RosettaNet-based Implementation for a Collaborative Continuous Replenishment Planning Model Utilizing Web Services and Ontologies

J. Gialelis[1], A.P. Kalogeras[2], A. Kaklis[2], and S. Koubias[1]

[1]Applied Electronics Laboratory
Electrical and Computer Engineering Dept
University of Patras Campus
26500 Rio Patras, Greece
{gialelis, koubias}@ee.upatras.gr

[2]Industrial Systems Institute
Science Park of Patras Building
Stadiou Str, 26506 Platani Patras, Greece
{kalogeras, kaklis}@isi.gr

Abstract

Collaborative e-commerce provides an increased level of flexibility and dynamicity and generates an entrepreneurial competitive edge. Widely accepted implementation frameworks like RosettaNet may be utilized to provide generic inter-enterprise B2B infrastructure. This paper presents an innovative approach towards utilizing RosettaNet in combination with Web Services and Ontologies for the enforcement of the Continuous Replenishment Planning Model.

1. Introduction

B2B interaction frameworks provide the necessary building blocks for B2B applications. Such blocks comprise the necessary modules for: (a) the specification and deployment of inter-enterprise (public) and intra-enterprise (private) processes, (b) the integration of these processes and (c) the required real time interaction with back-end legacy systems, such as ERP (Enterprise Resource Planning), MES (Manufacturing Execution Systems) and SCADA (Supervisory Control And Data Acquisition) systems. [1] Given that advanced B2B applications mandate the utilization/engineering of the interactions of different, complex and heterogeneous applications/systems of diverse enterprises, interoperability is a key issue. [2]

Interactions between the "public" modules of collaborating business partners may be carried out based on specific emerging B2B standards. B2B standards may be considered as business protocols handing inter-enterprise processes. They define the structure and the semantics of the exchanged messages, the bindings to the underlying communication means / protocols, the high level bilateral conversations and the security mechanisms.

In this context, several standards and languages have been developed aiming at supporting the interchange of information among "public" modules. Dominant generic standards addressing horizontal as well as vertical integration issues are the following: EDI (Electronic Data Interchange) [3], EDIINT(EDI over the Internet) [4], ebXML (e-business eXtended Markup Language) [5] and BizTalk [6]. Furthermore, standards such as RosettaNet [7], IOTP (Internet Open Trading Protocol) [8], ICE (Information and Content Exchange) [9] and OBI (Open Buying on the Internet) [10] are considered more vertical market-oriented.

This paper presents an architecture for dealing with collaborative processes between enterprise partners. In section 2 a business model governing enterprise collaboration is presented. Section 3 designates the required architectural elements for the business model implementation. Section 4 depicts a use case implementation. Finally, conclusions are given in section 5.

2. Business Model

A business model presents the necessary methodology that should be used in order to do business with an enterprise and defines an architecture for the deployment of enterprise products/services and the associated information flows, including a description of the various business actors (suppliers, customers) and their roles, a description of the potential benefits for the various actors and a description of the sources of the revenues [11]. Acquiring knowledge of the business model of an enterprise is critical in order to understand and derive the corresponding business processes and their requirements. Given the existence of electronic links between the different business actors (manufacturers, suppliers, customers) and the increasing importance of such links in the enforcement of the enterprise business model, relevant B2B e-commerce applications, providing enterprises with a competitive advantage, have been rapidly growing around the globe.

0-7803-9402-X/05/$20.00 © 2005 IEEE

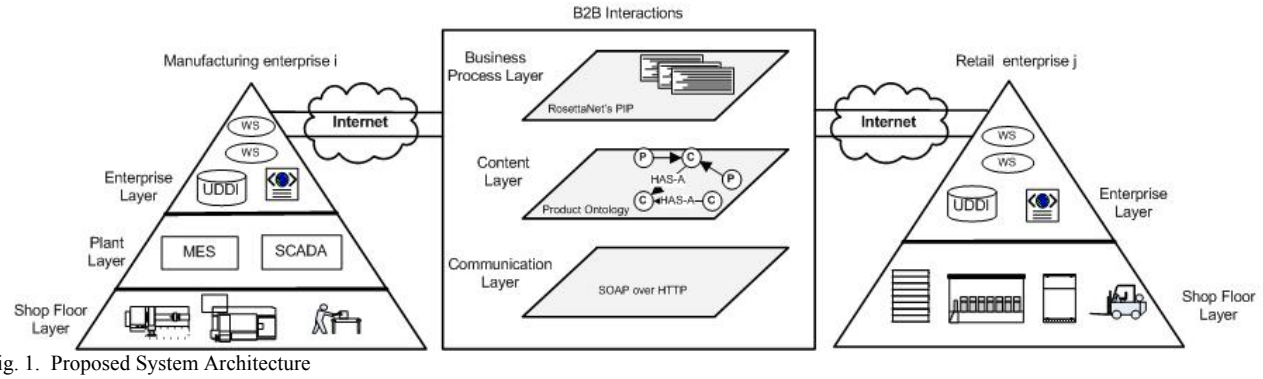


Fig. 1. Proposed System Architecture

In order to best exploit and gain the most out of the adoption of electronic commerce networks, enterprises should establish collaborative B2B e-commerce processes with cooperating manufacturing enterprises. We consider as a collaborative B2B process any end-to-end business process, which results from the seamless integration of intra-enterprise processes with inter-enterprise processes. A designated example of a collaborative B2B process is the Continuous Replenishment Process (CRP) on top of which one of the most promising inventory management functions may be built. This function, called continuous replenishment planning, is defined as the practice of partnering between distribution channel members that changes the traditional replenishment process from a warehouse/distributor centric process handling purchase orders, to an end-user/retailer centric replenishment process involving the manufacturing enterprise in the overall process utilizing actual and forecast data [12].

More specifically, by deploying the CRP, retail enterprises do not have to place orders to manufacturing enterprises taking into account the outcome of independent warehouse inventory algorithms. Relevant information concerning real-time sales and inventory specific data is rather transmitted, utilizing the B2B network, to the specific collaborating manufacturing entity. Using this data, manufacturing enterprises determine the needed product quantity and the appropriate shipping timing to maintain the desirable product amount at retailers' premises.

Some of the major advantages which are accrued by deploying CRP are the following: (a) inventories and stock outs at retail enterprise warehouses are minimized, (b) manufacturing enterprises achieve greater productivity levels through better management of production, (c) significant alteration of interdependency between enterprises is achieved and (d) the Shop-in-Shop methodology is enforced, making the participating manufacturing enterprises to gain a competitive edge. Collateral benefits include improved warehouse productivity, reduced transportation costs, minimal product diversion from location to location and smaller damaged product quantities.

Although, CRP has the potential to alter significantly inventory management practices and make the enterprises "enjoy" the mentioned benefits, survey results do not keep up with these expectations. [12], [13] This drawback is caused, mainly, due to CRP deployment strategies. These strategies are based on point-to-point architectures and on the utilization of proprietary solutions/technologies, thus, presenting limited flexibility to system/applications alterations/upgrading and minimum scalability to value-network expansion.

In order to overcome these drawbacks, we propose a generic deployment approach utilizing the ubiquity of the Internet and the emergence of standard technologies that allow the creation of an infrastructure that makes possible the integration of heterogeneous systems/applications at whatever layer in the enterprises they reside in and the unlimited expansion of the value-network. Our work presented in this paper is in alignment with the Service oriented Architecture (SoA) architectural style and is based on Rosetta Net B2B interaction standard for the satisfaction of business process requirements, ontologies to standardize content semantics and the utilization of SOAP [14] as communications protocols. Figure 1 depicts the overall system architecture.

3. Proposed Architectural Elements

B2B applications interactions are graded according to a three-layer model, which comprises three layers: the communication, the content and the business process layers [15]. The communication layer provides the appropriate communication means for exchanging the required messages among the business partners. The content layer provides the necessary tools (languages and models) so that the involved parties comprehend the semantics of contents and the types of the exchanged documents. The business process layer provides the suitable tools for the establishment of high-level (executive) arrangements and agreements prior to conducting electronic business. The next three sections present the proposed solutions for the three aforementioned layers.

3.1. Communication Layer

Our work utilizes the state-of-the-art technology of Web Services (WSs) [16] in order to open up enterprise systems and make available their functionalities. According to the Web Service architecture, the Simple Object Access Protocol (SOAP) is utilized at the communication layer. In this section we provide an overall presentation of the WS technology analyzing its constituent elements.

Web Services represent a technology for distributed computing, implementing a new paradigm of reusable software building blocks that are URL addressable, which expose application functionalities via open standard application interfaces and protocols. WS invocation makes the integration of the aforementioned functionalities easy and straightforward. Loose coupling of the capabilities provided by a WS leads to achieving the benefits of integration without incurring the difficulties thereof.

WSs promote implementation independence, since they expose client application capabilities rather than implementations. Their benefits include: reduction of application development costs, integration of both data and business processes, reduction of errors, simplification of application maintenance and significant reduction time-to-market.

The breakthrough of the WS technology is the anybody-to-anybody communication easiness. Former distributed computing application environments, such as CORBA [17], EJB[18] and DCOM [19], also aimed at providing a distributed computing environment across heterogeneous environments, but lacked in terms of deployment easiness, mainly regarding the anybody-to-anybody communication.

The key enabler for the WSs technology is XML and the initiatives that form its foundations are SOAP, Web Services Description Language (WSDL) [20], and Universal Description Discovery & Integration (UDDI) [21].

Simple Object Access Protocol (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It is not tied to any particular transport protocol, although HTTP is the most popular. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics. This means that a client and a server can communicate as long as they can forward and understand SOAP messages. SOAP consists of the following parts: an envelope that describes the content of the message and how to process it, a set of encoding rules in order to express instances, application-defined data types and a convention for representing remote procedure calls and responses.

WSDL provides a model and an XML format for describing Web services, enabling the separation of the description of the abstract functionality offered by a service from its concrete details, such as "how" and "where" that functionality is offered. WSDL describes a Web service in two fundamental stages: an abstract one and a concrete one. The WS abstract description provides the messages that it sends and it receives. At the concrete level, a binding specifies transport and wire format details for one or more interfaces. An endpoint associates a network address with a binding. Finally, a service groups together endpoints that implement a common interface.

The last piece of the puzzle is UDDI that supports both design-time and run-time discovery of Web Services.

3.2. Content Layer

Collaboration between business entities utilizing Internet ubiquity requires a strong foundation of commonly shared knowledge. Without such shared common knowledge, participants in a collaboration construct will not be able to decode exchanged information.

An adopted solution to overcome this drawback is the development of fixed pre-defined schemas on which the collaborative entities have agreed upon. The adoption of XML in combination with pre-defined schemas emerged in the business world as a standard representation format, enabling the definition, transmission, validation and interpretation of information. This solution, which is usually adequate for businesses deploying point-to-point B2B processes (one-to-one relationship where the buyer is used to supplier’s terminology), has the disadvantage of not providing any semantics to the information content. Even style sheet transformations are based on the syntax and do not process the semantics contained in XML documents [22].

Consequently, in spite of partially inducing interoperability, the adoption of such approaches reduces flexibility and functional scalability of collaborative activities, since it would be difficult to alter bilateral pre-defined schemas in a multiple partner collaborative environment. This becomes obvious in dynamic electronic commerce environments implementing for instance RosettaNet’s Partner Interface Processes (PIPs) of the Continuous Inventory Allocation and Replenishment (CRP) process using Web Services. In such environments and in order to overcome the above drawbacks, distributed collaborating businesses require common representation of their internal knowledge inducing full interoperability among systems/applications.

Product common terminology is an issue, since businesses describe the same products in different ways. To overcome this constraint the content of the exchanged information must be modeled or annotated. So far, many different ways exist for product description and classification. Structuring and standardizing the product descriptions is a significant task in B2B e-commerce [23]. Ontologies provide a standard means to model knowledge, thus defining the necessary semantics to annotate / model the content of the exchanged information during the CRP process.

We propose the development of product ontologies in order to give a uniform product terminology. For this purpose, existing work in different standards, such as RosettaNet Technical Dictionary (RNTD), may be utilized wherever applicable. RNTD provides common properties for defining products for RosettaNet Partner Interface Processes (PIPs). This dictionary, coupled with the RosettaNet Business Dictionary, provides a common vocabulary for conducting e-business, eliminating confusion in the CRP process stemming from enterprise proprietary terminology. According to our approach there is a two-way one-to-one relationship between product ontologies and RNTDs. Product ontologies can thus be appropriately transformed to RNTDs so that the RosettaNet framework is successfully applied.

3.3. Business Process Layer

Our work focuses on RosettaNet framework in order to address business process layer needs. RosettaNet is an initiative aiming at creating, implementing and promoting e-business process standards. RosettaNet goals include improvements in supply-chain efficiency, increasing supply-chain visibility, and enhancing collaboration in a secure and reliable fashion across trading networks.

RosettaNet focuses on standardizing and automating public processes – it does not attempt to standardize private processes. This is achieved by standardizing the sequence of operations in a business process, the documents themselves, and the messaging system that it sends and receives these documents.

RosettaNet focuses on three key areas of standardization to automate B2B interactions. First of all, the vocabulary needs to be aligned. Secondly, the way in which Business Messages are wrapped and transported must be specified. Finally, the business process governing the interchange of the Business Messages themselves must be harmonized and specified.

Regarding the first aspect, the RosettaNet Business Dictionary contains vocabulary that can be used to describe business properties (e.g., business name, address). The RosettaNet Technical Dictionary contains properties that can be used to describe characteristics of products and services (e.g., purchase order).

With reference to the second, the RosettaNet Implementation Framework (RNIF) specifies the content of messages, transport protocols for communication, and common security mechanism.

The third aspect is handled via RosettaNet's PIPs (Partner Interface Processes). A PIP is a pre-defined XML-based conversation. A conversation consists of a set of business documents (e.g. purchase order, product quote) and message exchange logic (e.g. the sequence of the actions that take place during a product quote request). A PIP is defined using a combination of textual and graphical (UML-cased state machine) representations. A PIP defines the following:

- Roles for the trading partners that use the specific business process.
- Business activities required for business process definition.
- The messages exchanged between the roles, which are called "action" messages. The PIP defines the content and format of the action messages in XML.
- The sequence in which these messages are exchanged and the quality of service attributes for the message exchanges.

For the successful adoption of RosettaNet across a business collaborative network, a key requirement is that the solution is capable of implementing and exchanging business information using RosettaNet PIPs, in a way that complies with RosettaNet specifications and moreover having public process components standardized in machine-readable languages. For this purpose, RosettaNet proposes the creation of PIPs based on Document Type Definition (DTD). Moreover the recent move from DTD-based PIPs to XML Schema-based PIPs is one more step to increase compliance automation. Our approach introduces ontologies as defined in section 3.2., enabling both machine-readability, eliminating human intervention and drastically increasing compliance automation.

4. Example Use Case

The present section of this paper depicts an example case study in order to further elaborate the proposed architectural elements. A specific architecture is presented and implementation details regarding a specific retailer-manufacturer collaborative scenario are depicted.

4.1. Use Case Scenario

The scenario chosen in order to showcase the above architecture is related to the Continuous Replenishment

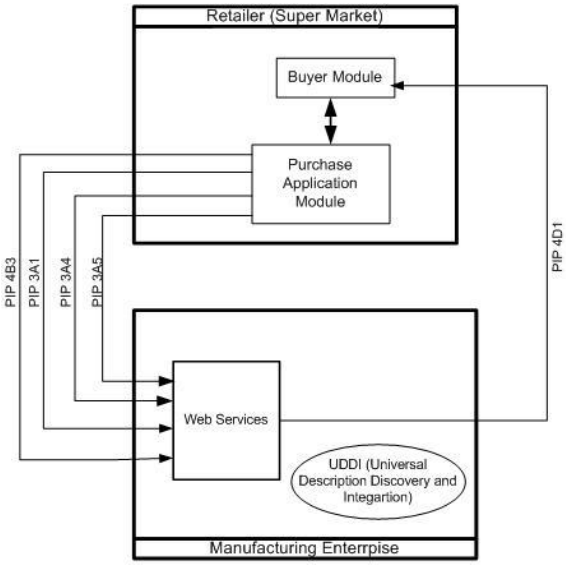


Fig. 2. Use Case Scenario

Process between two collaborating enterprises: a manufacturing enterprise and a retailer (Figure 2).

As mentioned in section 2 the proposed continuous replenishment business model differs from the traditional replenishment model substantially, since it associates replenishment with actual consumption data, which in conjunction to forecasted product demand increases the level of automation regarding order generation between the collaborating partners systems/applications.

The first step in the process involves the notification of the manufacturing enterprise about the consumption of the specific product at the retailer side. This data, which explicitly leads to the estimation of the retailer inventory status, feeds the relevant inventory algorithm at the manufacturer side. This algorithm takes also into account high-level managerial information, such as possible advertising campaign launches, seasonal forecasts as well as international market conditions, in order to generate an advisory signal to the appropriate retailer application/system stating the need for inventory replenishment.

This advisory flag initiates the replenishment process at the retailer side. Based on the established business bilateral partnership rules, and taking into account current quotes, the inventory replenishment algorithm at the retailer side places the necessary orders.

4.2. Use Case Scenario Implementation

The above scenario has been deployed between two specific enterprises: a food industry and a large supermarket chain. The implementation is based on RosettaNet B2B framework.

The appropriate RosettaNet PIPs are utilized and more specifically PIP 4B3 - Notification of

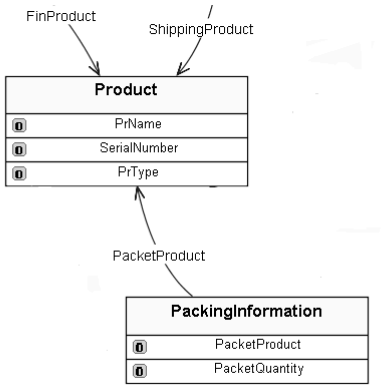


Fig. 3. Part of the Product Ontology

Consumption, PIP 4D1- Notification of Material Release, PIP 3A1- Request Quote, PIP 3A4 - Request Purchase Order and PIP 3A5- Query Order Status.

PIP 4B3 governs the notification of the manufacturer of the retailer side consumption. PIP 4D1 steps implement the advisory signaling to the retailer. PIP 3A1 is used by the retailer in order to get current manufacturer quotes. An order is placed to the manufacturer via PIP 3A4. Finally, the retailer gets information regarding the placed order status via PIP 3A5.

In the subsequent sections, parts of the implementation work are presented for reasons of paper space economy.

4.3. Product Ontology

As aforementioned, product ontologies and RNTDs have a one-to-one relationship. An appropriate transformation is needed in order to produce suitable RNTDs from product ontologies. For the purposes of our implementation work presented in this paper, a product ontology related to the food industry involved has been deployed. Part of this ontology is shown in Figure 3. Further work will make the appropriate transformation in order to produce the needed RNTD for the application of the RosettaNet framework.

Our approach combining RNTDs and ontologies, provides the needed product categorization and classification. Focusing on RNTDs, classes of products with their properties are described in XML DTD or Schema, that is, a collection of XML tags is associated with each product.

Product identification in RosettaNet, ia accomplished via the use of the Global Trade Item Number (GTIN). Product information details can be obtained by querying a supply chain partner's catalog by using the standard tags through “PIP2A5/EC Query Technical Information” in order to return one or more GTINs along with product data. Hence RNTD is used in associating the product data with GTINs. To implement the Technical Dictionary, an organization

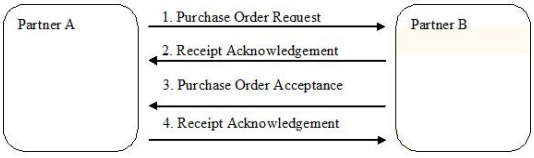


Fig. 4. PIP 3A4 Choreography

must categorize all saleable products according to the product classes and class properties specified in the Technical Dictionary.

4.4. Web Services Description

This section gives implementation details regarding the use case needed Web Services. The Web Services we are describing are related to the Replenishment process which is based as mentioned above on the RosettaNet PIPs 4B3, 4D1, 3A1, 3A4 and 3A5.

For reasons of paper size economy we subsequently present the Purchase Order Web Service. The associated choreography, leading to increased Web Service robustness is shown in Figure 4. Choreography is an extremely important aspect of an e-business dialogue, since it identifies the partners involved, and describes the sequence and content of the exchanges between them.

The required XML Schema definitions for the presented Web Service, are partly shown in Figure 5. The implemented Scheme for our use case example utilizes the RosettaNet Business Dictionary (RNBD), for acquiring the RosettaNet provided common set of PIP properties, as well as direct product definition stemming from the proposed ontology, since existing RNTD do not cover food industry products.

Using existing schema technology and tools, further restrictions can be added to the XML Schema definitions. Additional restrictions can then be applied via tooling during subsequent layers of processing. Thus, partner requirements can be easily altered and appropriately met, increasing the overall system flexibility.

```
<xsd:element
name="Pip3A4PurchaseOrderRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="tns:fromRole"/>
<xsd:element maxOccurs="1" minOccurs="0"
      ref="tns:GlobalDocumentFunctionCode"/>
<xsd:element ref="tns:PurchaseOrder"/>
<xsd:element

ref="tns:thisDocumentGenerationDateTime"/>
<xsd:element ref="tns:thisDocumentIdentifier"/>
<xsd:element ref="tns:toRole"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Figure 5: Part of a sample XML Schema for the 3A4 Purchase Order Request action message.

In addition to the action messages defined in the Web Service choreography, XML Schema definitions must also be generated for the standard SOAP header messages (Preamble, Delivery Header, and Service Header) as well as business signal messages (Receipt Acknowledgment and Exception) defined in RNIF.

4.5. WSDL - Content Designation

The current section provides the needed WSDL for the above-presented Web Service. The Business Message package will typically contain one WSDL document for each role defined in the interaction. Each WSDL document only includes the abstract section of a WSDL definition shown in Figure 6, namely the messages, operations, and portTypes. The portType will contain the operations exposed by each party. A message exchange for a typical standard message would therefore correspond to the invocation of a Web Service operation from such a portType.

For example, the RosettaNet PIP 3A4 Request Purchase Order incorporates a buyer role (retailer super market) and a seller role (manufacturing food industry) within its envisioned process. In order for such a buyer

```
<message name="PurchaseOrderRequestMessage">
<part name="Preamble"
type="preamble:Preamble"></part>
<part name="DeliveryHeader"
type="delivery:DeliveryHeader"></part>
<part name="ServiceHeader"
type="service:ServiceHeader"></part>
<part name="Pip3A4PurchaseOrderRequest"
type="pip3a4req:Pip3A4PurchaseOrderRequest"></par
t>
</message>
<portType name="PIP3A4ReqServicePort">
<operation name="PurchaseOrderRequest">
<input
message="tns:PurchaseOrderRequestMessage"></inpu
t>
</operation>
<operation name="Exception">
<input message="tns:ExceptionSignal"></input>
</operation>
<operation name="ReceiptAcknowledgment">
<input message="tns:ReceiptAckSignal"></input>
</operation>
</portType>
```

Figure 6: Part of a sample WSDL definition for the 3A4 Purchase Order Request action message.

to send a purchase order to the seller, the seller must expose a receivePurchaseOrder Web Service on its portType that consumes a WSDL message containing the PurchaseOrder.

The binding and port information, which specify the actual Web Service implementation (the "how" and "where"), will not be added until the WSDL definitions are deployed by the given party.

A SOAP representation of the Business Message would be constructed using the WSDL message definitions. The SOAP body will contain all elements of the standard Business Message. Keeping all components of the Business Message together allows easier conversion to / from the RNIF 2.0 format by a trading partner or trusted hub and may provide advantages for message processing and archiving.

5. Conclusions

This paper addresses an architecture for the implementation of collaborative processes in a flexible and efficient way. This architecture builds upon the RosettaNet Implementation Framework, which is used for the resolution of the business process layer requirements. Furthermore, ontologies are used for the definition of common terminologies and the provision of a unifying approach regarding the needed content. The overall implementation is based on the utilization of Web Services. As a case study example the Continuous Replenishment process between two collaborating enterprises has been deployed.

The work presented in this paper has been partially supported by the General Secretariat for Research and Technology (GSRT) of the Hellenic Ministry of Development under the framework "Competitiveness" through the project No26, Action 4.5.1 [24].

References

[1] *A. Kalogeras et al. "*Vertical Integration of Enterprise Industrial Systems Utilizing Web Services", WFCS2004, pp.187-192.
[2] S. Shim et al., "Business-to-Business E-Commerce Frameworks",IEEE Computer, pp.40-47,October 2000.
[3] X12, EDI (Electronic Data Interchange) ANSI X12, http//www.x12.org.
[4] IETF, Electronic Data Interchange – Internet Integration (EDIINT), http//www.ietf.org.
[5] ebXML, http://www.ebxml.org.
[6] BizTalk , http/:/www.BizTalk.org.
[7] RosettaNet, http://www.rosettanet.org
[8] IOTP, http:// www.iotp.org.
[9] ICE, http://www.icestandard.org.
[10] OBI, http://www.openbuy.org.
[11] P.Timmers, "Business Models for Electronic Markets", Journal on Electronic Markets, 8 (2): 3-8. 1998.
[12] R. Vergin, K. Barr, "Building Competitiveness in Grocery Supply Through CRP" , Industrial Marketing Management 28, pp. 145-153, 1999.
[13] S. Lee et al., "Business Value of B2B Electronic Commerce", Electronic Commerce Research and Applications 2, pp. 350-361, 2003.
[14] Simple Object Access Protocol, http:// www. W3.org/TR/soap.
[15] B. Medjahed et al., "Business-to-Business Interactions: issues and enabling technologies", The VLDB Journal 12, pp. 59-85, 2003.
[16] Web Services Activity, http://www.w3.org/2002/ws/.
[17] The OMG CORBA Website, http://www.corba.org/.
[18] Enterprise JavaBeans Technology (EJB), java.sun.com/products/ejb/
[19] [The Distributed Component Object Model (DCOM), Microsoft COM Technologies, http://www.microsoft.com/com/tech/DCOM.asp.
[20] (WSDL)2.0. W3C, 2003, http://www.w3.org/tr/wsdl
[21] UDDI ver, 3.0, UDDI.org, 2002, http://uddi.org
[22] J. Vergara et al., "Applying the Web Ontology Language to Management Information Definitions" , IEEE Communications Magazine, pp. 68-74, July 2004.
[23] D. Fensel et al., "Product Data Integration in B2B E-Commerce" , IEEE Intelligent Systems, pp. 54-59, July/August 2001.
[24] http://www.gsrt.gr