Evaluation of Technologies for Business Process Automation

Maja Pušnik, Matjaž B. Jurič and Ivan Rozman
University of Maribor, Faculty of Electrical Engineering, Computer and Information Science,
Institute of Informatics,
Smetanova 17, 2000 Maribor
E-mail: maja.pusnik@uni-mb.si

Keywords: business process automation technologies, decision models, ebXML, XLANG, RosettaNet

Received: June 2, 2002

The importance of process automation for B2B (business to business) collaboration is rising. The efforts are directed towards automating business processes and forming a global electronic market. In this paper we present and evaluate the three most important technologies for business process automation: ebXML (Electronic Business XML- eXtensible Markup Language), RosettaNet and XLANG. They differ in terms of features, quality and serviceability. We analyze, compare and evaluate those technologies from the perspective of SME (small and medium enterprises). Based on the comparison we define a multi-criteria decision model with twenty parameters and the corresponding weights, we evaluate the alternatives and define a utility function, which helps us to select the most suitable technology. The contributions of this paper are the in-depth evaluation of technologies and the definition of a multi-criteria decision model.

1 Introduction

The well-known fact is that business must be altered to survive the upcoming changes and progress. To make the idea of a global marketplace and B2B work, proper technologies, which will assure safety and efficiency, must be created. They have to be appropriate for all kinds of enterprises, small and large, for those with great financial recourses and responsibilities and for those with limited budgets. Only with such universal technologies, a global market and complete serviceability will be realized.

In the paper we will review and compare the three most important technologies for business process automation: ebXML, XLANG and RosettaNet. We will define criteria for their evaluation and build a decision model with twenty criteria. We will evaluate the results and choose the most suitable technology from the perspective of a SME. All three technologies are based on XML and build on the functionality of web services, where they reuse existing web service technologies, such as SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration) and WSDL (Web Service Definition Language). We will see that they differ in some features while in others they are complementary. Because they are based on open standards, they are reachable in aspects of price and complexity, not only to large enterprises, but also to small and medium enterprises.

The review of related research has shown that there are not many similar analyses. The comparison made in [6] only compares ebXML and RosettaNet in an informal way and does not define a decision model. The author in [20] compares B2B standards, which include RosettaNet, ebXML, OAGIS (Open Applications Group Integration Specification) and Simple Web Services. The same author explains in a different article [21] how RosettaNet, ebXML, OAGIS and EDI (Electronic Data Interchange) fit together. However the author does not define a formal decision model. In [22] the author again compares RosettaNet, ebXML, OAGIS, Web Services, xCBL (UBL) – XML Common Business Library (Universal Business Language) and cXML (commerce XML) and creates a comparison framework.

Our paper is organized in the following order: the needs of the market are evaluated in the second chapter. Third chapter makes a comparison of the ebXML, RosettaNet and XLANG. Fourth chapter defines a multi-criteria decision model and evaluates them. The last, fifth chapter, gives a conclusion of the results.

2 Needs of the Market

The way enterprises work, understand their existence and survive must be retained. But the way they do business and communicate with each other must be improved. So business processes must still work on and through the net, just as they have manually. Technologies must describe business processes in a consistent and safe way and more. They must enable changes, upgrades and adaptations, since business is a living process and therefore must be flexible and manageable.

But this is only the first step. There is still the question of automation. A business process consists of many steps and includes many people, some of them completely unnecessary, which only enlarges the possibility of making a mistake. One of the goals in creating a global electronic market is to automate everything that can be automated, including routine work or explicitly defined processes with long-term rules and foreseen conditions.

Some solutions have already been created in the past, more or less successfully, but by far not sufficient enough for goals and ambitions of the millennium. The web services have only created an initiation of what is yet to come. They enabled process describing, but not automation. The ultimate goal of those technologies is making business as safe and as accessible as possible for all businesses all over the world [19].

3 Comparison of Technologies

There are several technologies for coordination and automation of business processes. Some of them have been present on the market for quite a long time, for example EDI. But the problem with vintage ones is inaccessibility for smaller enterprises and an obvious inflexibility, since most of them require a large initial investment and expensive support. The up to date technologies build upon legacy technologies, which have used older proprietary standards.

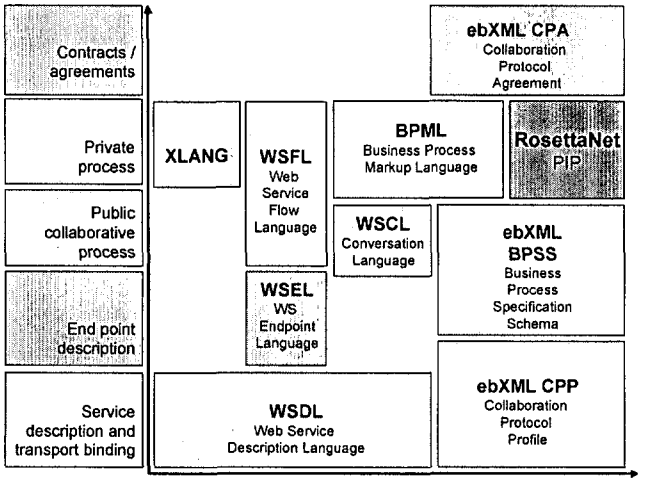

Figure 1: Process Coordination Framework [1]

The need for different kind of technologies has increased. Modern technologies are mutually connected and complemented. Figure 1 presents their relationships, horizontally divided by the level of provided services and vertically by the initiative organization, by which they were sponsored and created [1]. All of the technologies are an upgrade of web services and they are all based on the XML language. Their design priorities and fields of concentration however differ.

Service description and transport binding was assured in WSDL and in ebXML CPP (Collaboration Protocol Profile) as the EDI follower. With time, new technologies emerged from them in different directions. Their relations are seen from Figure 1: WSEL (Web Service Endpoint Language), ebXML BPSS (Business Process Specification Schema), WSCL (Web Service Conversation Language), WSFL (Web Service Flow Language), XLANG, BMPL (Business Management Markup Language), RosettaNet PIPs (Partner Interface Process) and ebXML CPA (Collaboration Protocol Agreement) [1].

WSDL is meant for describing network services as a set of endpoints, operating on messages. WSEL is meant for non-operational features of web services like security. WSCL allows that we define abstract interfaces for web services for business process conversation. WSFL allows the description of business processes or interaction patterns, based on the web services operations.

ebXML provides a set of technologies for describing various stages of business collaboration. CPPs enable companies to specify their profiles in which they define the terms for collaboration. CPAs are the computer equivalents of trading partner agreements. They can be defined manually or automatically generated from two or more CPPs. The actual flow of a business process is specified using BPSS. Shared public and private business processes for collaboration between two or more partners are specified using BPML. The focus of XLANG and RosettaNet PIPs is similar to BPML and will be further discussed later in this article.

XML, concentrated on the contents, enables remote systems to interchange and interpret the documents without the human intervention. XML document is basically an ordinary text file with markup [1]. The combination of structure, flexibility and verification makes XML useful not only for electronic publishing, but also for designing business messages, exchanged between enterprises [1]. While building larger processes, all business partners must agree upon the vocabulary, interfaces and the type of method invocation, before they send individual messages.

XML vocabularies can define all kinds of business documents or even whole frameworks, which provides interoperability and functionality.

3.1 ebXML

ebXML is a family of specifications that enable companies of all sizes to collaborate with each other, independently of the location [2], through the exchange of XML-based messages [8]. Development of the ebXML specifications is an on-going effort sponsored by OASIS (Organization for the Advancement of Structured Information) and UN/CEFACT (United Nations Center For Trade Facilitation & Electronic Business) [8].

The need for ebXML lies in the experience from the past. EDI, the anterior technology for data interchange among enterprises, was unreachable for most SMEs, since the

costs were too high and the implementation too complex. ebXML is based on XML, web services and open standards and is publicly available. It overcomes this barrier and enables the creation of software for building applications, based on mutual structure and syntax, which will lower the costs of business data interchange. ebXML mission is to provide an open XML-based infrastructure, enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties [8].

ebXML architecture was primarily designed for B2B interaction. UDDI and SOAP offer services with similar functionality on the low level. EbXML uses and builds upon these standards. It provides safe and reliable messaging and adds a set of higher level specifications for expressing the semantics of B2B collaborations. For these purposes it provides CPPs, CPAs, BPSS, core components, registry/repository and BPML [11].

ebXML provides an effective platform for long-term business transactions and enables us to express the following:

- quality of service,
- timeouts,
- conformations,
- multi-language support,
- authentication,
- authorization,
- privacy,
- integrity and
- non-repudiation.

Example of ebXML usage, shown in Figure 2:

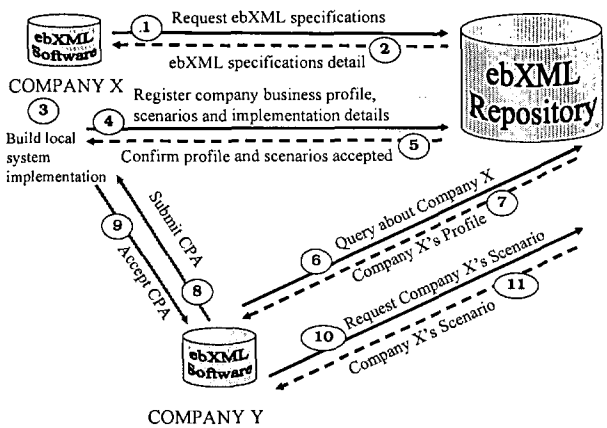

Figure 2: ebXML in practice

By using ebXML, companies have a standard method to exchange business messages, conduct trading relationships, communicate through data in common terms, define and register business processes [8]. It enables all parties to complement and extend current EC/EDI (electronic commerce/EDI) investment and it expands electronic business to new and existing partners. It also facilitates convergence of current and emerging XML efforts [8].

ebXML delivers the value by [8]:

- using the strengths of OASIS and UN/CEFACT to ensure a global, open process,
- developing technical specifications for the open ebXML infrastructure,
- creating the technical specifications with the world's best experts,
- collaborating with other initiatives and standard development organizations,
- building on the experience and strength of existing EDI knowledge,
- enlisting industry leaders to participate and adopt ebXML infrastructure and
- realizing the commitment by ebXML participants to implement the ebXML technical specifications.

3.2 XLANG

XLANG is a notation for the specification of message exchange behavior among participating web services, supporting especially the automation of business processes [9]. It is expected to serve as the basis for automated protocol engines that can track the state of process instances and help to enforce protocol correctness in message flows.

XLANG is based on XML and is used for describing business processes in the BizTalk initiative. It offers a model for orchestration of services and contract collaboration between partners [3]. XLANG is fully focused on public processes. It supports long-term operations and nesting. It enables:

- exception handling,
- restoring operations,
- behavior,
- actions,
- control flow,
- correlations,
- contents of transaction,
- service management,
- time-outs,
- custom correlation of messages,
- modular behavior description and
- contracts with multiple roles [3].

However, it does not define authentication or the quality of service nor the non-repudiation [4]. The goal of XLANG is to make it possible to formally specify business processes as state-full long-running interactions [9].

Main features of XLANG include [1]:

- *behavior;* container for the description of the service's behavioral aspects, including support for looping, concurrency and exception handling,
- *actions;* atoms of behavior, referencing WSDL operations on available ports,

- *control flow;* sequence in which the service performs actions,
- *correlations;* structure, the service uses to route messages to correct workflow instances,
- *context;* context for long-running transactions,
- *service management;* features of service instance management and
- *port mapping;* method for plugging in the service user and the service provider.

XLANG is an extension of WSDL and dynamics in processes are supported with different flows [3]:

1. *Message flow*, where actions are the basic constituents of an XLANG process definition that specifies the behavior of the service. The actions are request/response, solicit response, one way, notification, timeouts and exceptions.
2. *Data flow*, the base of XLANG is fed by the message flow and supports the control flow decisions.
3. *Control flow,* which provides support for looping, besides the regular elements. It also enables exception handling and transactional behavior.

XLANG also supports business process contracts, however they are merely mappings between two port types, which interact together. A contract can only map ports that are unidirectional [3].

The unit of action, offered by a service is an *operation.* An operation can be a single asynchronous message, or a request/response pair of messages with optional fault messages. The operation can be either incoming or outgoing. But WSDL does not say what is the operation semantics. There are three possibilities [17]:

1. In the first case the operation is a *stateless service that has no memory of previous operations,* such as a stock quote service.
2. The second possibility is an *operation on an object,* in the usual sense of object-oriented programming systems, in which case the object will have the ability to use its state variables to keep a record of the consequences of previous operations. In the latter case, we usually think of the object as being subservient to the caller, since the caller controls the entire life cycle of the object. The object itself has low influence regarding the order in which its operations are invoked and no independent behavior.
3. The third possibility is *autonomous agents with full state representation of the service.* In this case the service supports long-term interactions with full state, in which every interaction has a beginning, defined protocol for operation call and the ending. The supplier has to provide a service, which starts an interaction by receiving an order through the entering message, then returns the acknowledgement to the buyer, if the order can be accomplished.

Enterprise workflow systems today support the definition, execution and monitoring of long-running processes that coordinate the activities of multiple business applications. But they do not separate internal implementation from external protocol description [9].

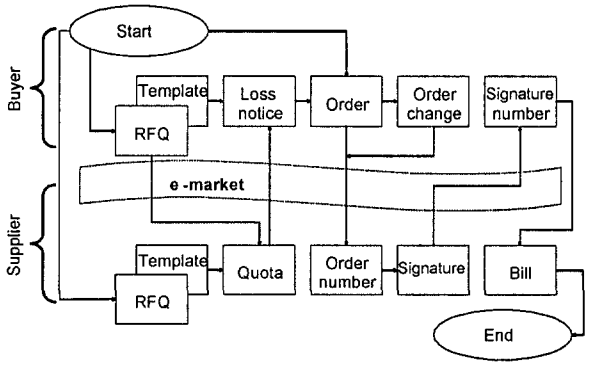

Figure 3: XLANG connecting two parties [9]

The Figure 3 represents the dynamics between two participants inside an electronic market, where XLANG is the translating key between a buyer and supplier that cooperate on the net using the advantages of the electronic market.

3.3 RosettaNet

RosettaNet is a non-profit consortium of more than 400 of the world's leading Information Technology, Electronic Components, Semiconductor Manufacturing and Solution Provider companies, working to create, implement and promote open electronic business process standards [7].

RosettaNet was created as a compromise between EDI and SOAP. Its main goals are reaching dynamic, flexible trading networks, operational efficiency and new business opportunities [10]. It enables:

- real time complex transitions,
- checking,
- confirmation,
- non-repudiation,
- multiple languages,
- additional standards in industry,
- SSL (Secure Socket Layer) authentication,
- digital signature and
- data encoding.

Its biggest advantage is the well defined although inflexible PIP [5]. The purpose of every PIP is to offer general business data models and documents, which enable interface implementation by system developers. Every interface includes [14]:

- XML document, which is based on the DTD (Document Type Definition) and specifies PIP services, transactions and management, which include dictionary properties,
- class and sequence diagrams in UML,
- validation tool and
- implementation guide.

PIP interface offers mechanism for sending messages and reporting failures. It demonstrates the integration of web services and its safety features, demanded at RosettaNet [19]:

- two – way SSL authentication,
- digital signature,
- data encryption and
- non-repudiation.

RosettaNet PIP defines an automated business process among trading partners for demanding and offering product prices and availability information [16]. Different business processes are covered with:

1. RosettaNet *executive plan*, which offers a general guidance, priorities of addresses and integration through tables.
2. Individual *plan of supply chain*, which address of the supply chain – specific theme, prioritization, sources, implementation and adaptation.
3. RosettaNet *partners*, which enable voting about standards, participants in workshops and implementation.

RosettaNet standards are managed on a global level. Locally they are focused on implementation and support. So partners can choose between global or local membership [13].

RosettaNet is very rich in its supporting tools: the RosettaNet implementation tool including the current PIP template, a Partner Agreement Wizard for quick importation, development and testing of customized PIP and more. It also contains RosettaNet dictionary and RosettaNet implementation framework. The template enables the development of new PIPs. The Partner Agreement Wizard enables importing of trading partners and a fast development of new processes. Embedded PIP enables implementation of only that certain PIP the partner needs. It includes support for all published RosettaNet PIPs as well as for CIDX (Chemical Industry Data Exchange) and PIDX (Petroleum Industry Data Interchange). PIP can also be tested before actually applied and used.

RosettaNet is also focused on the industry support; the adapter for industry development enables integration with new and existing applications and ways of business [15].

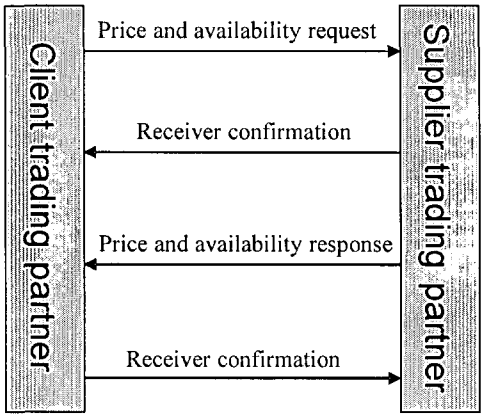

Figure 4: RosettaNet communication

RosettaNet plans to integrate support for the ebXML Messaging Services Specification in future releases of RosettaNet's Implementation Framework (RNIF). While RosettaNet remains committed to developing business process standards, required to support the complex needs of the high-technology industry, it also wants to ensure interoperability across all supply chains. Figure 4 represents the communication between two trading partners with help of RosettaNet PIP – which enables connection of business processes [12].

4 Evaluation Model

4.1 Criteria

To be able to evaluate the technologies for describing business processes for their suitability and quality, we have defined a multi-criteria decision model. We have identified the following criteria [18]:

Defining and describing processes: Evaluates the architectural support, syntax and semantics for describing all the features of the process and the support for the transition from classical to electronic business from aspects of flexibility, simplicity, user friendliness and compliance to standards.

Collaboration description: Evaluates the support for business interactions and defining relationships between partners, from aspects of flexibility, safety and complexity.

Role model: Evaluates the support with modeling tools for describing roles and collaboration between them.

Small/big/medium enterprises support: Evaluates the appropriateness and flexibility of the technology for different company sizes with different characteristics, needs and preferences.

Complexity and learning effort: Evaluates the amount of effort and change, needed to learn and understand the technology and all its features.

Efficiency: Evaluates how efficient is the technology at describing and specifying the business processes.

Maturity: Evaluates the maturity, based on the number of years the technology exists.

Tools support: Evaluates the support within tools and integrated development environments, which ease the development and assure quality.

Synchronous communication support: Evaluates support for synchronous, short-term transactions, which require immediate answer.

Asynchronous communication support: Evaluates support for asynchronous, long-term transactions.

Independency of communication protocols: Describes the relationship between communication protocol and the technology.

Quality of service: Evaluates the possibilities for specifying service quality of certain flows, which can be done either by raising the priority of a flow or limiting the priority of another flow.

Authentication: Evaluates the level of verification of the senders identity - whether the business message sender is or is not who he claims to be [6].

Authorization: Evaluates the level of verification, whether the sender of a message is permitted to send the subject message to the receiving partner [6].

Integrity: Evaluates, whether the messages remains unaltered during transportation [6].

Encryption: Evaluates the coding and the level of security of messages against unauthorized readers [6].

Non-repudiation: Evaluates the mechanism for verifying whether an originating trading partner can or cannot deny having originated and sent a message and that a receiving trading partner can or cannot deny having received a message, sent by its partner [6].

Exceptions handling: Evaluates the business preparation for every sort of failures, duplications and losses of data.

Claim detection: Evaluates the preparation and support for events of claim loss.

Data transformation: Evaluates the possibilities, tools and technologies for data transformation between collaborating enterprises.

	Criteria definition	*Scale defining*
c1	Defining and describing processes	2 – flexible, simple, compliant with standards 1 – simple and user friendly 0 – basic features only
c2	Collaboration description	2–multiple language, flexibility, safety 1 – safety and basic features 0 – basic features
c3	Role model	1 – yes 0 – no
c4	Small/big/ medium enterprises support	2 – big/medium/small 1 – big/medium 0 – big
c5	Complexity and learning effort	2 – simple 1 – moderate effort 0 – great effort
c6	Efficiency	2 – very efficient 1 – averagely efficient 0 – low efficiency
c7	Maturity	Actual number in years
c8	Tools support	2 – many 1 – medium 0 – low
c9	Synchronous communication support	1 – yes 0 – no
c10	Asynchronous communication support	1 – yes 0 – no
c11	Independency from communication protocols	1 – yes 0 – no
c12	Quality of service	1 – yes 0 – no
c13	Authentication	1 – yes 0 – no
c14	Authorization	1 – yes 0 – no
c15	Integrity	1 – yes 0 – no
c16	Encryption	1 – yes 0 – no
c17	Non-repudiation	1 – yes 0 – no
c18	Exceptions handling	2 – handling message loss, resolution, system recovery 1 – two above 0 – one above
c19	Claim detection	2 – good 1 – average 0 – poor
c20	Data transformation	2 – good 1 – average 0 – poor

Table I: Criteria and scale

4.2 Utility Function

We have defined the utility function, which organizes the results, for them to be comparable (on scale between 0 and 1). In the case, where input value is an actual number, the utility function transforms it to the closed interval from 0 to 1.

Equation 1: Utility function

$$u_j = \sum_{i=1}^{6}\left(\frac{w_i}{100}\cdot\frac{c_i(A_j)}{2}\right) + \left(\frac{w_7}{100}\cdot\frac{c_7(A_j)}{\sum_{k=1}^{N} c_7(A_k)}\right) + \sum_{i=8}^{29}\left(\frac{w_i}{100}\cdot\frac{c_i(A_j)}{2}\right)$$

Equation 2: Maximum utility

$$U = \max_{j=1}^{N}(u_j)$$

Meaning of the symbols:

- U – maximum utility,
- u_j – utility of alternative j,
- c_i – criterion i (Table I),
- A_j –alternative j (ebXML, XLANG, RosettaNet),
- w_i – weight of criterion i,
- N – total number of alternatives.

4.3 Results

For the purposes of the evaluation of the technologies in this article we have selected the weights based on the preferences of a SME, where security (authentication, authorization, integrity, encryption and non-repudiation), defining and describing processes, collaboration support, complexity and learning effort, maturity, tools support, data transformation, exception handling and quality of service are particularly important. The selection of the weights is based on the survey, done in [18]. The weights can however be altered according to the needs and priorities of each distinctive business. The Table II shows the evaluation of ebXML, RosettaNet and XLANG. It is divided in 5 columns. The first column presents criteria. The second column shows the weights, which we assigned to each criterion. The rest of the columns show evaluations for each technology, using the scale, explained in the third column of Table I. In the last row we show the results calculated using the utility function.

As seen in Table II ebXML has achieved the highest result. It turns out that ebXML is the best technology for most of the businesses.

XLANG is second best, although it lacks the quality of service, authentication and non-repudiation. However, it is integrated within the BizTalk Server Initiative, which is very promising. We believe that it will get improved over time.

RosettaNet is the least appropriate for general SMEs. Its main preference lies in technical features and level of development. Since it is the oldest technology of the three, it is the most mature one. Its main disadvantage is in the fact that it is suitable mainly for very large companies, since its framework PIP is very inflexible, and once created, very difficult to alter thus inappropriate for smaller businesses.

c	*w*	*ebXML*	*XLANG*	*RosettaNet*
c1	9	2	2	0
c2	8	2	1	1
c3	2	1	0	1
c4	10	2	2	0
c5	9	1	1	2
c6	7	2	1	1
c7	8	2	2	4
c8	2	1	1	2
c9	2	1	1	1
c10	2	1	1	1
c11	2	1	0	1
c12	6	1	0	1
c13	3	1	1	1
c14	3	1	1	1
c15	5	1	1	1
c16	3	1	1	1
c17	4	1	1	1
c18	6	2	2	1
c19	2	2	1	1
c20	7	2	2	1
		0,735	0,606	0,498

Table II: Evaluation matrix and results

5 Conclusions

The need to do business on the net and to automate business processes is increasing, as is the need for supporting technologies. Such technologies must satisfy certain standards, they must be flexible and available to all organizations, large but particularly to small and medium enterprises. Describing business processes must be relatively simple, so that even non-programmers can use it, since the business process experts usually do not have the necessary knowledge, needed to work with complex languages.

In the article we have identified, compared and evaluated the features of the three most important technologies and upon our findings defined a multi-criteria decision model for their quantitative evaluation. The defined decision model is usable for all kinds of enterprises. They can express their priorities through criteria weights. For the purposes of this article we have also defined a common set of weights for small and medium enterprises and done the evaluation of the technologies. From this perspective

we have determined that ebXML technology is the most suitable with the widest range of possibilities, followed by XLANG and RosettaNet.

References

[1] Selim Aissi, Pallavi Malu, Krishnamurthy Srinivasan (May 2002), E-Business Process Modeling: The Next Big Step, *IEEE Computer*, pp. 55-62.

[2] Alan Kotok, David R.R. Webber (2002), *The new global standard for doing business over the Internet ebXML,* New Riders Publishing.

[3] EbPML.org (2002), XLANG, http://www.ebpml.org/xlang.htm.

[4] David O'Riordan (April 10th 2002), Business Process Standards for Web Services, The candidates, *Services Business Strategies and Architectures,* http://www.webservicesarchitect.com/content/articles/oriordan01.asp

[5] Joe McKee (May/June 2002), RosettaNet at the Dance–an e-business standard does its own choreography, *Oracle technology network*, pp. 51-52.

[6] Pekka Kantola, Janne J. Korhonen (15.5.2002), RosettaNet vs. ebXML–Security Solutions and exception handling, *Helsinki University of Technology*, http://www.soberit.hut.fi/T-86/T-86.161/2002/RosettaNet

[7] Vitria (February 26th 2002), RosettaNet E-Business Process Standards for the High-Tech Industry, *Vitria Technology, Inc.,* http://www.vitria.com/news/press_releases/pr_2002-02-26.html.

[8] EbXML, Technical specifications, http://www.ebxml.org/.

[9] Cover Pages hosted by Oasis (June 6th 2001), XLANG, Technology Reports, http://xml.coverpages.org/xlang.html.

[10] RosettaNet Overview, Background Information, http://www.rosettanet.org/RosettaNet/Rooms/DisplayPages/LayoutInitial.

[11] Madhu Siddalingaiah (August 17th 2001), Overview of ebXML, , *Technical overviews, SUN Microsystems,* http://dcb.sun.com/practices/webservices/overviews/overview_ebxml.jsp

[12] Arsin Corporation (2002), Solution Integration Services, *Arsin RosettaNet PIP Solution,* http://www.arsin.com/docs/RNTFactSheet_final.pdf.

[13] Andy Moir (April 2002), Introduction to RosettaNet, *XML.gov,* http://xml.gov/presentations/rosettanet.

[14] Cover Pages hosted by Oasis (November 2002), RosettaNet, Technology report, http://xml.coverpages.org/rosettaNet.htm.

[15] Microsoft BizTalk Server (2002), BizTalk Accelerator for RosettaNet Features, *Microsoft,* http://www.microsoft.com/biztalk/evaluation/features/rosettanet.asp.

[16] Bea Web-Logic Integration (2002), RosettaNet 2.0 Security Sample, *Bea,* http://edocs.bea.com/wli/docs70/b2bsampl/rn2sec.htm .

[17] XLANG (2001), Web Services for Business Process Design, *2001 Microsoft Corporation,* http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm.

[18] Object Technology Center (2002), Preferences of small and medium businesses, *Technical Report, FERI*

[19] Matjaz B. Juric, S. Jeelani Basha, Rick Leander, Ramesh Nagappan (December 2001), *Professional J2EE EAI*, Wrox Press Ltd.

[20] Arijit Sengupta (2002), Oracle's support for open eBusiness standards, *Oracle corporation,* http://www.idealliance.org/papers/xmle02/slides/Sengupta/sengupta.ppt.

[21] Arijit Sengupta (2002), Data integration, process integration and trading partner agreements, *Oracle corporation,* http://www.edifice.org/ERUG/Sengupta_ERUG.ppt.

[22] Paavo Kotinurmi (Oktober 22nd 2002), Comparing XML Based B2B Integration Frameworks, *Helsinki University of Technology SoberIT,* http://www.soberit.hut.fi/ICTEC/lectures/20021022_Kotinurmi.pdf.