B2B Integration -
Aligning ebXML and Ontology Approaches

Birgit Hofreiter & Christian Huemer
Institute for Computer Science and Business Informatics
University of Vienna, Liebiggasse 4, 1010 Vienna, Austria
{birgit.hofreiter,christian.huemer}@univie.ac.at

Abstract. In B2B e-commerce, XML provides means to exchange data between applications. It does not guarantee interoperability. On the syntactic level, this requires an agreement on an e-business vocabulary. Even more important, on the semantic level, business partners must share a common view unambiguously constraining the generic document types. In this paper, we present a framework that brings together work in the area of ontologies and work in the area of XML-based data interchange, namely ebXML. The framework uses an ontology based on ebXML corecomponents expressed in RDF to allow for bridging between different e-business vocabularies. Since a bridging mechanism is required, but not specified within ebXML, our approach complements ebXML. The integration of the ontology-based approach into ebXML is realized in four major steps. In this paper we exactly identify the requirements and the architecture of each step. This provides exact guidelines for future research towards implementing these steps.

1 Introduction

XML is said to overcome the most significant obstacles of traditional electronic data interchange (EDI) standards. After the first hype, people realized that XML provides means to exchange data between applications, but does not guarantee interoperability. XML only provides a syntax that could be used for data transfer in B2B, which is only one level of interoperability that must be met in B2B. Fig. 1 gives an overview of all B2B levels that must be agreed upon, or that a mapping between different protocols must be realized for. On the lowest level, interoperability on the level of the transport protocol (e.g. HTTP, SMTP, X.400) must be reached. Using a message-oriented middleware approach in a B2B setting requires reliable messaging and additional messaging envelope mechanisms, e.g. SOAP. It should be noted that even if both business partners use SOAP, interoperability is not guaranteed, since they might use incompatible SOAP variants, e.g. ebXML SOAP vs. BizTalk SOAP.

The third level has to ensure interoperability on the syntax used to encode business documents, like XML or UN/EDIFACT. In this paper, we only consider XML-based middleware and ignore all other syntaxes as well as the document transport on the lower levels. XML provides syntax, not semantics, since tags have no predefined meaning [2]. The meaning of XML languages is defined by the document designer. This resulted in a proliferation of XML-based e-business vocabularies within the first few years of XML in existence [12]. Although we expect vocabularies disappearing and merging, a certain number of well known "standard" vocabularies will co-exist. This means that on the fourth level business partners must either agree on a certain e-business vocabulary, or a mapping between their preferred vocabularies must be realized. We expect that companies will prefer a single interface that automatically maps to the different e-business vocabularies over implementing an interface to the in-house information system for each e-business vocabulary. Therefore, this paper emphasizes interoperability between different e-business vocabularies.

However, document types of e-business vocabularies are much too ambiguous including a lot of optionality and covering much more semantics than an involved application is able to process. In other words, a valid XML business document does not guarantee that the business partner is able to process the document. It has to follow the shared view of the business partners on the business content. This shared view on a document's semantics is an agreement to be met on the fifth level of interoperability. Implementing and maintaining these agreements - called message implementation guidlines (MIGs) in EDI - makes EDI expensive [10].

Level 6	Business Process Semantics	order management
Level 5	Document Semantics	purchase order
Level 4	E-Business Vocabulary (incl. generic document types)	xCBL order
Level 3	Transfersyntax	XML
Level 2	Messaging Envelope	SOAP
Level 1	Transport Protocoll	HTTP

Fig. 1. Levels of Interoperability in B2B

Instead of standardizing voluminous and vague business document types, ebXML is based on unambiguous business collaborations that represent an agreement on the sixth level of interoperability. For this purpose, a clear choreography of business activities including an unambiguous and context-specific definition for the business content exchanged in each activity is defined. In other words, this paper concentrates on a framework for context-specific views into the core components-based ontology for each single business activity. Furthermore, the framework aims to automatically transform a corresponding view into various e-business vocabularies, provided an existing core component binding for the respective vocabulary.

The remainder of the paper is structured as follows: In Section 2 we give a brief introduction into the concepts we have adopted from related work, namely e-business vocabularies, ontologies, Open-edi, UMM, and ebXML. The main contribution of our work is elaborated in Section 3, where we present the necessary steps to integrage the basic ideas on ontologies into the ebXML framework. These steps are the following: definition of a document ontology, languange binding for e-business vocabularies, definition of contex-specific views into documents to support a specific business activity, and representing these views in different e-business vocabularies. Each of these steps is presented in its own subsection. The paper concludes with a short summary.

2 Related Work

The work presented in this paper does not by itself create any new B2B technology, rather does it interlink and coordinate already existing technologies to ensure B2B interoperability. This section introduces the key concepts the paper refers to.

E-business vocabularies: XML became the preferred way to exchange business data over the Internet. A lot of organizations developed their own vocabulary. Lacking guiding standards for interoperability, those solutions use different data structures and tagging to encode the same business concept. Although some of the vocabularies have become the first choice within a vertical, there are still some competing efforts. Popular e-business vocabularies include languages of market place providers, like Commerce One's xCBL and Ariba's cXML, the Open Application Group's OAGI to interconnect ERP systems, and domain-specific solutions like RosettaNet in the IT sector. Getting them all to interoperate is still a challenge for the B2B community. An overview of E-business vocabularies is provided in [13].

Ontologies: An ontology is defined by Gruber as a "formal, explicit specification of a shared conceptualization" [7]. According to this definition, even a DTD or an XML schema for a business document type can be regarded as a very primitive ontology.

However, both DTDs and XML schemas are basically just a set of terms and do not define relationships between different terms (cf. [6]). More advanced applications require more expressive ontology languages, like RDF/RDF schema [3], DAML+OIL [4], SHOE [8], or OML [15]. A lot of ontology approaches are directed towards the semantic web [1]. It is the goal to define rules and meanings of web data that precisely enough that machines can correctly interpret them. Similar to the problem of web data is that of e-business vocabularies. For interoperability of different vocabularies a shared set of terms and their interrelationships with a common understanding is needed. An approach to develop an ontology for business documents based on reverse engineering existing e-business vocabularies is described in [14].

Open-edi: The idea of separating the business semantics and its representation in a certain e-business vocabulary was alreay a key concept of the Open-edi initiative started in 1988. Open-edi distinguishes between a business operational view (BOV) and a functional service view (FSV). The BOV is defined as ´a perspective of business transactions limited to those aspects regarding the making of business decisions and commitments among organizations, which are needed for the description of a business transaction', while the FSV focuses on implementation-specific technological aspects of Open-edi. The Open-edi reference model [11], which became ISO standard 14662, guides B2B standard works to ensure the coherence and integration of related standardized modeling and descriptive techniques, services, service interfaces, and protocols.

UMM: UN/CEFACT's Modelling Methodology (UMM) is a modeling technique to describe the BOV aspects of Open-edi. 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 that utilizes UML [17]. A business process describes in detail how trading partners take on shared roles, relationships and responsibilities to facilitate interaction with each other. An interaction between roles follows a choreographed set of business transactions, whereby each transaction is expressed as an exchange of electronic business documents. Business partners will be able to communicate with each other if they support the same unambiguously defined choreography of transactions using unambiguously defined business document types.

ebXML: In order to provide an FSV layer that takes full advantage of the Open-edi concept and UMM, UN/CEFACT joined with OASIS in the ebXML initiative [5]. ebXML offers a modular suite of specifications. These specifications provide a standard method to exchange business messages, conduct trading relationships, communicate data in common terms, and define and register business processes [9]. In the context of this paper the ebXML specifications for business processes and core components are of particular relevance. The ebXML business process specification schema (BPSS) adopts a subset of UMM needed to configure ebXML-compliant software. An ebXML-compliant software will then be able to control a business process from the corresponding business partners view by monitoring state changes resulting from document exchanges. In ebXML, a document type does not correspond to the union set of all possibly required data structures needed for anyone's version of a given transaction type. A document is defined by an unambiguous data structure exactly meeting the business requirements to reach the business goals of a single activity in a business process. However, ebXML does not use its own e-business vocabulary to describe business documents. Instead, ebXML document types are assembled from so-called core components which are syntax-neutral descriptions of semantically meaningful business concepts. Currently, ebXML does not specify any methododology to represent syntax-neutral core components in targeted e-business vocabularies. Thus, the framework presented in this paper will complement the ebXML approach.

3 ebXML Core Component-based Ontology Framework

In this section we present our framework to extend ebXML by ontology concepts. In order to develop a document ontology, two main approaches are introduced by Ontoprise's Semantic B2B Broker [16]: a *top-down* and a *bottom-up* approach. In a top-down approach business experts will first define a document ontology that describes their shared understanding of a business document type. The conceptual model of this document ontology builds the foundation to develop a new e-business vocabulary (represented as DTD or XML schema). Vice versa, a bottom-up approach takes DTDs or XML schemas from existing e-business vocabularies to analyze their semantic content. The result is an harmonized ontology of all considered e-business vocabularies have to be harmonized in order to define a unified document ontology.

The ebXML initiative is currently the strongest supported initiative by industry with respect to development of vocabulary-independent components, so-called core components. It is our goal to take advantage of a future pool of core components. Hence, our ontology layer is not defined by reverse engineering of existing e-business vocabularies like in the bottom-up approach. Instead, our ontology layer is based on ebXML core components. However, we do not use a pure top-down approach, because we will not develop a new e-business vocabulary. We have to mediate the ebXML-based ontology layer with existing e-business vocabularies. Since we are coming from top as well as from bottom, we call our approach "meet in the middle".

The presented framework is based on 4 major steps depicted in Fig. 2. The building of a document ontology starting from ebXML core components constituting the first one. The second step covers the definition of language bindings for various e-business vocabularies. The third step requires the definition of a view into the ontology that exactly meets the requirements of the document exchange supporting an ebXML business activity. Finally, the fourth step, which can be done automatically, takes on the language binding and the view specification and derives an implementation guideline in a certain e-business vocabulary. Each of these steps is introduced in the following subsections.

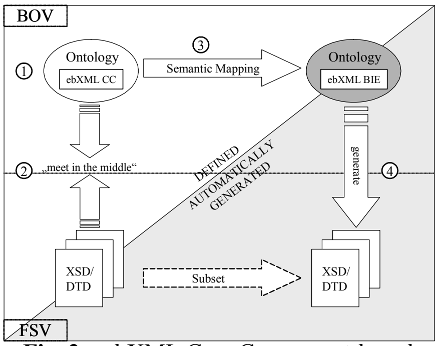

Fig. 2. ebXML Core Component-based Ontology Framework

3.1 Definition of a Core Components-based Document Ontology

In the first step we develop an ontology that follows the latest draft of the ebXML core components specification [18]. For this purpose we have developed an RDF schema (RDFS) [3] for the core components meta model that is depicted as a graph in Fig. 3. Note that all boxes with solid lines are of the RDFS type Class and those with broken lines are of type Property.

A *core component* is defined as a semantic building block that is used as a basis to construct all electronic business messages. There exist 3 different types of core components. A *basic core component* represents a singular business concept with a unique business-semantic definition. Each basic core component is of a certain core component type. A *core component type* (e.g. amount type) consists of a content component that carries the actual content (e.g. 12) plus one or more supplementary components giving an essential extra definition to the content component (e.g. Euros). Note, that content and supplementary components are nothing else than core components. Core component types do not have business meaning. An *aggregate core component* is a bag of core components that convey a distinct business meaning.

In the RDF Schema in Fig. 3 the three different types *AggregateCoreComponent*, *Basic Core Component*, and *CoreComponentType* are represented as subclasses of the class *CoreComponent*. The property elementType is used to assign a core component type to a basic core component. The composition of a core component type is defined by the properties *contentComponent* and *supplementaryComponent*, referencing basic core components. The property coreComponentChild is used to reference the components within an aggregate.

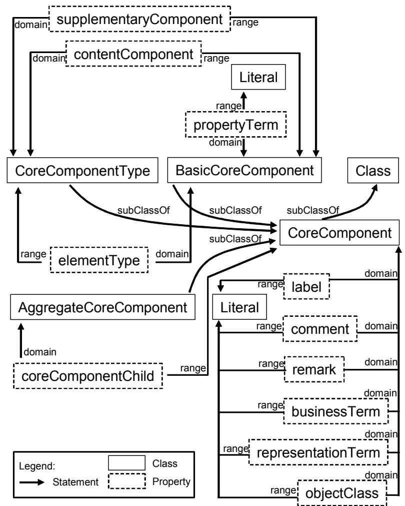

Fig. 3. RDFS meta model for CC

Each ebXML core component contains the following dictionary information: A dictionary entry name is the unique official name of the core component. It corresponds to the RDFS property *label*. The definition of the unique semantic business meaning of the core component is given in the RDFS property *comment*. The property *remark* is used to further clarify the definition, to provide examples and/ or to reference a recognized standard. If there exist further synonym terms under which the core component is commonly known and used in the business, the property *businessTerm* is used to define them. We assign the properties *objectClass*, *representationTerm*, and *propertyTerm* to core components as defined in the ebXML specification.

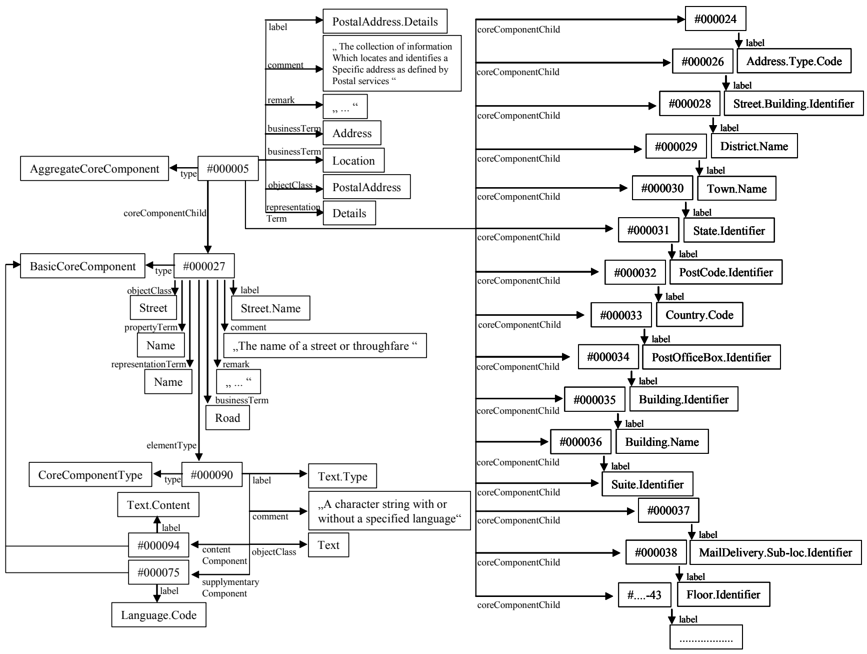

Fig. 4. RDF-Model for the Aggregate Core Component "Postal Address Details"

The RDF library of core components must be populated with all core components currently being developed by UN/CEFACT. Each core component will be expressed as a RDF model that follows the RDF (meta) schema of the core components' meta model. Fig. 4 demonstrates the population of the RDF library by the means of the aggregate core component *PostalAddress.Details*. Due to space limitations, the resource representing this aggregate core component is marked *#000005* in Fig. 4. This represents only the fragment identifier of a URI uniquely identifying core components. The unique identifications (UID) assigned to core components by ebXML are used as fragment identifier. The label of the resource is equal to the data dictionary entry name *PostalAddress.Details*. The comment states the definition of the PostalAddress.Details. The business terms *Address* and *Location* which are commonly used to refer to PostalAddress are assigned to the aggregate core component. Each of the components aggregated within PostalAddress.Details is assigned as coreComponentChild. Owing to space limitations we have detailed only the basic core component *#00027 Street.Name*. In addition to label, comment, remark and business term (*Road*), the basic core component gets assigned an object class (*Street*), property term (*Name*), and a presentation term (*Name*). More important is the fact that a basic core component is of exactly one core component type. *Street.Name* is of type *#000090 Text.Type*. This core component type includes the content component Text.Content (which includes at one instance the name of the street) and the supplementary component Language.Code.

So far we have been considering the meta model of core components to capture the content of an ebXML core component library by the means of RDFS. A document ontology will use identified core components and their interrelationships as basic semantic building blocks for document types. The document ontology is also expressed in RDFS. Consequently, document instances are valid RDF models of the document's RDFS. Fig. 5 depicts an example of a document instance that is a valid fragment of a postal address representing the German (ISO Code 936: *DE*) name of the street *Liebiggasse*. For a better understanding, we have marked the anonymous resources in the grey boxes with a meaningful name. Each of these resources is an instance of a core component. The corresponding core component is referenced as the resource's type. This allows to identify the semantic context of the resource. Consequently, the semantic relationship between core component instances is comprehended even using an anonymous referencing mechanism via the property "references" between all instances of any types of core components.

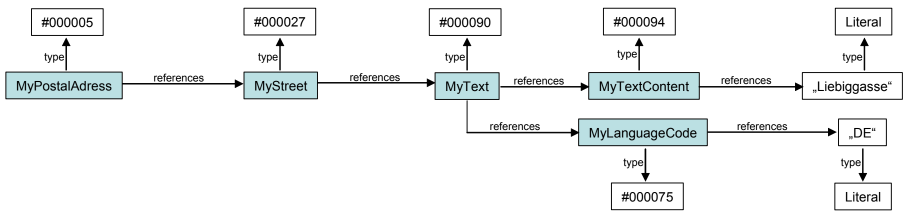

Fig. 5. RDF Instantiation using Aggregate Core Component "Postal Address Details"

3.2 Language Binding for e-Business Vocabularies

Having defined a document ontology, the next step is to provide language bindings between the document ontology and the corresponding document type of an e-business vocabulary. Since our "meet-in-the-middle" approach and a bottom-up approach only differ in the way the ontology is built, the problem of defining a language binding remains the same. Thus, our framework considers the language binding defined in the bottom-up approach by Omelayenko and Fensel [14].

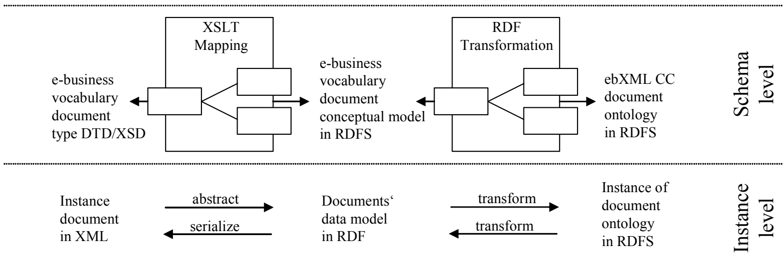

Fig. 6. Language Binding between Ontology and e-business vocabulary

The basic concept of this language binding is depicted in [6] It is based on the definition of a common conceptual model for document types of different e-business vocabularies. The conceptual models are described in RDFS. Thus, a mapping between the vocabulary's DTD or XML schema on the one side and the conceptual data model on the other side is required. This mapping on the schema level is defined by the means of XSLT. On the instance level, an incoming document is abstracted from its XML serialization and translated into its RDF data model. Vice versa, in order to create an outgoing document, the RDF data model of the target vocabulary is serialzed according to the target XML format.

Furthermore, the conceptual data model of an e-business vocabulary's document type (expressed in RDFS) must be mapped to the document ontologies data model of the same document type (also expressed in RDFS). This means that the equivalence mapping between the terminologies requires a transformation from one RDFS model to the other. The mapping must be described by the means of a RDFS mapping language.The mappings must be automatically translated into an RDF transformation language. This transformation language is applied to translate the conceptual RDF model of an incoming document into an instance of the document ontology and, vice versa, to translate an instance of the document ontology into the conceptual RDF model of an outgoing document. It follows that all mappings between different vocabularies will also be managed via mapping each standard to the core components-based document ontology. Unfortunately, there neither does exist a standard for the RDFS mapping language nor for the RDF transformation language. The specification of these languages to support the overall framework is an essential future work item.

3.3 Context-specific View Definitions

If business partners were able to process all the semantics that are usually included in a business document type of a standard vocabulary, step 2 would already be the final step. Consider the fact that e.g., flattening an xCBL purchase order will result in about 16,000 data element types covering most probably a similar number of semantic concepts. Then it becomes evident that a company usually supports only a subset of the concepts or in other words a specific view of the general standard document.

Hence, exchanging valid XML documents is not enough to ensure interoperability, because these documents might carry data representing business concepts not supported by the business partner. It is an indispensable requirement that business partners share a common view on the semantic concepts included in a document type to collaborate in a B2B transaction. In the context of ontologies, a view setting approach as defined by Ontoprise's Semantic B2B Broker is required [16]. The view setting is used if different users have to be provided with different views on the same information source (= document type). In this case, the ontology of the information source may be semanically restricted to the conceptual view shared by all business partners.

According to UMM, the information requirements of each business activity resulting in a document exchange are analyzed to support the overall business process. The information requirements result in an unambiguous set of concepts that must be supported in the exchanged document type. Unambiguous means that there is no open space for partner-specific views agreed on at run time. The semantic concepts to be shared by all business partners supporting a business process are fixed at design time.

ebXML uses UMM artifacts to define a corresponding conceptual data model based on core components. When a core component is used in the context of a business activity, its refinement becomes a so-called *business information entity*. A business information entity is a piece of data or a group thereof with a unique business semantic definition in a given context. Accordingly, a language to define refinements on a semantic level is necessary. For this purpose, ebXML specifies a constraint language. The scope of the constraint language is to refine an assembly as appropriate. As depicted in Fig. 7, the document ontology representing the general document structure based on core components (expressed in RDFS) must be refined into an activity-specific document structure based on business information entities (also expressed in RDFS). Thus, a language allowing for a refinement of an RDFS must be devloped. This language should follow the semantics captured by ebXML's constraint language.

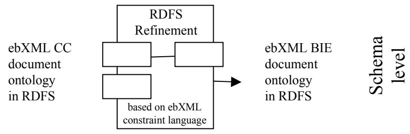

Fig. 7. Document ontology refinement

3.4 Representing Views in e-Business Vocabularies

The last step of the framework covers the definition of unambiguous document types in the XML syntax of e-business vocabularies. The document ontology based on business information entities to support a specific activity is more restrictive than the general document ontology. Consequently, the DTD or XML schema for the e-business vocabulary supporting a well-defined activity must be more restrictive than the general DTD or XML schema of the corresponding document type. Taking the example of the previous subsection, a certain step in a well-defined business activity will use only a well-defined subset of the 16,000 data elements in a general xCBL purchase order.

This time the mapping from the document ontology into the e-business vocabulary's DTD or XML schema should be done automatically. This will be enabled by the information about the mapping of the gerneral document type (step 2) as well as the information about the constraint language (step 3). We expect two major types of refinements in regard to their consequences on the automatic mapping. Firstly, some semantic concepts are not used at all. In this case applying the same RDF transformation language as in step 2 will result in the desired output. This is due to the fact that missing input will simply be ignored. Secondly, a component is used differently in different situations according to more complex constraints. This can be expressed neither in the document's DTD nor XML schema. Thus, it does not have any consequences on the output of an appropriate DTD or XML schema. However, these more complex constraints should be expressed as an declarative (XML-based) language which accompanies the document type [10]. In order to check the validity of a document for a given business activity, not only conformance to the document type, but also conformance to the rules of an instance of the declarative constraint language is required.

4 Summary

In this paper we presented an ebXML core component-based ontology framework to be used in B2B e-commerce. It is based on the idea of Open-edi to separate a business operational view represented by the document ontologies and a functional service view represented by the XML schemas or DTDs of business vocabularies. It specifyies four major steps to describe semantically equivalent document types in different e-business vocabularies for the same activity in a well defined business process.

Each of these steps is described in this paper on a conceptual level and directs our future research towards the implementations of these steps. In the first step the semantics of ebXML core components are defined in a RDFS-based ontology. The second step defines a mapping between the ontology and RDF representations of e-business vocabularies as well as another one between the latter RDF representations and the DTDs or XML schemas of the e-business vocabularies. Beyond traditional ontology approaches, we take on the ebXML idea to further restrict a general document ontology to the specific needs of a certain business activity. This refinement will lead to a certain view of the general document ontology and must be specified by means of a constraint language. Steps 1 to 3 have to be done manually, whereas the last step will be derived automatically. Using the information of the previous steps a more restrictive and appropriated XML schema or DTD for a specific activity will be created. The XSDs and DTDs are a subset of their corresponding general ones. Further restictions expressed in a declarative language will accompany the document types. This framework focuses both, a mapping between different e-business vocabularies and at the same time guaranteeing their semantical equivalence in support of a specific activity.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic Web. Scientific American (2001)
2. Bosak, J.: Media-Independent Publishing: Four Myths about XML. IEEE Computer, Vol. 31, No. 10 (1998)
3. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft. (2002), http://www.w3.org/TR/2002/WD-rdf-schema-20020430
4. DARPA: The DARPA Agent Markup Language Homepage. http://www.daml.org
5. ebXML: Homepage of the ebXML Initiative. http://www.ebXML.org
6. Erdmann, M., Studer, R.: Ontologies as Conceptual Models for XML Documents. 12th Workshop on Knowledge Acquisition, Modeling and Management, Canada, (1999)
7. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition Vol. 6, No. 2 (1993)
8. Heflin, J., Hendler J.: A Portrait of the Semantic Web in Action. IEEE Intelligent Systems, Vol. 16, No. 2 (2001)
9. Hofreiter, B., Huemer, C., Klas, W.: ebXML: Status, Research Issues and Obstacles, Proc. of 12th Int. Workshop on Research Issues on Data Engineering (RIDE02), San Jose (2002)
10. Huemer, C.: <<DIR>>-XML^2 -Unambiguous Access to XML-based Business Documents in B2B E-Commerce. Proc. of 3rd ACM Conference on Electronic Commerce, Tampa (2001)
11. ISO: Open-edi Reference Model. ISO/IEC JTC 1/SC30 ISO Standard 14662 (1995)
12. Kotok, A.: Extensible and More - A Survey of XML Business Data Exchange Vocabularies. O'Reilly xml.com, http://www.xml.com/pub/2000/02/23/ebiz/index.html
13. Li, H.: XML and Industrial Standards for Electronic Commerce. Knowledge and Information Systems, Vol. 2, No. 4 (2000)
14. Omelayenko, B., Fensel, D.: Scalable Document Integration For B2B Electronic Commerce. submitted to Electronic Commerce Research Journal
15. Ontology Consortium: Ontology Markup Language Project. http://www.ontologos.org/OML/OML.html
16. Ontoprise, B^3 Semantic B2B Broker, http://www.ontoprise.de, 2000
17. UN/CEFACT TMWG, UN/CEFACT Modelling Methodology, http://www.gefeg.com/tmwg
18. UN/CEFACT ebTWG, Core Component Technical Specification, Version 1.8, (2002) http://www.ebtwg.org/projects/documentation/core/CoreComponentsTS1.80.pdf