The Maturity of Open Systems for B2B

EJUB KAJAN

University of Nis, Serbia

This paper deals with the B2B interoperability in the sense of open systems and available component and framework standards. Special attention is given to the common reference framework for B2B interactions at the middleware level. Some relevant frameworks at the content and business layers are also described. Finally, concluding remarks emphasize future directions in the area of B2B interoperability. Many of the highlighted issues may also be used in other e-applications and in heterogeneous distributed systems in general.

Categories and Subject Descriptors: K.4.4 [**Computers and Society**]: Electronic Commerce
General Terms: B2B, interoperability, standards, open systems, middleware

1. INTRODUCTION

The idea to build a distributed open network consisting of systems that may communicate with no boundaries was born in the middle of 1980s [21]. Despite OSI failure due to its robust specification and expensive implementation, OSI era made a significant movement from proprietary to open systems. Several factors influenced OSI failure, which may be summarized as bad timing, bad technology, bad implementations and bad politics [44]. The goal of this paper is to analyse the current state and future trends of open technologies that may be used to overcome current limitations in full B2B interoperability in a fair way for companies of any sizes and any kind of business.

Today we may define open systems as computer systems which satisfy portability, scalability and interoperability (PSI) attributes [23] and consist of components, either hardware or software, that meet widely accepted standards (de facto or de jure) and where every interested party may participate. *Portability* refers to source code portability to different hardware platforms. It allows application programmers to concentrate on the application problem instead of doing the same things over and over again. *Scalability* refers to application portability between different sizes of machines as soon as a machine resources satisfy application needs. *Interoperability* refers to communications across heterogeneous geographicaly distributed systems at the required level. The modern approach to open systems leads to so-called network centric computing where the whole network may be thought of as a big distributed multiprocessor machine.

Author address: JUNIS Department, University of Nis, Univerzitetski trg 2, 18000 Nis, Serbia
Email: ekajan@ni.ac.yu, ekajan@ieee.org

ACM SIGEcom Exchanges, Vol. 5, No. 2, November 2004, Pages 34-44.

Nowadays we can easily imagine or we should better just remember a marketing presentation of a software vendor that starts with an ACME [23] message like this: “Hi folks, we have a perfect software solution for your business, based on international standards, easy to implement, etc., etc…” In practice, things are different and customers are often confronted with that neither they have got what they wanted nor what their vendors claimed. In fact, both sides did not correctly understand the term “openness” at the given time. The question of how much openness has been really achieved may be judged by recent application requirements such as the emerging field of electronic commerce. In order to address the problem of B2B openness rather than to be comprehensive, and to save space as well, some relevant technologies and frameworks such as workflows, OTP, OFX, OAGIS, UBL, BPML, BPQL, S2ML, OTA, RosettaNet, etc and B2B integration systems on the market such as SunOne, Microsoft .NET, IBM WebSphere, Oracle9iAS, etc. are not covered. A complete overview of these can be found in [45].

2. ONLINE ECONOMY

The idea of exchanging business data between applications is nothing new and it has been implemented since 1970s. The framework used was Electronic Data Interchange (EDI) based on international standards X.12 and UN/EDIFACT both aimed to minimize the cost, effort, and time for processing paper-based business documents. Despite its standard-based specifications, EDI failed for the same reasons as OSI. Big companies (e.g. Fortune1000) invested a lot of money in EDI implementation without the ability to conduct their business electronically because their partners, mostly Small-to-Medium Enterprises (SMEs) stayed out.

The growth of the Internet and Web is revolutionizing the way of business interactions. Numerous organizations started to use the Web as a way of doing business. It was a way out from the bottleneck caused by EDI failure as well as a new business opportunity known today as electronic commerce [37], [41]. *Electronic commerce (EC)* can be thought of as an emerging concept that describes the process of buying, selling, or exchanging data, services and products over the Internet [24]. In fact, when we think about EC today we usually assume e-business.

EC affects almost every aspect of how business is conducted. Information gathering, shopping, trading, brokering, banking, accounting, negotiating, manufacturing, scheduling, marketing, supplying, servicing, retailing, tax authorities, etc., all experienced the benefits from this emerging field [41]. EC may happen at any place at any time in any way, which makes the big difference to any other traditional way of doing business. It may be classified by the nature of transactions or by the business models [46]. By nature of transactions, we may distinguish B2B (business-to-business), B2C, B2E, C2B, C2C, etc., where C and E stands for customers and employees respectively. Today, most of EC is usually B2B. Various new business models are also established thanks to the EC concept. Some well-known are "find the best price", "name your price", group purchasing, supply chain management, on-line tendering and auctions, etc.

Let us define B2B as electronic commerce between two or more business partners via the third wave of EC [20], [24]. The first wave of EC consists of a company website that offers a catalogue of its products and services. Web farming is another familiar way for discovering the websites of interest and then systematically collecting relevant business

data in order to integrate them into the existing data warehouse. The second wave of EC carries the customers who have already established some kind of on-line shopping with a limited number of their suppliers (like classical EDI) using their websites and central database support. A surfer on the third wave of EC (B2B) requires the interactions among a vast variety of organizations to be handled seamlessly and dynamically with no problems in ad hoc integrations.

3. THE REQUIREMENTS FOR B2B INTEROPERABILITY

In order to have efficient EC, many heterogeneous applications must be integrated both *within and outside* an enterprise into a single coherent environment, as shown in Fig. 1, where "*be*" stands for a business entity (an enterprise).

The integration *within* an enterprise requires all exisiting applications, for instance Enterprise Resource Planing (ERP), Business Information System (BIS), Web portal, etc., to comunicate with no data latency. This approach is known as ZLE (Zero Latency Enterprise), the term first coined by Compaq and Gartner, and defines a business-technical strategy applied to data exchange across technical and organizational boundaries within an enterprise in order to make an almost perfect match between strategic goals and information available from technical resources (data bases, data warehouses, etc.). It is being done over proprietary value added networks (VANs) or over the Internet.

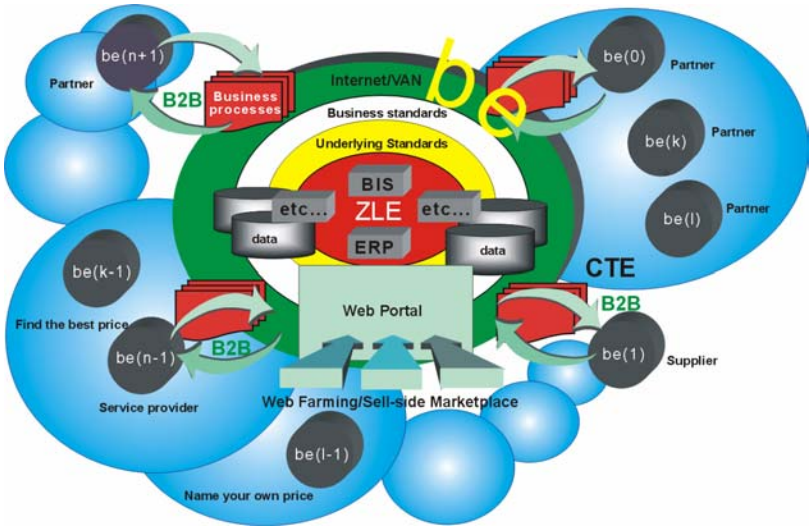

Fig. 1. A vision of open B2B

The *outside* integration is represented by application-to application (A2A) integration across enterprise boundaries (e.g. firewalls) over the Internet rather than over VANs. In the simplest way it allows automated exchange of information between business entities at the business process level, instead at the communication level only. It may include product catalogs, common supply chain management, process coordination, backend integration and many other topics. Various business models should also be supported. For instance, an enterprise may decide to join a CTE (customer trading exchange) to improve supply chain management, as shown in Fig. 1, with be(0), be(k) and be(l), or to establish several different business models for own sales.

Many lessons have been learned [9], [22], [24] after OSI, EDI, dot-com and similar failures. A heavyweight globally unique super-standard for B2B exchange is not a magic solution and it is not solution at all. Instead, the solution for EC must be fair for all and it must be open and scalable so that any size of enterprise with any kind of business may participate. It should be based on the lightweight, easy-to-implement, reusable open components. It requires the common framework to be defined for both underlying and business standards at all layers and at all dimensions.

4. THE COMMON FRAMEWORK AND UNDERLYING STANDARDS

By default, B2B systems are both heterogeneous and distributed, which cause their strengths and weaknesses. Building and programming such applications is difficult due to the lack of a common underlying framework. A framework is a generic template that provides the desired functionality. In the case of B2B interoperability, the framework must provide basic functionality, such as standard data format, security and content management [42] Fig. 2 shows a common framework suitable for building an appropriate B2B architecture. The platform itself is multi-layered and multi-dimensional. Behind the framework is a three-tier Web application model consisting of a thin client, a middle-tier (usually a Web server and/or Application Server AS), and a database server, DB. This should be recognized as the widely accepted architectural background for other dimensions of the framework.

The framework follows component-based EC [6], [16], [43], the main rules and achievements of open systems [24] and B2B integration frameworks [11]. The components are software modules that can be developed and delivered independently and that can be combined to form larger systems. Instead of delivering a system as a prepackaged monolith system, this platform should work as the Rubik cube offering a lightweight kernel and many other features to choose. Anyone, designers, development engineers, end-users, IT managers, etc., should be able to rotate a cube and tailor a platform of their own, select the appropriate B2B architecture and finally get what they want.

The standards arena offers plenty of programming tools for developing portable code. Examples are C, C++, C#, Java, HTML, Perl, Visual Basic, PHP, some variants of COBOL, etc. There are also integrated platforms that allow easy coding such as IBM VisualAge, Microsoft .NET, Sun J2EE, etc. XML is specially designed for Web applications as a technology that allows the creation of an unlimited number of different markup languages for different purposes. UN/CEFACT and OASIS defined ebXML (e-business XML) to provide an XML-based open technical framework that allows XML for the exchange of business data in EC environments [18].

In the field of security, various standards are proposed. These include secure E-mail messaging using PGP (Pretty Good Privacy) [52] or S/MIME [34] over SSL (Secure Socket Layer) [32] and PKI (Public Key Infrastructure) with underlying crypto algorithms such as RSA [36], MD5 (Message Digest) [31], etc. In addition, good standard guidelines for trusted computer base is given in the Orange Book [29] that contains the evaluation criteria for accessing the degree of assurance in the security features of computer systems divided these into classes ranging from D (minimum security) to A1 (maximum security). Although standards are important, dealing with security in EC is not just about standards. Many other issues should also be considered carefully such as fighting against malicious code, legal and social aspects, infrastructure development, digital rights management intra- and extra-organizational weaknesses of the

partners involved (e.g. insider abuse), discrepancies between countries in the global economy, etc.

The run-time environment shown in Fig. 2 consists of three groups of layers, bottom, middle and top. The lowest part of the bottom group is the hardware layer based on different kind of microprocessors (CISC or RISC) made by many vendors (Intel, IBM, TI, SGI, Digital, AMD, etc.). The layered architecture hides differences between these, so there are no influences for platform openness. In the operating systems arena, three platforms are established as dominant and almost unique: UNIX, Linux and Windows. At the top of the bottom group is the communication protocol layer based on the TCP/IP protocol suite and many communication services based on it. Demands for additional services, communication overhead between end systems, overloaded address space, wasting time during protocol encapsulation at every intermediate node, etc. influenced the development of IPv6 and many other protocols listed in [36]. At this layer of open B2B, the recent protocols are SOAP [49] and BEEP [35].

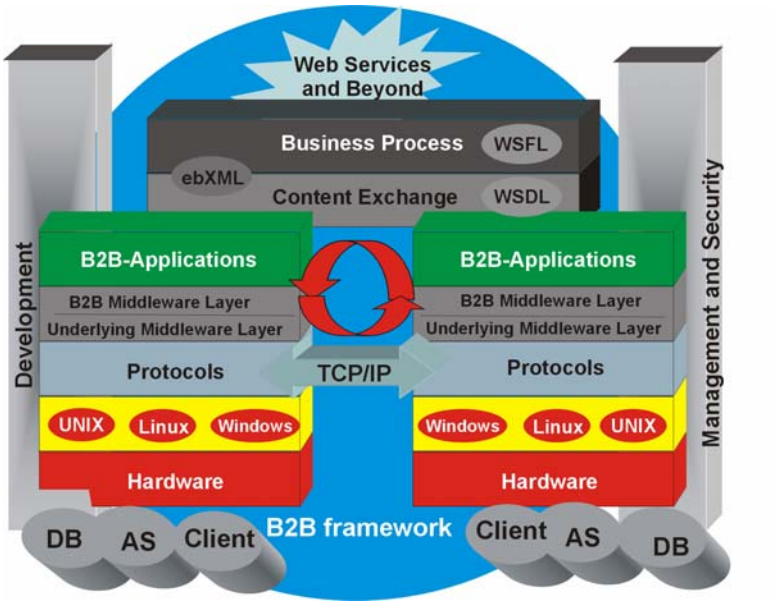

Fig. 2. The B2B reference framework

SOAP (Simple Object Access Protocol) is a lightweight protocol intended for exchanging structured XML data in a distributed environment. It defines message formats and describes a variety of message patterns, including, but not limited to, RPC, asynchronous event notification and forwarding via SOAP nodes. SOAP can be used with a variety of protocols such as HTTP, SMTP and FTP, either using BEEP or not.

BEEP (Blocks Extensible Exchange Protocol) is a connection-oriented, message-oriented asynchronous protocol that supports either peer-to-peer or client/server interactions. Messages are arbitrary MIME content but are usually textual using XML, thus BEEP does for Internet protocols what XML has done for document and data. It takes care of the connections, the authentication and the packing up at the TCP/IP level of the messages. The protocol is very simple. Five types of messages are supported that are exchanged in the context of the channels (up to 257), the bindings to the well-defined

aspects of the applications such as transport security, user authentication or data exchange. In fact, the BEEP is not just another Internet protocol. It is also a framework that allows development of application protocols in a manner that programmers do not need to waste their time thinking about low-level protocol details, such as bindings, etc.

In the middle of the execution environment is so-called *middleware*. The role of middleware is to make uniformity between bottom layers, which are different by default, and various applications that are different by nature (e.g. by B2B business models). Various middleware platforms, which may also have their own protocol stack, are proposed and are in use. These provide distributed file system services, naming, messaging, resource sharing, etc. The idea of Web is an example of a document-based middleware, where the simple combinations of websites and hyperlinks hide the differences between servers where the websites reside making them readable by any browser. Another group of middleware technologies are directory services such as X.500 Directory services and LDAP [33]. A variety of MOM (Message-Oriented Middleware) technologies are also in use [25]. Component middleware are mostly represented by CORBA [48], DCOM [12], EJB [40] and various RPCs [7], etc. All of these may be viewed as underlying middleware components because these may used in B2B, but not specifically designed for B2B.

5. B2B SPECIFIC STANDARDS

Generaly, a B2B protocol standard is the description of the message format exchange, binding to transport protocols, sequencing, process, security to be provided, and many more properties [9]. At this layer of middleware, two sublayers should be carefully considered depending on the subject of integration: the layer of the content exchange and the layer where the business processes cooperate as shown in Fig. 2. At the *content layer* semantics and structural heterogenity issues are resolved as well as the transport binding to specific transport protocols either statically at development time or dynamically at runtime. A business process is defined as a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships [11]. Unlike content layer where the business entities should agree on data formats, data models and languages, at a *business process layer* partners are concerned with the conversional interactions. Among many existing frameworks at this layer [9], [11], [13], two of these OBI [30] and Web Services [1], are described.

An open flexible framework for secure and interoperable B2B at the content level is *OBI (Open Buying on the Internet)* standard. The OBI architecture recognizes four entities (Requester, Buying Organization, Selling Organization and Payment Authority) in a trading process (see Fig. 3). The OBI trading chain starts when a requester at a buying organization searches an on-line catalogue, using a standard Web browser, at a selling organization and places an order into the shopping basket (step 1). Such an order is processed by the selling organization resulting in an order request (step 2), transmitted to the buying organization for approval (step 3). The underlying secure communications are based on the SSL, while the OBI transactions are based on the standard X.12 EDI data formats. After receiving the order approval, the selling organization issues the invoice and transmits it to the payment authority (step 4) using Public Key Cryptography Standards (PKCS) [39] over SSL. For authorizing itself, the payment authority uses the

X.509 certificate, passes the crypted invoice to the buying organization, which pays the bill (step 5) using PKCS over SSL for reverse transaction.

The key idea behind *Web services* is to allow a program to be a reqeuster which can discover Web services and invoke them fully automated. In terms of B2B interactions that means that an interaction between business partners may be established without human influence. The concept is based on the loosely coupled reusable software components which can be brought togheter at the time of service invocation. Three major standardization initiatives are submitted to the W3 consortium: SOAP, WSDL and UDDI. *UDDI (Universal Description, Discovery and Integration)* [47] provides a mechanism for clients to find web services either to publish a service description or to obtain a needed service. *WSDL (Web Services Description Language)* [50] is an XML-based language used for describing Web services independently of concrete network deployment and data formats.

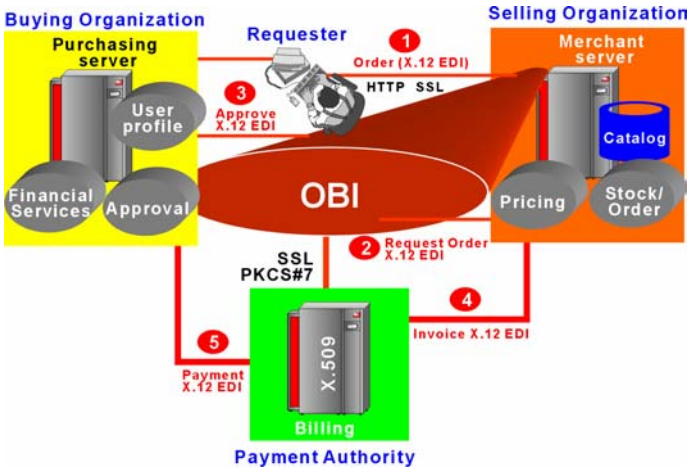

Fig. 3. OBI in action

Web services can be compared with the classical static underlying technologies of the Web. Static Web is organized around three key technologies: HTML is used to describe the content of the Web documents, HTTP to publish or to obtain these documents and the URI to locate documents. In case of Web services deployed, we may think about dynamic Web [10], [15]. A service provider defines a service offered using WSDL and publish such a service by UDDI. A service customer (program looking for a service) uses UDDI to consult the UDDI registry and to request the needed service. If such a service exists, SOAP is used for message exchange.

At the current state of Web services definitions, there is no support for interactions at the business process layer, so the full maturity of their openness is not reached yet. However, some efforts to provide standard-based semantic interoperability between business processes using content markup languages are on the way [27]. There are also attempts for providing *composite* Web services [2]. These include *WSFL (Web Services Flow Language)* [26], XLANG [51] and *BPEL4WS (Business Process Execution Language for Web Services)* [8]. The key idea behind composite Web services is to combine partial basic services and form new services that have something in common.

Much wider approach for fully enabled B2B based on Web services is represented by *WSMF (Web Service Modeling Framework)* [10], [15]. WSMF is centered around

complementary principles: strong de-coupling of the various components that realize an B2B application and strong mediation service that should allow fully P2P (peer-to-peer) communications between business entities as show in Fig. 1. Without strong mediation by the third party, every P2P connection requires all the transformations to be handled by both peers. With the current heterogeneity of EC environment that means that every *be* having *n-1* partners must implement *n(n-1)* translators for every incompatible component involved in business transactions. If we are dealing with few *bes* (small n), a small number of incompatible components the P2P is probably costly but still posible, but this is a step back, far from open B2B and from reality. In reality the hundreds of transalators are required. There are few initiatives which are trying to avoid the limitations above. The examples are WSMF and OGSA, both based on Web services, but most important is the Semantic Web, as follows.

To solve the above issues, WSMF defines four different main elements: ontologies, goal repositories, web services descriptions and mediators. *Ontologies* provide a shared and common understanding of a domain and they are used here to define the terminology that is used by other elements of WSMF. *Goal repositories* describe the objectives that a client may have asking for a Web service. *Mediators* should solve various P2P incompatibilities such as data formats, business logics, service invocation, message exchange etc. Web services descriptions should allow the distinctions between internal an external processes and the external complexity of a Web service as well.

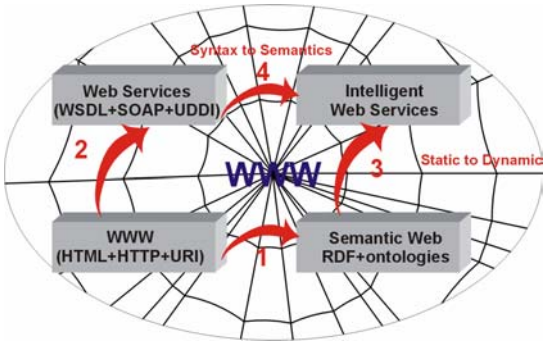

Fig. 4. The evolution of the Web in order to reach full potential

WSMF is based on another idea of the future Web, known as *Semantic Web* [5], new Web based on artificial intelligence and knowledgeable, machine-readable data that is much easier to find, access, present and maintain. The stack of knowledge technologies that build the semantic Web starts with minimum knowledge of data (XML that may describe the structure of the data but there is no support to tell something about data), followed by intermediate knowledge (*RDF, Resource Description Framework*, that allows creating metadata, the data about data [56]), and ends with maximum knowledge about domains, processess, etc. RDF and RDF schema alone are not able to bring all the knowledge required for full open B2B interoperability because they do not support the logical semantics of the resources. The higher degree of knowledge may be reached by ontologies, a backbone of the semantic and intelligent Web as shown in Fig. 4.

An ontology is a formal explicit specification of a shared conceptualization [19]. Three keywords are dominant here: shared conceptualization, explicitness and formality. Shared conceptualization refers to an abstract model of something (e.g., a service, a

product) that describes the relevant concepts of that thing and that could be shared among diffrent users (people, other services, etc.). Such concepts must be explicitly defined, including the constrains of their use whilst refers to machine-readable form of the ontology.

Early efforts to provide openness of ontologies are represented by Ontolingua [14], [19] and KIF [4]. Ontolingua is a very high power tool for describing ontologies, syntax and semantics of which are described in KIF, a language for knowledge interchange among disparate systems. Recent efforts are represented by OIL [57], DAML-S [3] and most recent by OWL-S [28], the successor of DAML-S. *OWL-S (Ontology Web Language for Web Services)* goes beyond all previously described technologies in terms of expressing meaning and semantics on the Web, especially Web services. Three interrelated OWL subontologies, the profile, process model and grounding form OWL-S, are used to express what a service does, how it works and to map process model onto detailed specifications of message formats, protocols and so on.

Another general approach for creating and composing sophisticated distributed systems capable of integrating services across distributed heterogeneous virtual environments is represented with the emerging *grid computing* and proposed open architecture. The name of the game is *OGSA (Open Grid Service Architecture)* [17] that combines concepts and technologies from the Grid and Web services comunities. Grid computing has originally been intended for high performance resource sharing such are, for instance, *European Grid* [53] or *SETI (Search for Extraterestrial Intelligence)* project [54]. A general overview of Grid initiatives can be found at [55]. OGSA tends to align Grid technologies and Web services to obtain standards-based distributed service systems that support the creation of the sophisticated distributed services required for open B2B. OGSA requires only minor extension to existing technologies and represents a natural evolution of Web services.

6. CONCLUDING REMARKS

Many B2B standards are proposed by different groups, some of them even overlapping or competing with each other. It seems that we currently have just a bunch of semi-isolated platforms instead of isolated data islands which existed in the proprietary environments. However, the current state of technologies promise at least a fairer solution for any kind of enterprise. Despite many implementations of Web services, the full openness of B2B is not yet reached. Most recent research show the posible way out: development and deployment of the specialized ontologies that may understand both sides of the business and help them to establish automatic business transactions, no matter how much they are disparate.

REFERENCES

[1] AALST, van der W. et al. 2003. Web Services: Been There, Done That? *IEEE Intelligent Systems, January/February 2003*, pp. 72-85.

[2] AISSI, S., PALLAVI, M. and SRINIVASAN, K. 2002. E-Business Process Modeling: The Next Big Step. *IEEE Computer, Vol. 35, No. 5*, pp. 55-62.

[3] ANKOLEKAR, A. et al. 2002. DAML-S: Web Service Description for the Semantic Web. *In Proceedings of the 1st ISWC*, Sardinia, Italy.

[4] ANSI draft NCITS T2/98-004

[5] BERNERS-LEE, T., HENDLER, J. and LASSILA, O. 2001. The Semantic Web. *Scientific American, Vol. 284, No. 5*, pp.34-43.

[6] BICHLER, M. SEGEV, A. and ZHAO, L. J. 1998. Component-based E-Commerce: Assessment of Current Practices and Future Directions. *ACM SIGMOD Record, Vol. 27, No. 4*, pp. 7-14.
[7] BIRREL, A.D. and NELSON, B.J. 1984. Implementing Remote Procedure Calls. *ACM Transactions on Computer Systems, Vol. 2, No. 1,* pp. 39-59.
[8] BPEL. http://www-106.ibm.com/developerworks/Webservices/library/ws-bpel
[9] BUSSLER, C. 2001. B2B Protocol Standards and their Role in Semantic B2B Integration Engines. *IEEE Data Engineering Bulletin, Vol. 24, No. 1*, pp. 3-11
[10] BUSSLER, C. FENSEL, D. and MAEDCHE, A. 2002. A Conceptual Architecture for Semantic Web Enabled Web Services. *ACM SIGMOD Record, Vol. 31. No. 4*, pp. 24-29.
[11] DABOUS, F.T., RABHI, F.A. and RAY, P.K. 2003. Middleware Technologies for B2B Integration. *IEC Technical Review*, IEC Press, pp. 1-13.
[12] DCOM/1.0. http://www.microsoft.com/Com/Recources/comdoc.asp
[13] DOGAC, A. and CINGIL, I. 2001. A Survey and Comparison of Business-to-Business E-Commerce Frameworks, *ACM SIGecom Exchanges, Vol. 2, No.2*, pp. 16-27.
[14] FARQUHAR, A. et al. 1997. Tools for Assembling Modular Ontologies in Ontolingua. *In Proceedings of AAAI97*, pp. 436-441.
[15] FENSEL, D. and BUSSLER, C. The WSMF, http://www.swsi.org/resources/wsmf-paper.pdf
[16] FINGAR, P. 2000. Component-Based Frameworks for E-Commerce. *CACM, Vol. 43, No. 10*, pp. 61-66.
[17] FOSTER I. et al. The Physiology of the Grid-An Open Grid Services Architecture for Distributed Systems Integration. http://www.globus.org/research/papers/ogsa.pdf
[18] GIBB, B. and DAMODARAN, S. 2003. *ebXML: Concepts and Applications*. Wiley, 2003.
[19] GRUBER, T.R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition, Vol. 5*, pp. 199-220.
[20] GUTIEREZ, D.B. and MARTINEZ, L.T. 2000. A Focus on a B2B E-Commerce Architecture, *Simposio Español de Informatica Distribuida (SEID 2000)*, Orense, Spain, pp. 305-312.
[21] ISO/IEC7948-1: Open Systems Interconnections: Basic Reference Model. 1984
[22] JANTAVONGSO, S. and LI, R.K-Y. 2003. A New Age e-business Model for SME. *In Proceedings of AUSWEB 2003, Gold Coast*, Australia, 5-9 July 2003.
[23] KAJAN, E. 2002. *Information Technology Encyclopedia and Acronyms*. Springer-Verlag, Germany
[24] KAJAN, E. 2003. The B2B Standard IT Framework. *In Proceedings of IADIS Conference e-society 2003*, Lisbon, 3-6 June 2003, pp. 597-602.
[25] KORHONEN, M. 1997. MOM. http://www.tml.hut.fi/Opinnot/Tik-110.551/1997/mqs.htm
[26] LEYMANN, F. WSFL 1.0, http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
[27] McILRAITH, S., SON, T.C. and ZENG H. 2000. Semantic Web Services. *IEEE Intelligent Systems, March/April 2001*, pp. 46-53.
[28] MARTIN D. et al. 2004, Bringing Semantics to Web Services: The-OWL-S Approach. *In Proceedings of the 1st SWSWPC*, July 2004, San Diego, CA.
[29] NCSC: Trusted Computer System Evaluation Criteria. 5200.28-STD, 1985.
[30] OBI, http://www.openbuy.org
[31] RFC1321: The MD5 Message Digest Algorithm, 1992.
[32] RFC2246: Transport Layer Security. 1999.
[33] RFC2251: Lightweight Directory Access Protocol v.3. 1997
[34] RFC2633: S/MIME Version 3 Message Syntax Specification. 1999.
[35] RFC3080: The Blocks Extensible Exchange Protocol Core. 2001.
[36] RFC3600: Internet Official Protocol Standards. November 2003.
[37] RIGGINS, F.J. and RHEE, H.S. 1998. Toward a Unified View of Electronic Commerce, *CACM, Vol. 41, No. 10*, pp. 88-94.
[38] RIVEST, R.L., SHAMIR, A. and ADLEMAN, L. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *CACM, Vol. 21, No. 2*, pp. 120-126.
[39] http://www.rsa.com/rsalabs/pkcs/index.html
[40] Sun: J2EE Enterprise Java Beans Technologies. http://java.sun.com/products/ejb
[41] SHAW, M.J. 1999. Electronic Commerce: Review of Critical Research Issues. *Information Systems Frontiers, Vol. 1, No. 1*, pp. 99-106
[42] SHIM, S.S.Y et al. 2000. Business-to-Business E-Commerce Frameworks, *IEEE Computer, Vol. 33, No. 10*, pp. 40-47.
[43] SPROTT, D. 2000. Componentizing the Enterprise Application Packages, *CACM, Vol. 43, No. 4*, pp. 63-69.
[44] TANENBAUM, A.S. 1996. Computer Networks. Prentice-Hall, Upper Saddle River, New Jersey.
[45] TELLMANN, R. and MAEDCHE, A. 2003. Analysis of B2B Standards and Systems, SWWS Consortium, http://swws.semanticweb.org/public_doc/D11.1.pdf

[46] TURBAN, E. et al. 2002. *Electronic Commerce 2002- A Managerial Perspective*. Prentice-Hall, Upper Saddle River, New Jersey
[47] UDDI, http://www.uddi.org
[48] VINOSKI, S. 1997. Integrated Diverse Applications within Distributed Heterogeneous Environments. *IEEE Communications Magazine, Vol. 35, No. 2*, pp. 46-55.
[49] W3C: Simple Object Access Protocol (SOAP). http://www.w3.org/TR/soap
[50] WSDL, http://www.w3.org/TR/wsdl
[51] http://xml.coverpages.org/xlang.html
[52] ZIMMERMAN, P. 1995. *PGP Source Code and Internals*. MIT Press.
[53] http://eu-datagrid.web.cern.ch/eu-datagrid
[54] http://www.set@home.ssl.berkeley.edu
[55] http:/www.gridcomputingplanet.com/resources
[56] http://www.w3.org/TR/rdf-concepts
[57] http://www.ontoknowledge.org/oil