Semantic Interoperability in Electronic Business

Juha Puustjärvi[1]

[1] Aalto University
Espoo, Finland

Abstract

E-business refers to the utilization of information and communication technologies (ICT) in support of all the activities of business. The standards developed for e-business help to facilitate the deployment of e-business. In particular, several organizations in e-business sector have produced standards and representation forms using XML. It serves as an interchange format for exchanging data between communicating applications. However, XML says nothing about the semantics of the used tags. XML is merely a standard notation for markup languages, which provides a means for structuring documents. Therefore the XML-based e-business software is developed by hard-coding. Hard-coding is proven to be a valuable and powerful way for exchanging structured and persistent business documents. However, if we use hard-coding in the case of non- persistent documents and non-static environments we will encounter problems in deploying new document types as it requires a long lasting standardization process. Replacing existing hard-coded e-business systems by open systems that support semantic interoperability, and which are easily extensible, is the topic of this article. We first consider XML-based technologies and standards developed for B2B interoperation. Then, we consider electronic auctions, which represent a form of e-business. In particular, we represent how semantic interoperability can be achieved in electronic auctions.

Keywords: *B2B, Open Systems, Electronic Auctions, Semantic Interoperability, Web Services, Ontologies.*

1. Introduction

Electronic business, or shortly e-business, refers to a wide range of online business activities for products and services. It is usually associated with buying and selling over the Internet, or conducting any transaction involving the transfer of ownership or rights to use goods or services through a computer-mediated network.

Business-to-business (B2B) is a form of e-business. It describes commerce transactions between businesses, such as between a manufacturer and a wholesaler, or between a wholesaler and a retailer. Other forms of e-business are business-to-consumer (B2C) and business-to-government (B2G). The volume of B2B transactions is much higher than the volume of B2C transactions [1].

Automating interoperation is crucial for B2B. By interoperation we refer to the creation of coordination among discrete business activities conducted by different organizations or different groups within an organization, so that a unified business process is formed. It can be implemented at various levels of automation, ranging from partially to fully automated interoperation. The first B2B interoperations were partial point-to-point interoperations and they were not based on any standards. As an example of a fully automated implementation in the case of inventory replenishment, a system can automatically order products from partners, based on current stock levels going below the reorder point. Also, as we will present, the interoperation within electronic auctions can be fully automated.

In B2B all the trading partners must have a shared understanding about how to work together. In particular trading partners must have shared understanding of the

- exchanged business documents,
- business processes, and
- messaging mechanism.

To exchange documents trading partners need a common language through which to exchange documents between their computer systems. HTML, the first-generation language of the Internet, is not suited for this task as it defines only the formatting of information, not its meaning. Instead XML is rapidly becoming the key standard for data representation and transportation. In XML [2] the tags can define the structure of the information, enabling computer tools to use that information directly. However, as XML-documents themselves do not capture any semantics, the introduction of XML-messaging in exchanging business documents requires hard-coding.

By hard-coding we refer to the software development practice of embedding the semantics input-messages into the application program, instead of obtaining the semantics from external sources. Hard-coding is proven to be a valuable and powerful way for exchanging structured and persistent business documents. However, if we use hard-coding in the case of non- persistent documents and non-static environments we will encounter problems in deploying new document types and extending the systems into new domains, e.g., on electronic auctions.

A more flexible way for achieving consensus on exchanged messages is to develop appropriate domain ontology [3], and use it as a vocabulary in exchanging business documents. Essentially the developed ontology must be shared and consensual terminology among the communicating parties as it is used for information sharing and exchange.

Deploying open ontology based interoperable systems for e-business is a challenging problem. In this article, we focus on this problem in the context of Service Oriented Architecture [4]. To begin with, we first characterize the terms integration and interoperability. Then we consider open information systems from autonomy, heterogeneity and dynamicity point of views. After that, we give a short overview of the state of the art with respect to interoperation in B2B by considering the major standardization efforts of e-business. In particular, we give an overview of EDI, ebXML and RosettaNet. Then we consider XML and RDF based messaging from semantics' point of view. In addition, to illustrate semantic interoperation in e-business, we represent a way for implementing semantic interoperation in electronic auction system. After that we shortly discuss the future and emerging trends of semantic interoperability in e-business. Finally, we conclude the article by discussing the advantages and disadvantages of the deployment of semantic interoperability in B2B.

2. Integration and Interoperability

The problem of integrating and interoperability of heterogeneous data sources is not new: the rapid deployment of heterogeneous distributed databases after the 1980s led to the need to share or merge heterogeneous local databases [5]. This required solving the structural and semantic heterogeneity of local databases. Structural heterogeneity caused by the diversity of the data models of the integrated local database systems (e.g., relational and hierarchical data models) as well for the independent design of local databases (e.g., address can be an attribute in a schema and an entity set in another schema) [6].

The integration of data sources means that the data sources are pulled together into one logical data source with a single global schema [7]. Semantic integration means that a new ontology is derived from existing ontologies such that the new ontology facilitates the interoperability of the systems. Nowadays integration is usually not appropriate because it violates the autonomy of the integrated systems. Moreover the problems related to the maintainability of integrated systems make integration undesirable and the interoperability of autonomous systems more appropriate.

The issues related to interoperability are discussed in many articles including [8, 9, 10, 11]. Unfortunately in the literature the terms integration and interoperation is often used interchangeably. The interoperability of two systems means that the systems are able to work together by exchanging information and using the exchanged information. That is, interoperation refers to making applications work together by sharing the appropriate messages but without any single conceptual integration. As e-business frameworks enable the sharing of appropriate messages, but do not usually enable integration, we use the term B2B interoperation though the term e-business integration is commonly used in the literature.

We also make the difference between the terms syntactic interoperability and semantic interoperability. Syntactic interoperability means that the communicating systems are able to exchange structured data such as XML documents. This requires the detection of syntactic errors. In the case of such errors receiving system usually request the sending system to resending the incorrect message.

Beyond the requirements of syntactic interoperability semantic interoperability means that after data were transmitted from a sender system to a receiver, all implications made by one party had to hold and be provable by the other.

There are two thoroughly different approaches for achieving semantic interoperability: hard-coding and semantic messaging.

- By hard-coding we refer to the software development practice of embedding the semantics input-messages into the application program, instead of obtaining the semantics from external sources. Hard-coding requires that the developers of the communicating applications are familiar with the used conceptual scheme (vocabulary) and use that information in developing the communicating applications. Hence, hard-coded systems are able to understand each other as long as they exchange persistent types of documents as the semantics of the messages (exchanged documents) cannot be interpreted by the machines just based on the message and the conceptual schema. However, hard-coding is proven to be a valuable and powerful way for exchanging structured and persistent business documents. In contrast, if we use hard-coding in the case of non- persistent documents and non-static environments we will encounter problems in deploying new document types and extending the system by new participants. Therefore it does not satisfy the goals of open, extensible information systems that support semantic interoperability

- By semantic messaging we refer to the practice of including the semantics of the exchanged document in a machine understandable form in the messages. For semantic interoperability it is necessary to provide standardized ways to describe the meanings of the exchanged documents. Exchanging semantic messages represents an open, easily maintainable and extensible way for developing interoperable open systems.

In this article our analysis of semantic interoperability in the context of electronic auctions is restricted to semantic messaging.

3. Open Systems and Web Services

Open information systems have components that cross organizational boundaries and in this sense are open. This implies that the components in open systems are autonomous and heterogeneous. Further the configuration of the open system can change dynamically. For example, if a component fails, then the failed component can be easily replaced by another component.

Fundamentally components´ autonomy means that they function under their own control, and each component can locally decide how to proceed in its interaction with others. The reason for this is that the components reflect the autonomy of the organization interests that they represent. In addition, there may be technical reasons for the autonomy, e.g., as a result of a hardware failure or error in a software.

In open systems heterogeneity can arise in a variety of formats, e.g., in networking protocols, in encoding information, and in used data models. Heterogeneity may also arise at semantic levels, e.g., the same concept is used for different meanings, or two different concepts are used for the same meaning. The reason for heterogeneity is historical: the components may have arisen out of legacy systems that are initially developed for local uses, but are eventually expanded to participate in open environments. Developing open systems is challenging as the system should cope with the scale of the number of participant and preserve the autonomy of local heterogeneous systems while maintaining coordination over these systems.

Web services [12] provide a methodology that supports open, distributed systems. They are frequently application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. Technically Web services are self-describing modular applications that can be published, located and invoked across the Web. Once a service is deployed, other applications can invoke the deployed service.

There are two ways of using Web services: the RPC-centric view (Remote Procedure Call–centric) and the document-centric view [13]. The RPC-centric view treats services as offering a set of methods to be invoked remotely while the document–centric view treats Web-services as exchanging documents with one another. Although in both approaches transmitted messages are XML-documents, there is a conceptual difference between these two views.

In the RPC-centric view the application determines what functionality the service will support, and the documents are only business documents on which the computation takes place. Instead the document-centric view considers documents as the main representation and purpose of the distributed computing: each component of the communicating system reads, produces, stores, and transmits documents. The documents to be processed determine the functionality of the service. Therefore, document centric view corresponds better with our goal of applying services in open environments.

4. E-Business Frameworks

The standards developed for B2B interoperation, which are also called e-business frameworks, guide the development B2B implementations by specifying the details for business processes, exchanged business documents, and secure messaging.

Even though the interoperation in B2B is nowadays usually based on Web services, it is useful to make a classification of the interoperation/integration approaches [14].

- In Information-oriented approaches applications interoperate through a database or knowledge base. If the database developed by integrating existing databases, then this approach represent B2B integration.

- In Process-oriented (also called workflow–oriented) approach the interoperation is controlled through a process model that binds processes and information within many systems.

- In Service-oriented interoperation applications share methods (e.g., through Web service interface) by providing the infrastructure for such method sharing.

- In Portal-oriented application integration a multitude of systems can be viewed through a single user interface, i.e., the interfaces of a multitude of systems are captured in a portal that user access by their browsers.

4.1 EDI

Electronic Data Interchange (EDI) [15, 16] refers to the transmission of electronic documents between organizations by electronic means. These documents generally contain the same information that would normally be found in a paper document used for the same organizational function. However, EDI is not confined to just business data but encompasses all fields including medicine, transport, engineering and construction.

The first B2B implementations were bilateral private message-oriented solutions which were not based on any standard. The need of common B2B standards strengthened as the amount of private point-to-point solutions that the companies had to maintain increased.

The development of EDI (Electronic Data Interchange) standards for B2B began in 1970's. The first EDI standards versions (X12) were published in 1983. It is most commonly used EDI syntax in North America. The next EDI standard (EDIFACT) originated in 1985. It is dominant EDI standard outside North America.

In the 1970's, when the development of the EDI standards began, messaging information was expensive. Therefore the EDI syntax is very compact in size, which in turn gives rise that EDI documents are hard to read and maintain. However, EDI has advantages over manual business interactions as it reduce paper consumption, eliminates data entry errors, and speed up the transfer of business documents. Newer XML/EDIFACT is an EDI- format that allows EDIFACT message types to be used by XML systems.

Organizations that send or receive documents between each other agree on the specific information to be transmitted and how it should be used. This is done in human readable specifications (also called Message Implementation Guidelines). The EDI standards prescribe the formats, character sets, and data elements used in the exchanged business documents. The complete Document List includes all major business documents, e.g., such as purchase order. The standard says which pieces of information are mandatory for a particular document, which pieces are optional and give the rules for the structure of the document. However, it does not give any semantics for the documents.

4.2 ebXML

The goal of ebXML 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 [17]. The objective of ebXML is to be a global standard for governmental and commercial organizations of all sizes to find business partners and interact with them.

The development of the ebXML started at 1999 and it was sponsored by UN/CEFACT (United Nations centre for Trade Facilitation and Electronic Business) and OASIS (Organization for the Advancement of Structured Information Standards).

The ebXML standard is comprised of a set of specifications designed to meet the common business requirements and conditions for e-business [13]. The CC (Core Components) provides the way business information is encoded in the exchanged business documents. The BPSS (Business Process Specification Schema) is an XML-based specification language that can be used in defining the collaboration of the communicating business partners. However, BPSS is quite limited in that it can only express simple request-response protocols. In addition, BPSS lacks formal semantics, and thereby it cannot be ensured that both communicating parties have the same interpretation of the exchanged documents.

The vocabulary that is used for an ebXML specification consists of a Process-Specification Document, a Collaboration Protocol Profile (CPP), and a Collaborative Partner Agreement (CPA). Process-Specification Document describes the activities of the parties in an ebXML interaction. It is expressed in BPSS. CPP describes the business processes that the organization supports. CPA is technical agreement between two or more partners, and it may have legal bindings. All specifications are stored in ebXML registry.

4.3 RosettaNet

RosettaNet [18] is an industry-driven consortium of information technology, semiconductor manufacturing, and telecommunications companies [19]. Mostly addressed is the supply chain area, but also manufacturing, product and material data and service processes are in scope [19].

RosettaNet develops open e-business process standards, which are based on XML. It defines message guidelines, business processes interface and implementation frameworks for interactions between companies. Its main standardized components are PIPs (Partner Interface Processes), directories and the RNIF (RosettaNet Implementation Framework).

The PIPs are divided into eight clusters noted by numbers. The clusters are further divided into segments noted by letters. For example, 3A4 is for Purchase Order and acknowledgement. Each PIP defines the process of exchanging messages between two partners. However, after a message departs a partner´s computer system, it is not possible to find out whether it was received and correctly processed by the other partner organization. RosettaNet only offers a fixed time-out for the confirmation of each message. In the case that the message is not confirmed in this time, the original partner resends the message.

RosettaNet uses UML diagrams to describe PIPs and relationships among the messages exchanged as part of the PIPs. However, the meaning of the UML diagrams is informal, and hence no direct machine interpretation is possible.

RNIF concerns the messaging. It specifies the business message that contains the business documents specified in PIP, and the necessary headers and security features needed to process the message. RNIF also contains exception-handling mechanism and makes sure that the delivery is non-repudiated.

5. XML-Based Technologies and E-Business Frameworks

We now give a short overview of the key technologies that are required for implementing semantic interoperability in B2B. In particular, we give an overview of XML and its applications RDF and OWL, which are the key technologies in Semantic web, and hence also the key technologies in implementing semantic interoperability.

5.1 XML – Extensible Markup Language

XML (Extensible Mark-up Language) is a metalanguage for defining markup languages. By a metalanguage we refer to a language used to make statements about statements in another language, which is called the object language [2]. There are also XML-based query languages such as XPath, XQuery and XML-DMQL developed for querying XML documents [20, 21].

In XML there is no fixed collection of markup tags such as in HTML. In XML one can define own tags, which is tailored for the information one wish to present. Hence, each XML language (i.e., XML application) is targeted at a particular application domain. However, the languages use the same basic markup syntax, and they all benefit from a common set of generic tools for processing XML-documents.

As a matter of fact, the name "extensible markup language" is misleading in the sense that XML is not a single markup language that can be extended but rather it is a common notation that markup languages can build on [21]. XML is merely a standard notation for markup languages. It serves as an interchange format for exchanging data between communicating applications.

XML says nothing about the semantics of the used tags. Hence, by just presenting exchanged documents in XML does not mean that the applications understand each other. It just provides a means for structuring documents.

5.2 OWL and RDF

The term ontology originates from philosophy, where it is used as the name of a subfield of philosophy, namely, the study of the nature of existence. In such a context it represents a branch of metaphysics [22].

In the context of information technology ontology tries to capture the meaning of a particular subject domain that corresponds to what a human being knows about that domain [23]. It also tries to characterize that meaning in terms of concepts and their relationships. Ontology is typically represented as classes, properties attributes and values [24]. These elements comprise the vocabulary for the exchanged messages.

An ontology language is a formal language used to encode the ontology [3]. There are a number of such languages including OWL (Web Ontology Language) as well as earlier developed ontology languages such as OIL, DAML and DAML+OIL [22, 25].

OWL is a language for making ontological statements, developed as a follow-on from RDF (Resource Description Framework) [24]. It is intended to be used over the World Wide Web, and all its elements (classes, properties and individuals) are defined as RDF resources, and identified by URIs. However, by generalizing the concept of a Web resource, it can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web, e.g., items and their prices that are available from on-line shops.

RDF is a language for representing information about resources in the World Wide Web [22]. It is intended for situations in which this information needs to be processed by applications, rather than being only displayed to people. RDF provides a common framework for expressing this information, and so it can be exchanged between applications without loss of meaning. The ability to exchange information between different applications means that the information represented in RDF may be made available to applications other than those for which it was originally created.

RDF defines a language for describing relationships among resources in terms of named properties and values. The relationship of XML and RDF is that XML provides a way to express RDF-statements. In other words, RDF is an application of XML.

OWL has more facilities for expressing meaning and semantics than XML and RDF, and thus OWL goes beyond these languages in its ability to represent machine interpretable content of the ontology. In particular, it adds more semantics for describing properties and classes, for example relations between classes, cardinality of relationships, and equality of classes and instances of the ontology.

6. Electronic Auctions

In e-business buyers and sellers should be able to interoperate inside an architecture that is easy to use and maintain [26, 27]. Electronic auctions (e-auctions) represent one approach to achieve this goal by bringing together business in the web [28, 29, 30].

E-auction is a system for accepting bids from bidders and computing a set of trades based on the offers according to a well defined policy [31]. Technically, an e-auction is a software system that resides somewhere in the Internet [32, 33].Such systems can provide several types of business processes [34] depending upon their target audience.
Online auctions have turned out to be popular and effective mediums for producing goods and services in both B2B and B2C. The well known auction houses such as eBay, Amazon and Priceline conduct many different types of auctions. The most popular auction types are English, Dutch, first-price sealed bid and second price sealed bid.

In an English auction, the auctioneer begins with the lowest acceptable price and bidders are free to raise their bids successively until there are no more offers to raise the bid or until the end of the auction is reached if there is a time limit . A Dutch auction is the opposite of an English auction in that the auctioneer starts with an initial high price, which is then lowered progressively until there is an offer from a bidder to claim the item.

In sealed auction each bidder (buyer or seller) is given just one chance to bid, and where he or she does not know the amount of other bids. In second-price auction bids are sealed and sent to an auctioneer, and like in sealed auction the highest bidder wins, but the price the winner pays is the price that the second highest bidder has bid.

At the moment most e-auction software is targeted only to B2B procurement. However, there is also a growing interest on new auction formats such as on combinatorial auctions and on multi-attribute auctions. In combinatorial auctions bidders can place offers on sets of items. In multi-attribute auction price is not the only negotiable parameter [35, 36].

Further, in classifying auctions, we can make the distinction between human oriented e-auctions and automated e-electronic auctions. In human oriented e-auctions the auction system communicates with the humans (buyers and sellers) in carrying out the auction. In automated e-auctions buyers' and sellers' software modules (auction agents) communicate with the auction system [37], and auction agents make offers based on the predefined rules. These rules can be classified into three classes [38]: rules that control the admission of bids, rules that control the information revealed by the auction, and rules that control how the auction computes trades.

By automating e-auctions both buyers and sellers can benefit in many ways. For example, they can achieve cost reductions and shorten the duration of the auction processes [39]. The key point in automating e-auctions is that both buyers and sellers have shared understanding of the auction process as well the semantics of the exchanged messages [40]. Shared understanding of the auction processes can be achieved by modeling the auction by process models. For example, we can use BPMN (Business Process Modeling Notation) [41, 42], BPEL [43] or UML activity diagram [44] for representing the flow of activities in auctions.

In e-auction processes bidding and announcing the current state of the auction requires messaging. In early auction systems, the participant received the state information either through text-based e-mail or by checking the Web site of the auction system [33]. However, such solutions are appropriate only in human oriented electronic auctions. The automation of e-auctions requires that when the auction information changes the auction system sends the information to the auction agents that participate to the auction. Understanding this information in a consistent way is crucial because auction agents make their decision based on this information. This in turn requires the introduction of auction ontologies.

7. Auction Ontologies

A feature of ontologies is that depending on the generality level of conceptualization, different types of ontologies are needed. Each type of ontology has a specific role in information sharing and exchange. For example, the purpose of the auction ontology is to describe the concepts of the domain in which auction take place.

An auction ontology describes the concepts and their relationships related to a variety of auction types, e.g., on English auction, combinatorial auction, second-price auction, sealed auction, and multi-attribute auction. To illustrate this, a simple auction ontology is graphically presented in Figure 1. This graphical representation is simplified in the sense that it does not specify cardinalities such as whether an offer may concern one or more items. Neither does it specify the properties of classes such as the identification of a bidder, the type of an auction, or the price of an offer.

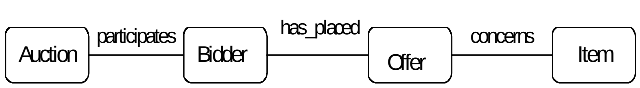

Fig. 1. A Simple auction ontology.

By the term auction instance ontology we refer to the auction ontology supplemented by instances of the ontology. To illustrate this, a simple auction instance ontology is presented in Figure 2. It describes an auction (having ID e254), where B2B-Corpotation has set the offer of $ 85on item p12, and ARGO-Corporation has set the offer of $87 on the same item.

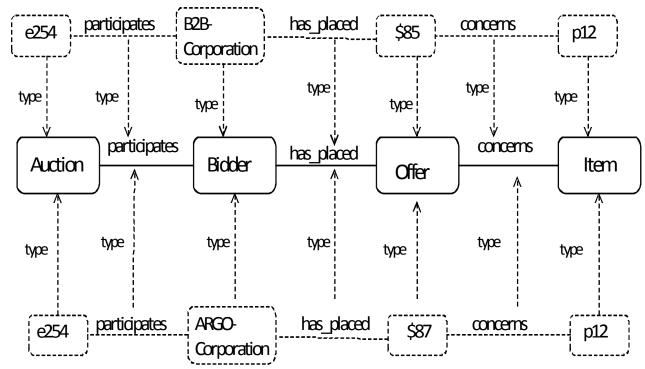

Fig. 2. A simple auction instance ontology.

Depending on the auction types the information concerning the state of the auction process is shared and exchanged. For example, in a sealed auction only the auction system (marketplace) knows all the bids while bidders only know their own bids and the highest bid, and so B2B-Corporation cannot see the bids of ARGO-Corporation, and vice versa. B2B-Corporation's instance ontology includes the value of the highest bid but not any information about its bidder while marketplace's instance ontology includes all the information concerning the state of the auction.

8. Software Architecture

By software architecture we refer to the structure, which comprises software components, the externally visible properties of those components, and the relationships between them [45]. The main goal of the auction software architecture is to allow the reconsideration and redesign of auction processes. This goal is important because the marketplace may be forced to rapidly change the existing auction processes or to develop new ones to better utilize the new information technology. In order to facilitate the changes of the auction processes we can use workflow-technology [46] in implementing the auction engine, which coordinates the auction processes.

The auction system has three types of users: buyers, sellers and the auction system administrator. Figure 3 illustrates the communication structure between the users and the system as well as the communication between the components of the system.

The Ontology managers are implemented by Knowledge Management Systems which are computer based systems for managing knowledge (ontologies) in organizations.

They include a knowledge base, which is a special kind of database for knowledge management. It provides the means for the computerized collection, organization, and retrieval of knowledge for various applications. The BPEL-engine runs Internet-based workflows. They deviate from traditional workflows in that their tasks corresponds the execution of Web services.

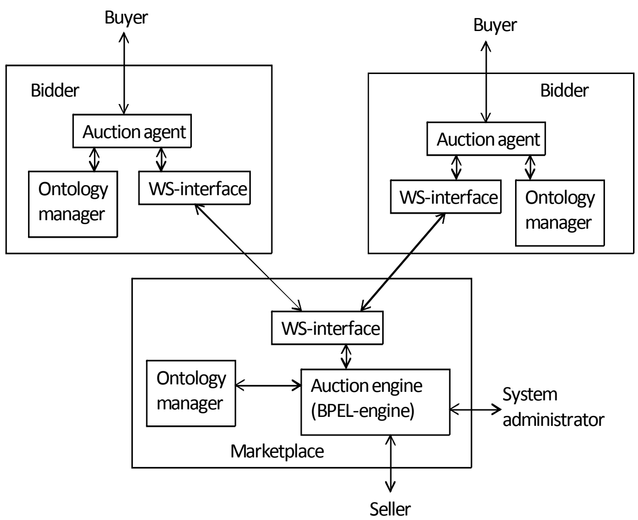

Fig. 3. The components of the system.

System administrator is a person (or a role) who maintains the auction system. In reverse auction (i.e., in procurement) a buyer is the auction initiator, and in other forms of auction seller is the auction initiator.

Basically, there are two approaches how companies can integrate their system with the Auction agent: A company communicates with the Auction agent through a Web service interface, or a company integrates its content management system with the Auction agent. The gain of the first approach is that it has minimal initial costs but has high operational cost as it requires duplication of content management effort. In the second approach the costs are other way around. However this approach is extremely fascinating as it allows (through a Web service) the integration of the ERP-system (Enterprise Resource Planning system) with the Auction agent. In particular, this approach nicely matches with the third wave ERP-systems, which are based on the use of Web services.

9. Auction Processes

We model auctions as business processes. A business process is a series of tasks from one to the next over time [13]. Depending on the specific business process, its tasks can be some combination of services that correspond to queries business transactions, applications and administrative activities. Further, these services may themselves be composite, i.e., they may also be implemented as business processes.

Process description languages [47] have a number of technical requirements. First, they must be able to model process, incorporating the correctness of the execution with respect to the constraint of the real world. Second, the process must be interfaced to underlying functionalities. For example, in the case of Web services, the process must be adapted to the interfaces provided by the Web services.

9.1 Using BPMN for Specifying Auction Processes

The BPMN (Business Process Modeling Notation) is a process description language developed for modeling business process flows and the coordination of Web services. The primary goal of BPMN is to provide a standard notation that is readily understandable by all business stakeholders such as the business analysts, who create and refine the processes, the technical developers responsible for implementing the processes, and the business managers who monitor and manage the processes. Hence, BPMN is intended to serve as common language to bridge the communication gap that frequently occurs between business process design and implementation.

Currently there are several competing standards for BPMN used by modeling tools and processes. From presentation point of view BPMN and the UML 2.0 Activity Diagram from the OMG [47] are quite similar. However, the latter has not adequate graphical presentation of parallel and interleaved processes, which are typical in workflow specifications.

The BPMN defines a Business Process Diagram (BPD), which is based on a flowcharting technique tailored for creating graphical models of business process operations. These elements enable the easy development of simple diagrams that will look familiar to most analysts.

In BPD there are tree Flow Objects: Event, Activity and Gateway: An Event is represented by a circle and it represents something that happens during the business process, and usually has a cause or impact. An Activity is represented by a rounded corner rectangle and it is a generic term for a task that is performed in companies. The types of tasks are Task and Sub-Process. So, activities can be presented as hierarchical structures. In particular, we use activities in representing the scope of transactional constraints. A Gateway is represented by a diamond shape, and it is used for controlling the divergence and convergence of sequence flow.

In BPD there are also three kind of connecting objects: Sequence Flow, Message Flow and Association. A Sequence Flow is represented by a solid line with a solid arrowhead. A Message Flow is represented by a dashed line with an open arrowhead and it is used to show the flow of messages between two separate process participants. An Association is represented by a dotted line with a line arrowhead, and it used to associate data and text with flow objects. We use Association in specifying the selectivity and the criterion of the isolation constraints.

In addition BPMN allows an easy way to connect documents and other artifacts to flow objects, and so narrows the gap between process models and conceptual models. In particular, this feature allows us to incorporate the transactional requirements to BPD-diagrams. Also, a notable gain of BPMN specification is that it can be used for generating executable WS-BPEL code.

Similar to many process modeling methodologies also BPMN utilizes the concept of swimlanes as a mechanism to organize activities into separate visual categories in order to illustrate different functional capabilities or responsibilities.

BPMN supports swimlanes with two main constructions. The two types of BPD swimlane objects are pool and lane. A pool represents a participant in a process. It also acts as a graphical container for partitioning a set of activities from other pools, usually in the context of B2B situations. A lane is a sub-partition within a pool and will extend the entire length of the pool, either vertically or horizontally.

In Figure 4, the message exchange between the marketplace and buyer is presented by two pools. A sequence flow is represented by s solid line, and it is used to show the order that activities will be performed in the auction process. A message flow is represented by a dashed line and is used to show the flow of messages between two separate process participants that send and receive them.

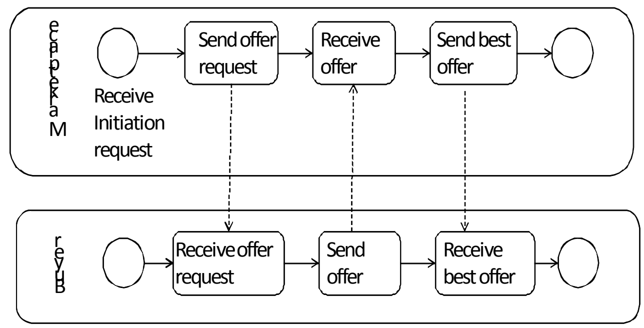

Fig. 4. Using swimlanes for presenting an auction.

9.2 Coordinating Auction Processes

Orchestration is a way of coordinating processes by combining simple Web services to create complex, sequence driven tasks. It views processes from the perspective of one orchestration engine, and so it implies a centralized control mechanism. The orchestration engine itself can also be a Web service.

Orchestration involves creating business logic to maintain conversation between multiple Web services [48, 49]. It can occur between an application and multiple Web services, or multiple Web services can be chained into a workflow, so that they can communicate with one another. Hence, orchestration as a process specification method suits well for workflows coordination.

A workflow specification describes business process tasks and their dependencies, and the requirements these tasks impose on information system functionality and human skills [50]. Hence, auction processes may be realized through workflows. In addition the workflow specification represented by BPD diagrams can be automatically translated into executable WS-BPEL code. In the context of electronic auctions the WS-BPEL code specifies the order in which participating Web services (Buyers and Sellers) are invoked. With WS-BPEL we can also specify conditional behaviors (e.g., whether the offer of B2B Corporation is higher than the offer of ARGO-Corporation).

10. Exchanging Messages

We next illustrate how the ontology managers (located at each site) are able to unambiguously interpret the elements of the exchanged messages, and thus maintain their instance ontologies. First we consider the case where exchanged messages are XML-coded message, and then we consider semantic interoperation where exchanged messages are XML-coded RDF-statements, i.e., RDF-statements are presented by XML-documents.

10.1 XML-based Messaging

Although XML-documents are commonly used for information exchange they do not provide any means of talking about the semantics of data. Instead, it is up to the applications that receive the XML-messages to interpret the nesting of the tags, i.e., the semantics of the messages is hard-coded in communicating applications.

Even if there is a conceptual schema or ontology [2] having the same naming (e.g., classes "offer" and "bidder") as the tags in the XML-message, it is up to the application to interpret the nesting of tags. To illustrate this consider the statement:

"B2B-Corporation has placed offer, which id is OF44 and value is $ 85, on item p12".

We can present this sentence by XML e.g., by the following two nesting ways:

```
<offer id =”OF44”>
    <value> $85 </value>
    <bidder> B2B-Corporation </bidder>
    <item> p12 </item>
</offer>

<offer >
    <id> OF44</id>
    <value> $85 </value>
    <bidder> B2B-Corporation </bidder>
    <item> p12 </item>
</offer>
```

However, the complex element offer does not provide any semantics for the tag nesting. Further, we can present id as an attribute or as an element. That is, there is no standard way of assigning meaning to tag nesting. Therefore the semantics of the messages must be specified by binding the message to an ontology, e.g., to the ontology presented in Figure 1. Such a binding can be done in RDF.

10.2 RDF-based Messaging

Using semantics in information exchange requires that the terms in the ontology are globally unique. This requirement is easily achieved by storing the ontology in the Web and identify it by its address, i.e., by its URL (Uniform Resource Locator) [2]. Hence the ontology can be identified for example by the URL:

http://www.it.lut.fi/ontologies/auction_ontology.

Using this URL as the prefix of an XML-element we can give globally unique names for auction models and their elements. For convenience, however, it is useful to specify an abbreviation for the URL, e.g., *ao*. This can be specified as follows:

```
<xmlns:
ao=”http://www.it.lut.fi/ontologies/auction_ontology” >
```

Now, for example, the element <ao:bidder> is a globally unique name of a class of the auction ontology. Hence, for example the previous natural language sentence can be bind to an ontology and presented in RDF as follows:

```
<rdf:RDF
   xmlns:rdf=
     ”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
    xmlns : xsd=
     ”http://www.w3.org/2001/XMLSchema#”
    xmlns:  ao=
      “http://www.lut.fi/ontologies/auction_ontology#”>
    <rdf:Description rdf:about=”OF44”>
      <rdf:type rdf:resource=“&ao;offer”/>
      <ao:value rdf:datatype=”xsd;integer”>
           85
       </ao:value>
       <ao:item rdf:resource=”p12”/>
       <ao:bidder rdf:resource=”B2B-Corporation”/>
     </rdf : Description>
</rdf:RDF>
```

Now, semantic interoperation can be carried out by including this RDF description in the body part of a SOAP message, and by sending the offer to the Web service of the marketplace.

11. Future Research Directions

Most e-business frameworks suffer from expressiveness and the lack of semantics. E-business frameworks are typically limited to simple request-response protocols. However, many business protocols are quite complex and long lived. Automating such business processes requires that complex interactions can be modeled in business processes. Further, many business processes require that they are executed in an atomic way. For example, assume that a business process (composed Web service) is composed of Flight reservation Web service and Hotel Web service. In this case the success of the hotel reservation may be useless if the flight reservation fails.

The atomicity protocols and transaction models developed for composing Web services tolerate semantic failures (also called logical failures) but not system and communication failures. That is, if a composed Web service fails as a result of fully booked hotel (a semantic failure), then neither the flight will be reserved. However, if the coordinator or the participant of the protocol does not receive a protocol message as a result of communication or system failure, then the execution of the composed Web service will become blocked, i.e., will not terminate (commit or abort).

A way to coordinate the activities of Web services in an atomic way is to provide a Web service which function is to do the coordination. In order to alleviate the development of such coordinators WS-Coordination [13] provides a specification that can be utilized in developing the coordinator. However, such a coordination tolerates only logical failures (e.g., reservation of a hotel room fails logically, if the hotel is fully booked), but it does not tolerates system and communication failures. Recovery from communication and system failures requires the deployment of termination protocols, which are activated when the coordinator or a participant of the commitment protocols fail to receive an anticipated message.

In addition, with respect to semantics, current Web services suffer from the lack of formal semantics. The WSDL (Web Service Definition Language) can only be used for specifying syntactic interoperability, i.e., it does not provide any means for machine understandable description of a Web service. Instead by introducing semantic Web services machines can understand the description of the Web services. Semantic Web Services comprise an ontology layer on the Web services in the layered semantic Web stack. There are multiple standardization efforts (e.g., DAML-S, OWL-S, WSMO and WSDL-S), which aim to define languages for describing relevant aspects of semantic Web services. It is assumed that the deployment of semantic Web Services will reduce the implementations of interoperable B2B processes from months to some minutes.

12. Conclusions

The sophistication of information technology and communications is changing our society and economy. Computers and other electronic devices increasingly communicate and interact directly with other devices over the Internet. Businesses have been particularly quick to recognize the potential and realize the benefits of adopting new computer-enabled networks. Businesses use networks even more extensively to conduct and re-engineer production processes and streamline procurement processes.

In order to automate B2B trading partners need a common language through which to exchange documents between their computer systems. Although XML is rapidly becoming the key standard for data representation and transportation XML-documents themselves do not capture any semantics. Therefore the introduction of XML-messaging in exchanging business documents requires hard-coding, which fights against the principles of open B2B networks. Instead exchanging semantic messages represents an open, easily maintainable and extensible way for developing interoperable B2B systems.

In this article, we have considered how to exploit semantic web technologies in developing open interoperable B2B systems. A challenging situation for the businesses is also the introduction of new technologies. The introduction of the semantic interoperation in B2B is challenging as it incorporate semantic web technologies into many tasks in organizations, including information production, presentation, and processing. The introduction of these technologies also changes the daily duties of the many ICT-employees of the organization. Therefore the most challenging aspect will not be the technology but rather changing the mind-set of the ICT-employees and the training of the new technology.

The introduction of a new technology for B2B is also an investment. The investment includes a variety of costs including software, hardware and training costs. Training the staff on new technology is of prime importance as the incorrect usage and implementation of a new technology, due to lack of proper training, might turn out to be expensive in the long run.

The corner stone of our developed solution is the auction ontology on which the communicating parties have to commit in their mutual communication, i.e., the used ontology must be shared and consensual terminology as it is used for information sharing and exchange. It, however, does not suppose the introduction of a universal ontology for B2B. This situation is analogous with natural languages: a company, or any business organization, may communicate in French with authorities and in English with business companies. Just as there is no universal natural language, so there is no universal ontology.

References

[1] Turban, E., King, D., Viehland, D., and Lee, J. (2006). Electronic commerce. PrenticeHall.
[2] Harold, E. and Scott Means, W. (2002). XML in a Nutshell. O'Reilly & Associates.

[3] Gruber, M. and Thomas R. (1993).Toward principles for the design of ontologies used for knowledge sharing. Padua workshop on Formal Ontology.
[4] Singh, M.P. (2004). Practical Handbook of Internet Computing, Chapman Hall & CRC Press.
[5] Ullman, J. and Widom, J. (1998). Principles of Database Systems. New Jersey: Prentice Hall.
[6] Baeza-Yates, R.and Ribeiro-Neto, B.(1999).Modern Information Retrieval. New York: Addison Wesley.
[7] Garcia-Molina, H., Ullman, J. and Widom, J. (2000). Database System Implementation. New Jersey: Prentice Hall.
[8] Natalya F. N. (2004). Semantic integration: a survey of ontology-based approaches, ACM SIGMOD Record, Vol.3, No. 6.
[9] Cruz I. F. and H. Xiao, X. (2005). The Role of Ontologies in Data Integration , Journal of Engineering Intelligent Systems 13(4). pp. 245-52.
[10] Mayo, E. and Steinberg, T. (2007).The Power of Information: an Independent Review. Available at: http://www.opsi.gov.uk/advice/poi/power-of-information-review.pdf
[11] Stasinopoulou , T., Bountouri, L., Kakali, C., Lourdi, C., Papatheodorou, C., Doerr M. and Gergatsoulis, M. (2007). Ontology-based metadata integration in the cultural heritage domain. In: D.H.-L. Goh et al. (eds), Lecture Notes in Computer Science, ICADL 4822 (Springer, Berlin/ Heidelberg, 2007) 165-75.
[12] Newcomer E. (2002). Understanding Web Services. Addison-Wesley.
[13] Singh, M.P. and Huhns, M.N. (2005). Service-Oriented-Computing, Jon Wiley & Sons.
[14] Lithicum, D. (2004). Next Generation Application Integration. Boston: Addison-Wesley.
[15] Foxvog, D. and Bussler, C. (2006). Ontologizing EDI Semantics. ER 2006 Workshop on Ontologising Industrial Standards (OIS), pp. 301-311.
[16] Iacovou, C. L., Benbasat, I. and Dexter, A. S. (1995). Electronic Data Interchange and Small Organizations: Adoption and Impact of Technology. MIS Quarterly, Vol. 19(4), pp. 465-485.
[17] Dogac, A., Tambag, Y., Pembecioglu, P., Pektas, S., Laleci, G. B. , Kurt, G., Toprak, S. and Kabak, Y. (2002). An ebXML infrastructure implementation through UDDI registries and RosettaNet PIPs. ACM Sigmod International Conference on Management of Data. pp. 512-523.
[18] RosettaNet consortium, Rosettanet Homepage, URL: http://www.rosettanet.org/, Accesses 8/12, 2009.
[19] Damodaran, S. (2004). B2B integration over the Internet with XML: RosettaNet successes and challenges, 13th International Wold Wide Web conference on Alternatr track papers & posters. pp. 188-195.
[20] Feng, L. and Dilon T. (2005). An XML-enabled data mining query language: XML-DMQL. Int. J. Business Intelligence and Data Mining, Vol. 1, No.1, pp. 22-41.
[21] Moller, A. and Schwartzbach, M. (2006). An Introduction to XML and Web Technologies. Addison Wesley.
[22] Antoniou, G. and F. Harmelen, F. (2004). A semantic web primer. The MIT Press.
[23] Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. Scientific American. Vol. 284(5), pp 34-43.
[24] Davies, J., Fensel, D. and Harmelen, F. (2002). Towards the semantic web: ontology driven knowledge management. John Wiley & Sons.
[25] Daconta, M. Obrst, L. and K. Smith K. (2003). The semantic web. John Wiley & Sons.
[26] Siddiqui, B. (2002). UDDI-based electronic marketplace, in Web Services Business strategies and architectures'. Ed. Flechter, P and Waterhouse, M., Expert Press.
[27] Sashi, C. M. and O'Leary B. (2002).The Role of Internet Auctions in the Expansionof B2B Markets. Industrial Marketing Management.
[28] Elliot, A. C. (2000). Getting Started in Internet Auctions. John Wiley & Sons.
[29] Klein, S. and Keefe, R. (1999). The Impact of Web on Auctions. International Journal of Electronic Commerce, Vol. 3, No. 3.
[30] Krishna, V. (2002). Auction Theory. Academic Press.
[31] Rothkopf, M. H. and Park S. (2001). An Elementary Introduction to Auctions' Interfaces. Interfaces, Vol. 31, No. 6, pp. 83-97.
[32] Esmahi, L. and Bernand, J. C. (2000). MIAMAP: A Virtual Marketplace for Intelligent Agents. Proceedings of the 33rd HICSS.
[33] Wurman, P. (2005). Online Auction Site Management. http://www.csc.ncsu.edu/faculty/wurman/Papers/Wurman-article.pdf.
[34] Grigoria, D., Casatib, F. et al. (2004). Business Process intelligence. Computers in Industry, Vol. 53, pp. 321-343.
[35] Nisan, N. (2000). Bidding and allocation in combinatorial auction. In proc of the Second ACM Conference on Electronic Commerce.
[36] Hoos, H. and Boutlier, C. (2001). Bidding Languages for Combinatorial Auctions. Proceedings of the 17th International Joint Conference on Artificial Intelligence.
[37] Esmahi, L. and Bernand, J. C. (2000). MIAMAP: A Virtual Marketplace for Intelligent Agents. Proceedings of the 33rd HICSS.
[38] Wurman, P. Wellman, P. M. and Walsh, W (2005). Specifying Rules for Electronic Auctions. http://www.csc.ncsu.edu/faculty/wurman/Papers/Wurman-article.pdf.
[39] Wurman, P. Wellman, P and Walsh, W. (2001). A parametrization of the auction design space. Games and Economic Behavior. Vol. 35, No. 1-2.
[40] Puustjärvi, J. (2005). Coordinating electronic auctions by workflow systems. Proceedings of the IADIS International Conference WWW/Internet, pp. 139-142.
[41] BPMN (2004) Business Process Modeling Notation (BPMN), http://www.bpmn.org/.
[42] Smith, H. and Fingar, P. (2003). Business Process Management (BPM): The Third Wave, Meghan-Kiffer Press.
[43] BPEL (2002) Business Process Execution Language for Web Services, http://www.ibm.com/developerworks/library/specification/ws-bpel/.

[44] UML (2007). Introduction to OMG's Unified Modeling Language. http://www.omg.org/gettingstarted/what_is_uml.htm

[45] Chalmeta, R.; Campos, C. and Grangel, L. (2001). Reference Architectures for Enterprise Integration. Journal of Systems and Software, Vol. 57(3), pp. 175-191.

[46] Marinescu, D. C. (2002). Internet-Based Workflow Management, John Wiley & Sons.

[47] Sommerville, I. (2004). Software Engineering. Addison Wesley.

[48] Cardoso, J. and Sheth, A. (2003). Semantic e-workflow composition. Journal of Intelligent Information Systems (JIIS), Vol. 21, No. 3, pp 191-225.

[49] Cardoso, J., Miller J. et al. (2004). Modeling quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web Journal, Vol. 1, No. 3, pp. 281-308.

[50] Cardoso, J. Lenic, M. (2006). 'Web process and workflow path mining using the Multimethod approach. Int. J. Business Intelligence and Data Mining, Vol. 1, No.3, pp. 304-328.

Juha Puustjärvi obtained his B.Sc. and M.Sc degree in computer science in 1985 and 1990, respectively, and his PhD degree in computer science in 1999, all from the University of Helsinki, Finland. Since 2003 he has been the professor of information society technologies at Lappeenranta University of Technology. Currently he is an adjunct professor of eBussines technologies at the Aalto University, and an adjunct professor of computer science at the University of Helsinki. His research interests include eLearning, eHealth, eBussines, knowledge management, semantic web and databases.