ebXML: Status, Research Issues, and Obstacles

Birgit Hofreiter, Christian Huemer, Wolfgang Klas
Institute for Computer Science and Business Informatics, University of Vienna
{birgit.hofreiter, christian.huemer, wolfgang.klas}@univie.ac.at

Abstract

Technologies and approaches in the field of Electronic Commerce are not mature enough in order to allow for their broad successful commercial application. Neither existing, successful approaches that are very much restricted to large companies, specific branches, and business domains, nor approaches just merging new technologies like Internet, WWW, and XML allow for scaling up electronic commerce by means of arbitrarily high numbers of partners. All these approaches lack substantial reflection and integration of business semantics as the basis of any electronic commerce partnership. ebXML is a world-wide initiative that tries to address the drawbacks of existing standards and approaches and has the potential to successfully deliver solutions to these problems. In this paper we address the status of ebXML and identify open research issues to be solved in order to meet some of the obstacles on the way of a commercial application of ebXML.

1. Introduction

The field of Electronic Commerce obviously can be seen from different perspectives, including the following viewpoints:

One can perceive the field as a hype of new, not very well-understood approaches for doing business electronically, which generated the (timely limited) age of the .com New Economy. In this viewpoint one may come up with the simplified, data-centric formula: *Internet + WWW + XML + some XML vocabulary + some new business model = the new way of doing business*. Approaches that are based on this model are for example BizTalk [15], OAG [17], cXML [1]. Observing the performance of the New Economy recently it looks like that this model to some extend lacks some substantial components in order to be applied successfully by commercial means.

Given the fact that commercially successfully implemented standards like ANSI X12 (since 1979), EDIFACT (since 1987), and SWIFT (since 1973) for exchanging data to do business have already been established long time ago, the field of Electronic Commerce could be seen as an area of nothing spectacular new, but as a remake of approaches trying to merge new technologies in order to scale up by means of businesses being able to participate in ad-hoc networks of partners performing business electronically. The simplified, data-centric formula for this could be: *EDIFACT concepts + XML + Internet + adapted existing business models = the new way of doing business*. In this model the only new aspect seems to be XML, a new way of organizing data compared to the way of organizing data by EDIFACT. A prominent example of such an approach is xCBL [2, 10] which is based on EDIFACT and ANSI X12. In 1999, due to the growing popularity of XML, United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) also started to look for an XML solution which should be compatible with existing EDIFACT to protect the huge investments taken world-wide. UN/CEFACT's Techniques and Methodology Working Group (TMWG) was responsible for doing a feasibility study on using XML for B2B information transfer. The TMWG report on this subject rejected the idea of creating 'Yet Another XML Solution' by converting EDIFACT to XML. This decision was mainly based on the fact that a syntactical transformation would hardly solve any EDI problem, but would just add another e-business vocabulary to the XML world.

Following up data-centric approaches mentioned above one can identify approaches trying to include or focusing on process-centric methods, i.e. business processes are becoming a central focus of Electronic Commerce models. Examples of this model are RosettaNet [18] and Open-edi [13]. In fact, the TMWG report mentioned above actually recommended to build up on the Open-edi reference model by using business process modeling to define the business aspects and by using XML as key concept for the technological aspects of e-business transactions. Additionally, TMWG suggested to cooperate in the development of the solution with the IT-industry to combine UN/CEFACT's business know-how with the experience of leading XML experts. The steering committee of UN/CEFACT accepted the TMWG recommendation and found an IT-partner in the Organization for the Advancement of Structured Information Standards (OASIS) that shares the goal of open and inter-operable standards. This was the starting point for the ebXML [14, 16] initiative, which started in November 1999. The approach taken by ebXML can be characterized by the simple formula *EDI + XML + Business Process Models including business objects = ebXML*.

In the following, the paper gives a short introduction to ebXML. Then we analyze and explain ebXML by means of the layers and their roles in the ebXML application scenario, the status of implementation, and open research topics.

2. An ebXML primer

The vision of ebXML is to create a single global electronic marketplace where businesses can find each other, agree to become trading partners, and conduct business. All these operations will be performed automatically by exchanging XML documents. In order to support the needs of SMEs, ebXML envisions that software industries will deliver commercial off-the-shelf software (COTS) for B2B scenarios to the SMEs. This goal is expressed in the following, typical ebXML scenario between a large corporation (Company A) and a SME (Company B) as illustrated in Figure 1. This scenario is described in the ebXML technical architecture specification [9]: Company A requests business details from the ebXML registry (step 1) and decides to build its own ebXML-compliant application. Company A submits its own business profile information to the ebXML registry. The business profile submitted to the ebXML registry describes the company's ebXML capabilities and constraints, as well as its supported business scenarios. Company B, which uses an ebXML-compliant shrink-wrapped application, discovers the business scenarios supported by Company A in the registry (step 4). Company B sends a request to Company A stating that they would like to engage in a business scenario (step 5). Before engaging in the scenario, company B submits a proposed business arrangement directly to Company A's ebXML-compliant software interface. The proposed business arrangement outlines the mutually agreed upon business scenarios and specific agreements. Company A then accepts the business agreement. Company A and B are now ready to engage in e-business using ebXML (step 6).

To support the ebXML scenario described above the ebXML specifications describe a way to define business processes and business documents that are exchanged to support these processes. Accordingly defined business processes and documents may be made public in a registry. ebXML specifies a mechanism to register and discover processes and documents. The total set of registered business processes in a registry defines the possibilities in an e-business world. Each organization participating in the e-business world may define its

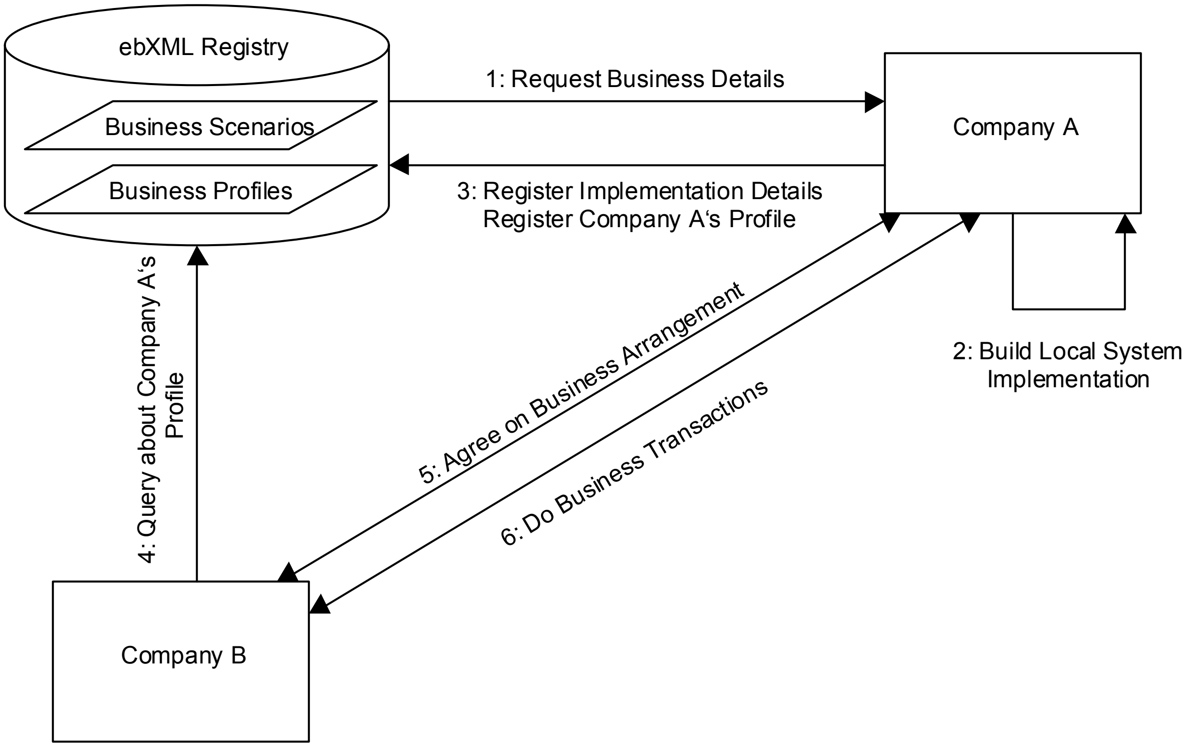
ebXML Registry
Business Scenarios
Business Profiles
1: Request Business Details
Company A
3: Register Implementation Details
Register Company A's Profile
2: Build Local System Implementation
4: Query about Company A's Profile
5: Agree on Business Arrangement
6: Do Business Transactions
Company B

Figure 1. ebXML scenario

capabilities (IT capabilities, communication protocols, security requirements, supported business processes) as a subset of what is possible. These company profiles called collaboration protocol profiles may be stored in a registry as well. This allows companies to query possible business partners and the way to conduct business with them. Before business partners can actually do business with each other they should build a trading partner agreement. This collaboration protocol agreement corresponds to an intersection of their profiles and includes additional results of negotiating variable parameters. In addition, ebXML defines a transport and routing layer to move the actual XML business documents between trading partners.

Accordingly, ebXML is structurally based on subsequent layers as shown in Figure 2. On top there are core components (CC), which will be assembled into business documents. The definition of business processes on the next layer will refer to business documents / core components supporting a single step in the choreography of a business process. The trading partner definitions on the next layer include a collaboration protocol profile that refers to the business processes and the roles therein a certain company is capable of. Furthermore, the layer includes collaboration protocol agreements that are formed by the intersection of individual collaboration protocol profiles. The registry and repository on the next layer must be able to register core components, business documents, the choreography of business processes, and collaboration protocol profiles. Each of the registry and repository must be assigned with a unique identification in order to allow a referencing mechanism as described above. The bottom layer of transport and routing has to ensure the messaging services needed for exchanging business documents at runtime.

3. ebXML status and research issues

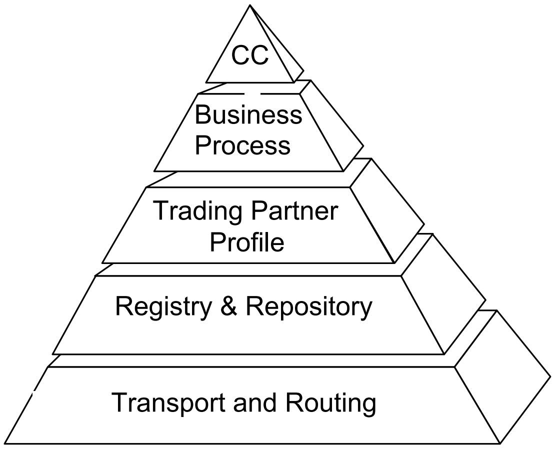

Figure 2. ebXML deliverables

Having introduced the basic concepts of ebXML, we go into more detail to illustrate the status of the ebXML specifications and to derive some obstacles and further research issues. In the following subsections, we concentrate on each of the layers of the pyramid depicted in Figure 2. Each subsection will have the same structure: We give a basic explanation of the respective layer and refer to the current status of implementation, followed by the responsibilities of the layer in the ebXML scenario presented in Figure 1. Based on this information we address unsolved problems and future research topics.

3.1. Transport and routing

Basic explanation - The ebXML Message Service Specification 1.0 [6] defines a wire format and protocol for exchanging XML-based e-business messages. It defines a communication protocol neutral method. Nevertheless, it describes how to use the specification with HTTP and SMTP. The specified enveloping constructs support reliable and secure delivery of business information. The specification covers the definition of an ebXML message structure used to package payload data for transport between parties and the behavior of the message service handler that sends and receives those messages over a data communication protocol. Although XML business documents are expected to be payload the technique permits any format type.

The ebXML message service is defined as a set of layered extensions to the base Simple Object Access Protocol (SOAP) and SOAP Messages with Attachments [20]. It extends these protocols by security and reliability features necessary to support international e-business. An ebXML message is a MIME/Multipart message envelope containing the header container, a SOAP 1.1 compliant message, and zero or more MIME parts containing application level payloads. The header container consists of the SOAP Envelope element. The SOAP Envelope element consist of a SOAP Header element with ebXML-specific header elements and a SOAP Body element containing message service handler control data and information related to the payload.

A possible implementation of the ebXML message service architecture will require the following functional modules: Header Processing, Header Parsing, Security Services, Reliable Messaging Services, Message Packaging, Error Handling, Message Service Interface (used by applications to interact with the message service handler)

Status - The ebXML Message Service Specification 1.0 is a quite solid specification that is ready to be implemented. Some organizations like the Open Travel Alliance (OTA), RosettaNet, etc. have started to incorporate this specification into their standards.

Scenario - With respect to the introduced ebXML scenario the ebXML Messaging Service supports the communication of a business partner with both the registry (steps 1, 3, and 4) and another business partner (steps 5 and 6). It is important that an ebXML-compliant system either a self-built solution (step 2) or a commercial off-the-shelf product supports this ebXML Messaging Service specification.

Research issues - The Messaging Service Specification does not leave any open issues for research. Success depends solely on the acceptance of the user community.

3.2. Registry and repository

Basic explanation - An ebXML registry provides a set of services that enables sharing of information between parties participating in an ebXML transaction. ebXML registry services maintain the information as objects in a repository. The registry has to support the information needs of ebXML trading partners. It has to manage information such as collaboration protocol profiles of trading partners, core libraries, business libraries, business processes and business documents. However, all the artifacts controlled by a registry and stored in the repository are nothing else than objects.

The ebXML Registry Service Specification 1.0 [8] defines the syntax and semantics of the registry interfaces. It uses two subservices to manage the objects in the registry: the object management service and object query management service. The object management service provides the functionality required by registry clients to manage the life cycle of repository items. The object query management service allows a client to search for a registry entry.

The entries of a registry are stored according to the ebXML Registry Information Model 1.0 [7]. This information model does not deal with the actual content of the repository. All elements of the information model represent meta data of the content and not the content itself. Software vendors may use this information model when creating ebXML-compliant software.

It is a prerequesite that all objects in the registry have a unique identification. The identification must be a universally unique identifier (UUID) and must conform to the format of a URN that specifies a DCE 128 bit UUID. An object’s id is usually generated by the registry and may be used by other objects to reference this object.

Status - The ebXML Registry Service Specification 1.0 and the Registry Information Model 1.0 are currently implemented in prototype registries, e.g. Sun ebXML Registry/Repository Implementation (Preview Release: July 2001) and the Korea Trade Network (KTNET).

Scenario - Looking at the ebXML scenario a company must be able to request the business details from the registry (step 1). This means that a company gets information about registered business process definitions, their choreography and supporting business documents. In this context it is important to limit the resulting business information to certain business criteria. A company participating in ebXML publishes its roles in supported business processes as well as its own implementation details (step 3). Just as important is the registration of the company itself and of the products and business services the company offers (step 3). A possible future business partner can now search the registry for companies offering a desired product/service and supporting a shared business process (step 4).

Research issues - The specification of an ebXML-compliant registry is quite generic and flexible. This is due to the fact that all entries are nothing else than objects from a registry’s point of view. The rather generic information model makes it easy to enter information in the repository, but it is hard to retrieve exactly the information needed. It is comparatively easy to find the right type of object (e.g. business process); nevertheless, to retrieve the objects of interest fulfilling certain criteria is a challenge. For this purpose, the registry information model offers to classify each entry. In this respect ebXML makes use of coding schemes existing in various industries to provide a structured coded vocabulary to define the context of industry, product, region, process, role, etc. ebXML does not define any specific coding schemes, it provides the capability of refering to them. In other words, ebXML offers nothing else than a structure of name-value pairs to classify objects. Future research work has to include a registry environment that allows the bridging of different taxonomies based on concepts in the field of ontologies. Furthermore, useful policies for describing certain object types with certain classification types must be developed. In this respect meta classification schemes might be of interest to inform business partners about the way they should describe their objects in order to ensure interoperability on this level. Another research topic should consider alternative approaches to the name-value pair structures to express more specific and realistic criteria for classification.

In a real-world business environment a single physical registry that is centrally controlled by a standard organization does not seem to be practical. Instead, a powerful system of distributed ebXML registries is desirable, allowing each registry to interact with and cross-reference another one. Yet, the current ebXML specification does not address this issue. In this context the harmonization of objects stored in different registries is one of the most important problems to be solved. The question is how to avoid in a timely manner a proliferation of semantically equivalent objects stored in different repositories with different identifications. Solutions automatically performing this check would be highly desired. Furthermore, a strategy for an optimal balance between multiple copies of and references to registered objects is of interest. This goes along with the need for fault-tolerant systems of ebXML registries.

Another important point - not even mentioned by ebXML - refers to the privacy issues of the registries. Due to the fact that companies have to publish their profiles including their supported business processes to take full advantage of the ebXML architecture, some critics claim that companies will never offer such sensitive information to the public. Thus, access to at least some of the information stored in a registry must be restricted to a certain degree. Conservative policies would allow access to the profiles only to business partners in an established extranet. This prevents ad-hoc business relationships between business partners unknown to each other before. Alternatively, access to the profiles may be granted to 'qualified' partners only. 'Qualified' does not necessarily mean that previous agreements have been made, but that a company fulfills certain criteria. Suitable policies and their technical implementation are considered important future research topics in the world of 'trusted' ebXML commerce. Note, that the secure communication between clients and the registry are important as well. Though, ebXML specifications and technical reports cover this topic.

Another challenge for ebXML is to provide alignment of ebXML and UDDI registries. The latter seems to become the registry type of choice for industry. The fundamental difference between UDDI and ebXML is that UDDI is aiming to create a standard registry for companies that will accelerate the integration of systems around marketplaces, while ebXML is working to standardize how XML is used in general B2B integration [21]. Although UDDI might be complementary to ebXML, some registry concepts are semantically equivalent. Therefore, an open registry architecture allowing a seamless integration of both, and may be other types of B2B registries, need to be approached in a research project.

3.3. Trading partner profile

Basic explanation - The ebXML Trading Partner Profile and Agreement Specification [4] provides definitions for documents called trading partner profile (TPP) and trading partner agreement (TPA). A business partner's capabilities (commercial/business and technical) may be described in the TPP. The agreed intersections between the profiles of two business partners are documented in a TPA.

The message exchange capabilities of a business party may be described by a collaboration protocol profile (CPP) within the TPP. And in turn the message exchange agreement between two parties may be described by a collaboration protocol agreement (CPA) within the TPA. The exchange of information between two parties requires each party to know the other party's supported business collaboration, the other party's role in the business collaboration, and the technology details about how the other party sends and receives messages. This means that details of transport, messaging, security constraints, and bindings to a business process specification document containing the definition of the two parties are included in the CPP and CPA.

The CPP describes the capabilities of an individual party. So a CPA is a document that represents the intersection of two CPPs and is agreed upon by both trading partners, i.e. CPA describes the agreement to be used to perform particular business collaborations.

Status - The ebXML Collaboration Protocol Profile and Agreement Specification 1.0 has been tested in the proof-of-concept sessions. However, real-world ebXML interchanges have to proof its suitability and practical applicability. Therefore, experience reports from software vendors and user communities are still missing. Since the ebXML work on trading partner profiles and agreements started off the work of IBM's Trading Partner Agreement Markup Language (tpaML) [19], experiences gained from tpaML could be considered as an indicator of its usability.

Scenario - In the ebXML scenario companies register their capabilities as profiles in the registry (step 3). Consequently, possible future business partners are able to query these profiles or, in other words, find business partners according to certain characteristics in their profiles. If the profiles of two trading partners match they can form a business agreement as basis for their future business transactions (step 5).

Research issues - The current specifications related to trading partner issues are limited to the technical aspects of the ebXML runtime environment focusing on the message exchange capabilities. However, real-world (e-)business requires first of all commercial and legal agreements. Thus, DTDs or schema definitions for trading partner profiles and agreements covering all commercial and legal aspects in addition to the CPP's and CPA's runtime aspects are needed. This should lead to the definition of ebXML-conform contracts. Research work has to identify the semantics to be covered in these trading partner profiles and agreements and has to transform it into an XML-like representation. We assume that there does not exist a single type of contract to be applied for all e-business transactions. Nevertheless, we feel that e-business transactions can be grouped into different categories leading to a definite set of ebXML-conform contracts. The challenge is to identify these different categories and to define a common contract pattern for each of them to be reused in e-business transactions belonging to them.

According to the ebXML scenario two companies agree on their trading partner agreement in a single step (step 5). It is mentioned that this step is based on standard or mutually agreed trading partner agreements. Since trading partner agreements represent the intersection of the capabilities specified in the profiles of each partner, we feel that mutually agreed ones are of more practical relevance. The challenge for research is to identify mechanisms to compute the intersections leading to automatic negotiations.

Furthermore, we believe that the agreement process itself must be regarded as a continuing feed-back loop until consensus is reached, rather than a single step. A single step process might be sufficient for CPAs, but not for agreements concerning commercial and legal matters. Therefore, previously mentioned patterns for e-business contract categories should not only address the static parts of a contract, but also patterns for their negotiation process. It is evident that a negotiation process is itself a certain type of an e-business transaction. Consequently, it should be described by the means of ebXML business process specifications. This guarantees the runtime execution of ebXML-conform negotiations by the exchange of ebXML messages.

An obvious limitation of the current approach to CPP and CPA are the hard references ('all or nothing') to other ebXML artifacts, like business processes and documents. In other words this means that a CPP or CPA can only reference whole business processes or roles therein and whole business documents. If a company publishes in its profile that it supports a certain role in an identified business process, this means that it supports all the activities assigned to this role and all the business documents assigned to these activities. There is no way to specify a company's profile on the business activities in the business process and the business documents. Since the specifications of business processes and documents provide the opportunity for optional elements, profiling these optional elements would guarantee exact definitions of business transactions between business partners. It should be noted that the high degree of optionality of traditional EDI messages caused the main problems in EDI and not the syntax. Consequently, the profiling of business processes and business documents is a research topic of primary interest. However, the employment of such profiles would also demand more sophisticated mechanisms for the automatic negotiation of CPAs.

Trading partner profiles are by their nature documents including a lot of optional elements. Once economic and legal issues are covered by the profiles, the situation will get even worse. In order to simplify a negotiation process between companies previously unknown to each other it would help to identify those elements the negotiation process should concentrate on. This requires some sort of meta profile that defines the expected structure of the partner's profile to allow the computing of the intersection in the negotiation process. In such an approach a company would register the expected partner's meta profile together with its own profile in step 3 of the ebXML scenario. A possible business partner can retrieve the meta profile in step 4 and knows the type of parameters to send in the negotiation process of step 5.

The deployment of ebXML-conform contracts is also dependent on its legal binding. This means that on the one hand side appropriate and globally accepted rules of law for ebXML-conform contracts have to be established. On the other hand technical solutions (e.g., electronic notaries) based on existing ebXML messaging specifications for reliable and secure delivery have to be developed.

3.4. Business processes

Basic explanation - Business process models describe interoperable business processes that allow business partners to collaborate. Business process and information modeling is not mandatory in ebXML. However, if implementers and users select to model business processes and information, then they shall use the UN/CEFACT Modeling Methodology (UMM) that utilizes UML [9]. The UMM meta model describes the business semantics that allows trading partners to capture the details for a specific business scenario using a consistent modeling methodology. A business process describes in detail how trading partners take on shared roles, relationships, and responsibilities to facilitate interaction with other trading partners. An interaction between roles follows a choreographed set of business transactions. Each business transaction is expressed as an exchange of electronic business documents. The sequence of the exchange is determined by the business process and by messaging and security considerations. To summarize, UMM covers a specification of the semantics and artifacts that are required to facilitate business process and information integration and interoperability.

Software components on each partner's side have to support the business process models. A full UMM model contains more information than needed to configure ebXML-compliant software. Thus, the ebXML business process specification schema (BPSS) adopts a subset of UMM for this particular purpose [3]. It provides a set of elements necessary to specify the choreography of interactions between business partners. Furthermore, it provides configuration parameters for the partner's runtime systems to execute these interactions. The BPSS is available in UML and XML. The XML version is intended to be interpretable by ebXML-compliant software.

Status - UMM has been developed by UN/CEFACT's Techniques and Methodologies Working Group (TMWG) since 1998. However, an UMM version [11] was merged with the RosettaNet methodology [18] during the ebXML initiative. The latter has proved its practical use by being used as underlying methodology to develop RosettaNet's Partner Interface Processes (PIPs). The current UMM version (Revision 10) will be used by UN/CEFACT for their ongoing work to standardize business processes and business documents.

From an implementation point of view the BPSS is in a preliminary state. However, it has been tested in proof-of-concept demonstrations. It is expected that BPSS supporting software components will be part of products of major software vendors released in 2002.

Scenario - In the ebXML scenario a company first requests existing business process definitions (step 1). The company chooses those business process they want to support. It has to implement interfaces for these business processes into its own applications in compliance to the BPSS definition (step 2). Note that a company can also define its own business processes supported by their applications and register them in the registry (not depicted in the ebXML scenario). Following the current approach the subsequent steps only reference the business process definition. The company registers its profile with references to supported business processes (step 3). A possible business partner might find the company according to references to business processes supported by both in complementary roles (step 4). They form a trading partner agreement on a certain number of business processes by referencing these processes (step 5). Finally, they run business transactions (step 6) in accordance with the business processes' definitions.

Research issues - One of the most essential questions concerning business processes is what makes up the information to be registered in the registry. At least a BPSS should be registered. However, it is questionable whether a defined series of activity steps for a business process each referencing a business document is the solution for the ambitious goals of ebXML. There is a high risk to repeat the failures of EDI by concentrating only on the runtime aspects which makes it hard to bind it back to the real business requirements. This would require a cumbersome reengineering process for integrating the definitions into software components leaving open space for different interpretations and, consequently, hindering interoperability.

Hence, the full business process definition, ranging from the business requirements to the runtime aspects as defined by following UMM, should be registered. This would require an XML-based format reflecting all the artifacts defined in the UMM meta model. It is important that such a format rather concentrates on the business aspects of an UMM model than on the UML specifics.

This leads to the next problem. Even if it is desirable that all ebXML-compliant specifications will follow the UMM methodology, it is rather unlikely that the whole world will follow a standardized process in developing B2B processes. This becomes even more evident when considering that there already exist B2B business process definitions that would not be defined again by following UMM. What is needed is something like a quality of service certificate (cf. ISO 9000 certificates) for B2B process definitions to be used in ebXML. Research has to identify those parameters that qualify for a certain certificate or better for a certain level of a certificate. The certified level should be stored together with the business process definition in the registry. It is obvious that the parameters used for certification purposes must be inline with the previously mentioned XML-based format for registering business process definitions. This would enable an automatic certification process for business process models in the registry.

Other important aspects concerning business processes are multiparty business collaborations. Currently, multiparty collaborations are always synthesized from two or more binary collaborations. This might work in most business cases, but will not work when a reply is coming from another party than the request was sent to (e.g. the business process scenario for a letter of credit). Research has to identify a concept for defining as well as monitoring multiparty transactions. Monitoring should allow each party to access the current state of the business collaboration that is rather straightforward in binary ones. To be accepted by businesses a concept for monitoring must not rely on a centrally controlled monitor.

Furthermore, there does not exist a concept for rolling back business transactions in ebXML. It is assumed that a possible roll-back is a business concept that is itself modelled explicitly as part of the choreography of a business process. Research should identify alternatives to this approach, e.g. adopt the work on transactional workflows to be incorporated into ebXML specifications. Again, such roll-back mechanisms are more complicated, but even more important in multiparty business collaborations.

Another important point is to find a mechanism avoiding the proliferation of business process specifications, especially for conflicting solutions for the same business semantics. This would need a set of commonly accepted core business processes, as already started by ebXML, and a clearly defined unambiguous concept for extending them to special needs of certain industries or businesses. However, more generalized definitions will also require the need for profiles on business processes as we discussed in the previous subsection. Research should also define a mechanism for conflict detection and/or conflict resolution, that might be based on ontology work.

Moreover, there still are open issues within UMM. One of the strengths of the RosettaNet approach adopted by UMM are patterns specified on the business activity level for different business requests and responses, that also lead to common patterns for message exchanges. As envisioned by ebTWG, research has now to extend the pattern approach to the level of business collaborations including multiple business activities.

3.5. Core components

Basic explanation - B2B business processes are supported by the interchange of business documents. Business process and information modeling identify the individual pieces of business information that are exchanged in each step. However, interoperability requires that the same business semantics, even used in different business processes, leads to the same information structures. Within ebXML this problem is addressed by the Core Component architecture that is based on a combination of reusable building blocks and the use of context [5].

A component is a 'building block' that contains pieces of business information that belong to a single concept. Core components are characterized by the fact that they appear in many different circumstances of business information and in many different areas of business. A core component is a general building block that is free of context. It is either an atomic information block or an aggregate of blocks semantically completing each other. A core component that is used in a business process has to be set into context. Context is the description of the environment use will occur within. Each context-specific use of a common component is catalogued under a new business information name. Atomic core components that naturally go together can be grouped into aggregate core components.

A domain component is specific to an individual industry area and is only used within that domain. It may be reused by another domain if it is found to be appropriate and adequate for its use, and it then becomes a core component.

Status - In May 2001 - by end of the 18-months of the first phase of ebXML – the output regarding core components was not considered to be mature enough to become an ebXML specification. Thus, the core component documents were released with the status of technical reports. This work is currently continued under the umbrella of UN/CEFACT in the electronic business transition working group (ebTWG). It is expected that a single core component document becomes a technical specification early in 2002. In order to define the actual content of a core component library it is necessary both to identify and document core components and to categorize different types of context, so-called context drivers. Since ebXML did not deliver a core component library by end of the first phase, it failed to define the backbone for data integration. However, initial content work has been started in parallel with the framework specification work mainly based on harmonizing business semantics in the area of materials management, banking, insurance, and transport. Furthermore, the existing draft of the core component library covers only basic information entities of these areas.

Scenario - As already mentioned, core components are used to build business documents that in turn support business processes. In the ebXML scenario there is no extra step related to core components only. Rather, core components are considered as part of the definition of a business process. Accordingly, looking at the ebXML scenario form a core component point of view is similar to that from business processes described in the previous subsection. Firstly, a company retrieves the core components included in requested business process definitions (step 1). It has to develop software components to access the core components of supported business processes (step 2). By registering its profile the company announces also its support of core components residing in referenced business processes (step 3). A future business partner might find the company according to its complementary rule in a business process using the same component structure in the supporting business documents (step 4). In a trading partner agreement the companies agree on the business processes and their component structure (step 5). Finally, they do business by exchanging business documents that instantiate core components (step 6).

Research issues - An essential problem area is the discovery and harmonization of core components. Currently, the technical reports describe a methodology that is solely based on human expertise. A domain-specific team defines the domain-specific data requirements. The experts in a special harmonization analysis team examine the discovered data requirements to ensure that they are semantically correct and to provide core components that are harmonized across industries. Since this approach depends on the knowledge and process know-how of the individual experts in the harmonization analysis team, research should identify solutions that enable organizational learning and capturing organizational know-how of the team in order to support the continuing efforts of the team. This also puts pressure on the way to document core components. Currently, syntax-neutral core components are defined by the following set of attributes: identification, reference term, definition, data type, and a list of synonyms. A definition of semantic relationships between core components is more or less missing. An alternative approach should consider work done in the area of ontologies and appropriate representation languages. It also could be of interest to develop domain-specific ontologies and to provide for each of them a referencing mechanism to a consolidated ontology.

Furthermore, the necessity of a central harmonization analysis team might lead to a cumbersome and time-consuming standardization process like in traditional EDI. This is in contrast to the idea of a flexible and open approach to define business processes not requiring the approval of a standard organization. If an organization defines business processes and defines a component structure according to its data requirements, it needs to check a resulting structure against core component definitions already registered. An automatic approach would minimize the 'standardization' time by avoiding a proliferation of different concepts for the same business semantics.

Another open question is to which extend the aggregation level of core components makes sense. Theoretically, each business document could become a core component consisting of multiple other core components. However this seems not to be practicable. An alternative approach would be to keep the aggregation level rather low and at the same time declare well-defined interfaces for each core component. This means, that for each core component it must be defined how relationships to other core components have to be established and, furthermore, interfaces to which other core components are useful or not. This should lead to a unit construction system for core components.

In addition to that further research on context drivers for core components used in a specific business process is needed. Open issues are the following: What is the complete list of context drivers or should this list be kept dynamic? How to define the relationship between core components and their context drivers, or in other words which context drivers are useful for which core components in which business processes? To which extent are context drivers populated to related core components in given business document definitions? How to include context drivers in trading partner profiles and are profile specific context drivers allowed or not?

According to the current ebXML specifications core components are syntax-neutral. This means that they are defined independently of any e-business vocabulary to be used for exchanging business documents. This is fully in accordance with the Open-edi approach on documenting the business aspects independently of the technology aspects of e-business transactions [13]. However, it is not clear today whether there will be an ebXML-specific XML-based e-business vocabulary. Some people prefer such a standardized representation language for business documents. Others claim that we are far too deep in the XML era and thus it is very unlikely that all the existing e-business vocabularies and their users will move towards an ebXML-specific one. Since we belong to the latter group, we feel that interoperability between various e-business vocabularies is a key issue for the future. This requires mapping rules for core components and whole business documents into each e-business vocabulary that is relevant. In this case interoperability is not reached by common element names, but by XML attributes referencing the unique identification (UUID) of core components as defined within the registry. This enables bridging between languages and even different XML-based business vocabularies.

However, research projects are needed to show the feasibility of such an approach. A more sophisticated concept has not only to overcome name clashes, but also structural clashes between different vocabularies. Additionally, exact rules for the handling of semantic incompatibilities between e-business vocabularies must be defined.

Another aspect concerns the usage of core components as drivers for the choreography of business processes. In the current ebXML specification the choreography of business processes is defined independently of the payload of business documents. However, some times the execution of an inter-organizational workflow might be dependent on the actual content of business messages. This requires the declaration of the 'workflow'-relevant core components together with an unambiguous rule set in the business process definition.

As stated earlier, ebXML's goal is to lead to inexpensive, commercial off-the-shelf (COTS) software solutions for small-and-medium enterprises. This goal can only be reached if software companies can easily adopt new business processes by the reuse of software components of already implemented ones. Thus, research projects should identify appropriate software component architectures to access core component structures that are put into context by specific context drivers and might be presented in various e-business vocabularies.

4. Summary

ebXML specification and development has reached quite a mature state with respect to the level of transport and routing. But open issues can be identified with any upper level, the registry and repository level, the trading partner profile level, the business process level, and the core component level. From a conceptual as well as technical viewpoint, the complexity of open issues seems to be highest at the level of business processes, followed by the issues identified at the core component level and the trading partner profile level. If ebXML is fully functional and implementations are available for trading partners world-wide there will be a need to establish a world-wide ebXML registry infrastructure – compareable to a world-wide public-private key infrastructure. In our opinion most of the open issues need interdisciplinary approaches in order to get feasibility solutions. It is of high importance that those mistakes that were made in the context of the EDIFACT standardization because of a very technical treatment of the problems will be avoided in the ebXML context. There is a very high demand and potential for computer science and applied research to foster appropriate solutions to open issues of ebXML as those identified in this paper.

5. References

[1] Ariba, *cXML User's Guide*, August 2001, http://xml.cxml.org/current/cXMLUsersGuide.pdf

[2] Commerce One, *XML Common Business Library*, http://www.xcbl.org

[3] ebXML, *ebXML Business Process Specification Schema*, Version 1.01, May 2001, http://www.ebxml.org/specs/ebBPSS.pdf

[4] ebXML, *ebXML Collaboration-Protocol Profile and Agreement Specification*, Version 1.0, May 2001, http://www.ebxml.org/specs/ebCPP.pdf

[5] ebXML, *ebXML Core Component Overview*, Version 1.05, May 2001, http://www.ebxml.org/specs/ccOVER.pdf

[6] ebXML, *ebXML Message Service Specification*, Version 1.0, May 2001, http://www.ebxml.org/specs/ebMS.pdf

[7] ebXML, *ebXML Registry Information Model*, Version 1.0, May 2001, http://www.ebxml.org/specs/ebRIM.pdf

[8] ebXML, *ebXML Registry Services Specification*, Version 1.0, May 2001, http://www.ebxml.org/specs/ebRS.pdf

[9] ebXML*, ebXML Technical Architecture Specification v1.04*, February 2001, http://www.ebxml.org/specs/ebTA.pdf

[10] Glushko, R.J., J.M. Tenenbaum, B. Meltzer, "An XML Framework for Agent-based E-commerce," *Communications of the ACM*, vol. 42, no. 3, 1999, pp106 – 114

[11] Huemer, C., "Defining Electronic Data Interchange Transactions with UML", *Proceedings of HICSS-34 (34th Hawaiian International Conference on System Sciences)*, Maui, January 2001

[12] Huemer, C., "DIR-XML^2 - Unambiguous Access to XML-based Business Documents in B2B E-Commerce", *Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, October 2001

[13] ISO Std. 14662, *Open-edi Reference Model*, Geneva, 1997

[14] Kotok, A., and D. R. Webber, *ebXML: The New Global Standard for Doing Business on the Internet*, New Riders Publishing, August 2001

[15] Microsoft, *BizTalk Framework 2.0: Document and Message Specification*, December 2000, http://www.microsoft.com/biztalk/techinfo/BizTalkFramework20.doc

[16] Nickull, D., et al., *Professional ebXML Foundations*, Wrox Press Inc, November 2001

[17] Open Applications Group, *OAGI White Paper*, 2000, http://www.openapplications.org/downloads/whitepapers/whitepaperdocs/whitepaper.htm

[18] RosettaNet, *Guide to RosettaNet Specifications*, 1999, http://www.rosettanet.org

[19] Dan, A., et al., *Business to Business Integration with TpaML and a B2B Protocol Framework (BPF)*, IBM Research Report, RC 21863 (98377), October 2000

[20] W3C, *SOAP Version 1.2 Part 1: Messaging Framework*, October 2001,

http://www.w3.org/TR/2001/WD-soap12-part1-20011002/

[21] Webber, D.R., and Dutton, "Understanding ebXML, UDDI, XML/EDI", *XML.org Newsletter*,

http://www.xml.org/feature_articles/2000_1107_miller.shtml