form containing the values or ranges for the properties of the item along with the criteria that the customer wishes to be optimized in the negotiation phase and the required parameters. The buying agent negotiates with the related selling agents to realize the transaction. A comparative analysis of the available alternatives can also be presented to the customer by the buying agent if the customer wishes so.

The paper is organized as follows: Section 2 briefly summarizes the recent data exchange and metadata representation standards for the Web. In Section 3, the distribution and the agent communication infrastructure is presented. Section 4 describes the overall architecture of the marketplace. In Section 5, workflow system architecture used in the system is discussed. Section 6 presents the feasibility of the proposed system. And the conclusions are given in Section 7.

2 Data Exchange and Metadata Representation Formats

In this section, the data exchange and semantic representation formats that are used in the marketplace architecture are discussed.

2.1 Extensible Markup Language (XML) and Document Type Definitions (DTDs)

World Wide Web Consortium's (W3C) Extensible Markup Language (XML) [11] defines a simple subset of SGML (the Standard Generalized Markup Language). Unlike HTML, which defines a fixed set of tags, XML allows the definition of customized markup languages with application specific tags [20]. That is, XML provides support for the representation of data in terms of attribute/value pairs with user defined tags.

XML differs from HTML in three major respects [3]:

1. Information providers can define new tag and attribute names at will.
2. Document structures can be nested to any level of complexity.
3. Any XML document can contain an optional description of its grammar for use by applications that need to perform structural validation.

Document Type Definitions (DTDs) which are defined for user groups provide a formal definition of documents for that group, that is, they define what names can be used for elements, where they may occur and how they all fit together in an XML file.

The XML working group is currently developing a facility [26] that will allow tag names to have a prefix which make them unique and prevent name clashes when developing documents that mix elements from different schemas. XML namespace proposal provides a way to define relationship between the use of particular element name in a document and the standard vocabulary (DTD) from which the name is taken.

2.2 Resource Description Framework (RDF)

An emerging solution to letting automated agents surf the Web is to provide a mechanism which allows a more precise description of the resources that are available on the Web [17]. The Resource Description Framework (RDF) [22] by the World Wide Web Consortium (W3C) is a standard for metadata that provides interoperability between applications that exchange machine-understandable information on the Web.

RDF [22] defines both a data model for representing RDF metadata, and an XML-based syntax for expressing and transporting metadata. RDF is a model for representing named properties and their values. These properties serve both to represent attributes of resources and to represent relationship between resources.

RDF will provide the much needed information for the agent technologies working on the Web. Agents can use RDF for describing their capabilities and negotiating the terminologies used in communication.

2.3 Document Object Model (DOM)

W3C's Document Object Model (DOM) [10] defines an object-oriented API for HTML and XML documents which a Web client can present to programs that need to process the documents [20]. DOM represents a document as a hierarchy of objects with proper inheritance relationship among them, called nodes, which are derived (by parsing) from a source representation of a

document (HTML or XML). In other words, the DOM object classes represent generic components of a document, and hence define a document object meta model. The representation of a Web page in terms of objects makes it easy to associate code with the various subcomponents of the page. For example, Document object has a "documentType" method which returns DTD for XML documents (and "null" for HTML and XML documents without DTDs) and a "getElementsByTagName" method which produces an enumerator that iterates over all Element nodes within the document whose "tagName" matches the input name provided. Thus DOM provides a general means for applications to access and traverse documents written in HTML and XML without having themselves to perform complex parsing.

3 Distribution and Communication Infrastructure

CORBA together with the Web is used as the distribution infrastructure. We have chosen to invoke methods in CORBA through XML. In this way a CORBA compliant ORB is not necessary at the client side. The client can send the method invocation requests and its parameters expressed in XML by "post" method of the HTTP protocol. An HTTP server receiving this request can pass it to a CORBA server through CGI. CORBA server invokes the method after parsing the XML document. If there is an ORB at the client side, the client sends the request directly to the CORBA server again in XML.

In this way it is also possible to replace the distribution infrastructure by a Web-native ORB in the future when the Web-native ORBs provide CORBA-like capabilities.

A Web-native distributed object communication architecture, namely WebBroker, is suggested in [23]. In this architecture HTTP is used as the transport protocol, XML is used as the syntax and URIs as the addresses. The work proposes two DTDs, one to define the message contents; other for describing interfaces on the objects. This approach does not yet have any mechanisms similar to OMG's Common Object Services.

The choice of CORBA also makes it possible to use Trading Object Service for the match maker mechanism. The OMG Trading Object Service [21] facilitates the offering and the discovery of instances of services of particular types. A trader is an object that supports the Trading Object Service in a distributed environment. It can be viewed as an object through which other objects can advertise their capabilities and match their needs against advertised capabilities. Advertising a capability or offering a service is called "export". Matching against needs or discovering services is called "import". Export and import facilitate dynamic discovery of, and late binding to, services.

CORBA itself is not enough to provide the communication among agents. Agents need to speak a common language with a set of known typed messages understood by all of them. We have chosen KQML [16] as the communication language. CORBA seems to offer an efficient medium for implementing KQML since it provides the transport protocol and the services (like naming, trading, event services) to implement the networking performatives, facilitators and domain agents of KQML [1].

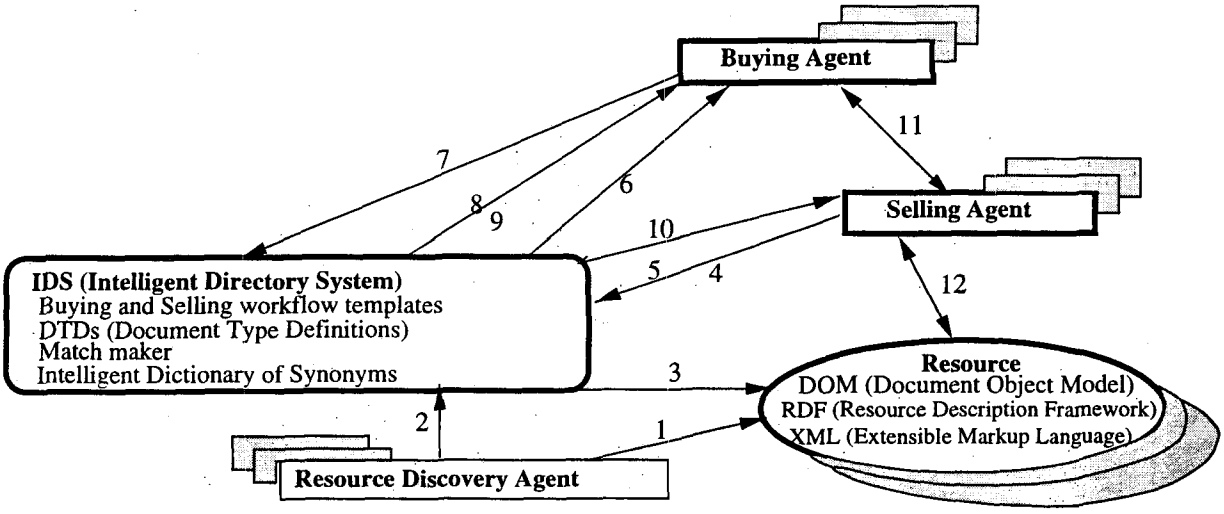
4 Architecture of the Marketplace

The electronic commerce in the proposed marketplace is realized in five phases as shown in Figure 1. Resource Discovery Agents finds out resources on the Web and the selling process templates for the registered resources are created in Phase I. In Phase II a buying process template is created for the customer that operates in the marketplace to make a purchase. In Phase III, the buying agents contact with all possible selling agents to determine the ones that can provide the item with required properties. Phase IV contains the negotiation between buying and selling agents. Finally, in Phase V, the control is returned back to the related workflows.

PHASE I: The resource discovery agents working in the background find out about the resources providing products and services. We expect resources to expose their semantics by using the Resource Description Framework (RDF) [18, 17] and the Extensible Markup Language (XML) [11]. As briefly summarized in Section 2.2, RDF defines both a data model for representing RDF metadata, and an XML-based syntax for expressing and transporting the metadata. Since resources use RDF to expose their metadata, the resource discovery agents do not need intelligence in extracting information from the resources. However, they do have other properties of agents like being autonomous, reactive and proactive.

Resource Discovery Agents contact the owner of the resource by using the means (possibly an e-mail address) available on the resource. If the resource wants to join the marketplace, the owner of the resource accesses to the marketplace's web site through a web browser and marketplace provides it with a template workflow of a selling process.

The selling workflow template contains the basic activities involved in the selling process (like obtaining the product information from the resource, activating the selling agent, shipment) as well as compensa-



Phase I: 1. Resource Discovery Agents (RDA) find out about resources using RDF
2. RDA informs of the resource to the IDS
3. The resource is provided with a selling workflow template (with a selling agent)
4. If the resource has already a selling agent, this is registered with the match maker
5. Else a selling agent is provided by the IDS and is registered with the match maker

Phase II: 6. Buying workflow template (with an agent) is provided to the customer
7. Buying workflow invokes an application to pass the item name to the IDS
8. IDS provides a form to the buying agent using the related DTD

Phase III: 9. IDS provides the OIDs of the matching selling agents
10. IDS provides the OIDs of the matching buying agents

Phase IV: 11. Buying agents and selling agents go into direct negotiation through KQML
12. Selling agents access the resources through DOM

Phase V: The rest of the activities specified in the workflows are executed

Figure 1: The Architecture of METU-EMar

tion activities and a selling agent. The workflow process has been designed to accomodate ad-hoc specification of further activities. In this way it is possible for the resource owner to specify the additional activities according to his requirements. As an example the shipment process can be delegated to a subprocess or an agent available.

The owner of the resource (user) customize the workflow template provided by the IDS. The first activity invoked presents a form to be filled in that contains information automaticaly extracted from the resource (items or services provided, their prices) and request information from the user (owner of the resource) about the negotiation policy and the desired user interaction characteristics like how and when the user wants to be informed about the progress of the negotiation. This information is passed to the selling agent the agent is registered with the match maker. If the resource already has a selling agent, this one is registered.

PHASE II: When a customer wants to make a purchase on the marketplace, s/he connects to the marketplace's web site through a web browser and obtains a workflow template. The first activity invoked requests from the buyer the name of the service or the product s/he wishes to purchase from the marketplace. This activity accesses to the related Document Type Definitions (DTDs) contained in the IDS to obtain names and types of properties of the product. DTDs which are defined for customer groups provide a formal definition of documents for that group, that is, DTDs define what names can be used for elements, where they may occur and how they all fit together in an XML file as described in Section 2.1. In our case, all the merchants use the same definition in their DTDs for the item accessed by the selling agent. Therefore, there is no need for a translation among terminologies (which is necessary when XML files have different DTDs and different customers define their own ways of using attribute/value pairs to represent the same information).

Different names can be provided for the same product by the customers, in other words, the customers may not know the standard terms used in DTDs. Therefore, a dictionary of synonyms is necessary in the IDS. This dictionary of synonyms may be implemented to contain some intelligence in the sense that whenever an item or service is not found in the dictionary, the customer may be asked to provide synonyms and these terms can be added to the dictionary for later use. The application returns the names and types of the properties of the item and a list of related selling agents to the buyer's site. The customer (user) using the standard form that is presented to him specifies the desired property values of the item, and the negotiation strategy. Using this information the buying agent is automatically created and registered with the match maker.

PHASE III: A buying agent contacts all the related selling agents using the list provided by the IDS in Phase II and asks them if they can provide the item with the desired properties and conditions. For example the buying agent which wants to buy a "second hand domestic car" has the list of all the selling agents that sell cars. But only some of them may have the car with desired properties ("second hand" and "domestic").

PHASE IV: The buying agents go in a direct negotiation with selling agents by sending and receiving proposals [24]. XML is used in encoding resources and KQML [16, 12] is used as the agent communication language among agents.

The buying and selling agents in the marketplace act autonomously, that is, once released in the marketplace, they negotiate and make decisions on their own, without requiring customer intervention. They are proactive in contacting the other interested agents and reactive to the changes in the marketplace like new agents.

The resources should provide semantic information about their content to the selling agent. In this respect, the resources should be defined in XML. DOM is used by the selling agents in processing XML pages to obtain specific product data, like the price of the product. The selling agents should be authorized to invoke certain applications at the resource to obtain the bargaining strategy and its parameters which implies that the resources should provide this information through a standard interface.

The work on automated negotiation within electronic commerce is an active research area. An electronic marketplace necessitates multi attribute, multi party negotiation. Multi-Attribute Utility Theory in conjunction with Distributed Constraint Satisfaction have been used for this purpose [13, 25]. There is need for further work in this area [2].

PHASE V: This phase starts after the successful completion of negotiation among agents. In this phase the other activities specified in the related workflows like shipment and payment are executed.

5 Electronic Commerce Workflows

In our system, the templates of buying and selling workflows exist in the IDS and are provided to buyers and sellers to automatically reconfigure their processes. The workflow technology provides for easy and controlled restructuring and dynamic modification of involved activities and agents which are critical for customization and adaptation of underlying processes.

Workflow approach allows buyers and sellers to define their own tasks and to invoke their already existing applications within the workflow and to re-structure the control and data flow between the tasks, in other words, to automatically create a custom built workflow from the workflow template. The higher level of abstraction provided by the workflow technology makes this customization for different users possible. The further advantages brought by the workflow technology are making the component-based architecture possible and also providing forward recovery and detailed logging of the processes.

Our market architecture necessitates the customization of workflow both at compile time and at run time. At compile time, according to the needs of the user, the particular activities to be invoked and their control flow is decided. For example the user may not wish to use an automated payment process in which case the payment activity is not included in the workflow specification generated for that customer.

Some of the tasks in the process can not be defined from the beginning (i.e., in compile time). Instead they are created and organized dynamically depending on the interaction characteristics of a user or exceptions that occur. Therefore the structure and control flow between the tasks must be defined dynamically and should be modified on the fly. These features require workflow technology which allow for run time modifications [15].

The added advantage of using workflow technology in realizing the electronic commerce process is the recovery facility provided. As an example, a selling process may have committed a payment activity after which the shipment activity fails. In this case the selling agent can be automatically rolled back by the system by executing a compensation activity for the payment.

5.1 Workflow Implementation

We have developed a workflow system architecture to support the market technology described. The workflow has componentized architecture and uses the block-structured specification language described in [8]. Blocks can be nested and the top level block represents the workflow process.

When it comes to the scheduling policy of the workflow, the scheduling should be distributed to avoid centalized scheduler bottleneck where the whole workflow comes to a halt when the centralized scheduling site fails. In distributed scheduling there are several schedulers running on different nodes of the network each executing a part of a process instance so that the execution can continue even when some of the sites fail. As an example if the site where the negotiation is handled fails after sending the appropriate messages to the rest of the sites, the workflow can successfully keep on scheduling and executing the activities like payment or shipment.

Yet when it comes to run time modification of a workflow with a distributed scheduling mechanism, finding out about the pieces of a process instance and exerting control on them becomes very inefficient.

As a solution to this problem we propose a component based adaptable workflow system architecture [5] where each process instance is an object that contains all the necessary data and control information as well as its execution history. This feature makes it possible to dynamically modify the process definition on instance basis at run time. Furthermore, it is possible to migrate the object in the network to provide load balancing. It should be noted that with this architecture, a site failure effects only the process instances running on that site.

6 Feasibility

The technological requirement of the architecture proposed is the semantic interoperability of the Web resources. The building blocks for this, although have already been defined or are being defined mostly as standards, are at their infancy. For example, work is underway to define XML-based data exchange formats in both the chemical and the health care communities. A number of industry groups defined SGML DTDs for their documents (e.g. the US Defense Department, which requires much of its documentation to be submitted according to SGML DTDs)[20]. A large US project aims to define specific property names for specific elements in computer industry that can possibly be implemented through XML DTDs [7].

The architecture we describe requires the DTDs for the user groups to be available. Note that since RDF assertions use properties defined in the schemas, i.e., DTDs, the use of RDF also depends on the availability of standard DTDs. Until the standard DTDs become available and the RDFs start using these schemas, there is a need for the following modifications in the architecture in realizing the proposed scenario:

1. The resource discovery agents utilize machine understandable information (RDF) and therefore can not be implemented easily when the standard vocabulary (DTDs) used by RDF is not available. In this case, resource discovery agents should either be more intelligent or include heuristic techniques to understand the content of the resources.

2. When XML files have different DTDs (i.e., different users define their own ways of using attribute/value pairs to represent the same information), there is a need for a mechanism to identify associations among the terminologies of the XML files. This can be achieved through a translation mechanism between terminologies. This translation is also needed in the negotiation phase among the buying and the selling agents.

7 Conclusions

It is clear that in a marketplace as large as the one provided by the Web, the service provided by the proposed architecture is invaluable. It will not only help to locate better opportunities for both the buyers and the sellers but it will also save a lot of their time in negotiations. In other words, the proposed marketplace aims to find the best conditions for its clients and help to overcome the limitations of direct communications between customers and suppliers. The marketplace enables the customers to reach various suppliers whose existence they are unaware of and hence it would be impossible for them to reach otherwise. Symmetrically, the marketplace also gives the suppliers the chance to contact to a much wider range of customers.

The advantages of the proposed architecture are that it considers the commerce process as a workflow and exploits the newly emerging interoperability standards on the Web.

References

[1] Benech, D., Deesprats, T., Raynaud, Y., "A KQML-CORBA based Architecture for Intelligent Agent Communication in Cooperative Service and Network Management", in Proc. of the First Intl. Conf. on Management of Multimedia Networks and Services, 1997.

[2] Bichler, M., Beam, C., Segev, A., "Offer: A Broker-centered Object Framework for Electronic Requisi-

tioning", in Proc. of the Intl. IFIP Working Conference: Trends in Electronic Commerce, Hamburg, Germany, June 1998.

[3] Bosak, J., "XML, Java, and the Future of the Web", http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.html.

[4] Chavez, A., Maes, P., "Kasbah: An Agent Marketplace for Buying and Selling Goods", in Proc. of the First Intl. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996, http://agents.www.media.mit.edu:80/groups/agents /Publications/kasbah-paam96.ps.

[5] Cingil, I., Koksal, P., Dogac, A., "A Component-based Adaptable Workflow System Architecture on the Internet for Electronic Commerce", Technical Report, Software R&D Center, METU, July 1998.

[6] *The Common Object Request Broker: Architecture and Specification.* OMG Document Number 93.12.43, December 1993.

[7] Danish, S., Personal Communication.

[8] Dogac, A., Gokkoca, E., Arpinar, S., Koksal, P., Cingil, I., Arpinar, B., Tatbul, N., Karagoz, P., Halici, U., Altinel, M., "Design and Implementation of a Distributed Workflow Management System: METUFlow", in [9].

[9] Dogac, A., Kalinichenko, L., Ozsu, T., Sheth, A., (Edtrs.), "Advances in Workflow Management Systems and Interoperability", Springer-Verlag, 1998.

[10] Document Object Model (DOM), http://www.w3.-org/DOM/.

[11] Extensible Markup Language (XML), http://-www.w3.org/XML/.

[12] Finin, T., Labrou, Y., Mayfield, J., "KQML as an agent communication language", in Jeffery M. Bradshaw, editor, Software Agents, MIT Press, 1995.

[13] Guttman, R., Maes, P., "Agent-mediated Integrative Negotiation for Retail Electronic Commerce." in Proc. of the Workshop on Agent Mediated Electronic Trading (AMET'98). May 1998.

[14] Koksal, P., Arpinar, S., Dogac, A., "Workflow History Management", ACM Sigmod Record, Vol. 27, No. 1, March 1998.

[15] Koksal, P., Cingil, I., Dogac, A., "Adaptable Workflow Systems", Technical Report, Software R&D Center, METU, July 1998.

[16] Labrou, Y., Finin, T., "A Proposal for a new KQML Specification", Report TR-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County. Available on-line as http://www.cs.umbc.edu/-~jklabrou/publications/tr9703.ps.

[17] Lassila, O., "RDF Metadata and Agent Architectures", http://www.objs.com/workshops -/ws9801/papers/paper056.html.

[18] Lassila, O., Swick, R. R. "Resource Description Framework (RDF) Model and Syntax", Working Draft, World Wide Web Consortium. Available online as http://www.w3.org/TR/WD-rdf-syntax/.

[19] Manola, F., "Towards a Web Object Model", http://www.objs. com/OSA/wom.htm.

[20] Manola, F., "Towards a Richer Web Object Model", ACM Sigmod Record, Vol. 27, No. 1, March 1998.

[21] OMG's Trading Object Service. OMG Document orbos/96-05-06, Version 1.0.0, May 10, 1996.

[22] Resource Description Framework (RDF), http://-www.w3. org/Metadata/RDF/.

[23] Tigue, J., Lavinder, J., "WebBroker: Distributed Object Communication on the Web", http://www.w3.org/TR/1998/ NOTE-webbroker-19980511.

[24] Utkun, G., "An Agent Architecture for an Electronic Marketplace", M. Sc. Thesis, Dept. of Computer Eng., Middle East Technical University, Sept. 1998.

[25] Woolridge, M., Jennings, N, "Intelligent Agents-Theory and Practice", Knowledge Engineering Journal, June 1995.

[26] XML Name-space facility, http://www.w3.org/TR /1998/ NOTE-xml-names-0119.