September 2001

B^3 - The Semantic B2B Broker

Jürgen Angele
ontoprise GmbH, angele@ontoprise.de

Michael Erdmann
Institut AIFB, University of Karlsruhe, angele@ontoprise.de

In: Buhl, Hans Ulrich, u.a. (Hg.) 2001. *Information Age Economy*; 5. Internationale Tagung Wirtschaftsinformatik 2001. Heidelberg: Physica-Verlag

ISBN: 3-7908-1427-X

B^3 – The Semantic B2B Broker

Jürgen Angele

ontoprise GmbH

Michael Erdmann

Institut AIFB, University of Karlsruhe

Abstract: This paper will discuss the necessity for standardizing semantics beyond simply standardizing syntactic and structural (XML) vocabularies in the area of E- commerce. We will present an ontology-based approach to handle semantics, which enables application developers to raise the level of abstraction from error-prone syntactic manipulations to a conceptual level. This approach has been implemented in the B^3 set of tools that is able to integrate conceptual knowledge modeled in ontologies and factual knowledge originating from business applications and transactions

Keywords: semantic web, ontology, e-commerce, XML, RDF

1 Introduction

The development of the World Wide Web changes the business relationships between enterprises. Already the "First Generation Web" that uses HTML or XML to make business information exchangeable in a platform independent way had a significant impact on the way enterprises exchange information among each other. While HTML comes with a built-in vocabulary of tags that is not customizable, its use for representing information in a machine-processible manner is rather limited. Furthermore, these tags are aimed at presenting information on a screen and not for further processing. This situation drastically changed with the appearance of XML [Bray et al. 98]. This language allows the definition of domain and application specific document types. XML documents that adhere to certain Document Type Definitions (DTDs) are said to be valid. Dozens of DTDs have been developed that claim to be *standards* for tasks in the area of E-commerce. The so-called standards define the *structure* of valid XML-documents formally (via DTDs or other schema languages). But the *meaning* of documents to be exchanged is typically specified in natural language. Thus processing XML-documents is limited to syntactical tasks, only, or must be focused upon a single DTD.

Thus, since even XML only describes the *structure* of documents in a machine-processible way, the "First Generation Web" does not provide the support that is needed for truly interoperable software, esp. (but not only) in the area of B2B E-commerce. Rather, the notion of the "Semantic Web" as coined by Tim Berners-Lee [Berners-Lee 99] paves the way to make the *semantics* of documents or more generally speaking the semantics of business information sources machine-processible. Thus, new types of B2B-solutions will become feasible that will exploit the semantic description of information sources. Currently, B2B-applications that are based on XML are put into practice. That XML is *not* the solution for interoperability in B2B-applications will be recognized as soon as a large number of enterprises will use these XML-based solutions: the syntactical and therefore unwieldy handling of interoperability aspects will end up in the demand to manage heterogeneous information sources on a conceptual and thus a semantic level.

The Semantic Web builds on two basic facilities to enable wide interoperability without human intervention: (i) a single representation language to represent data on the web and (ii) a mechanism to provide machine-processible schema information to add meaning to the bits of data. The language in (i) is not XML, due to the above-mentioned reasons, but needs a formal semantic underpinning. Currently RDF (the Resource Description Framework [Lassila, Swick 99]) is proposed by the W3C to play this fundamental role. RDF is accompanied with a schema language called RDF-Schema [Brickley, Guha 99], which provides basic semantic modeling primitives. Other, more powerful schema languages are currently under development like OIL [Fensel et al. 00] or DAML+OIL (http://www.daml.org/2000/12/daml+oil-index.html).

Even without an RDF-based representation language the advantages of explicitly stated conceptual schemas can be utilized in B2B applications. We will present the notion of ontologies as a means to represent powerful conceptual schemas. By defining mappings between semantic terms from an ontology and the structure of XML-documents we can make a huge step towards what is possible in a world-wide Semantic Web.

This paper presents a set of tools, called B^3 Semantic **B**2**B** **B**roker, that will provide advanced support for developing and managing new kinds of semantic B2B-solutions, based on XML and ontologies, that are already prepared to be extended by the languages and protocols that will be developed for the Semantic Web.

2 Example Scenario: Product Catalogues

Let us start with an example scenario taken from the product catalogue domain to illustrate the interoperability problems that occur within B2B-applications. Product catalogues must be handled whenever a seller offers products to buyers or vice versa a buyer requires such a catalogue from its sellers.

At the moment, product catalogues are most often handled in a way that consumes too much time and human resources: The seller's product catalogue is transformed into an electronic data format (data are often manually edited); programmers then write transformation programs to generate the different formats and views required by the buyers; the resulting data are then sent to the different buyers. Clearly this process has two main disadvantages: it requires too much human resources and more important it is too slow. The last point has the severe consequence that modifications of the product catalogue (e.g. a modification of a price) is reflected too late at the buyer's side which has severe disadvantages for both the sellers and the buyers. Subsequently, we will use the product catalogue of a computer dealer for further illustration. Among others, this catalogue contains data about printers. A printer is described by its name, its price, its features, its producer and its type, i.e. whether it is a laser printer, an ink printer or a matrix printer. Our dealer is located in the US, so the product prices are given in US $. Let us assume that we have a buyer in Europe: that buyer expects product prices to be given in Euro; furthermore, he expects the catalogue to be structured in another way, i.e., that a printer is a laser printer is not described as an attribute *value* of a printer, but as a special subclass of printers. So besides the transformation of one basic data format into another, at least three kinds of transformations are required:

- Naming transformations: a concept or attribute has to be renamed.
- Value transformations: the price in US $ has to be transformed into a price in Euro.
- Structural transformations: the value of an attribute on the seller's side defines the membership to a subclass on the buyer's side, i.e. values have to be transformed into schema information and vice versa.

Especially structural transformations may become very complex.

3 Current Approaches

3.1 EDIFACT

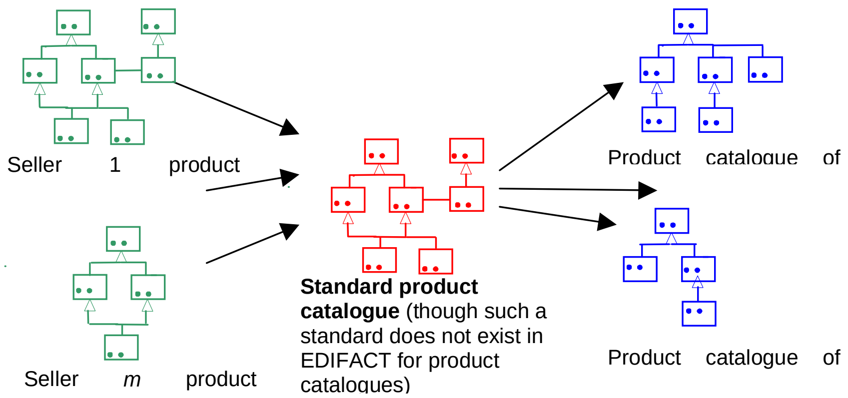

Figure 1: Product Catalogue Translation Steps based on a Standard

The huge effort of transforming different electronic data formats into each other has been tackled (but not entirely solved) for a long time, EDIFACT being one of the most notable examples. In essence, EDIFACT provides a standardized electronic format to exchange data between sellers and buyers. This rather syntactical aspect is supplemented with a more semantics-oriented aspect: the specification of standards for various application areas. These standards define the terms and structures that are used to describe products and thus provide a single reference point for the involved sellers and buyers (cf. Figure 1). As a consequence, the overhead for exchanging data between *m* sellers and *n* buyers is reduced to *m*+*n* transformations, instead of *m***n* transformations.

Thus EDI integrates two different kinds of description levels: the basic data format and the standards, which cover the semantics of different application areas. Nevertheless, EDIFACT is not the final solution for data exchange due to the following reasons:

- EDIFACT standards are costly to specify and maintain, only experts are able to understand these descriptions,
- EDIFACT formats are isolated, i.e. they do not integrate well with other document/data exchange standards. They are not robust with respect to modifications.

3.2 XML

The emerging standard for data exchange is XML [Bray et al. 98]. Basically, XML-specifications provide a semi-structured description of products, orders and the like by using appropriate tags for XML-elements. This approach is illustrated in Figure 2 where two different variants of XML-descriptions of a laser printer are given (compare Section 2).

In order to define the structure of product catalogues or more generally speaking the structure of the data to be exchanged between different stakeholders, Data Type Definitions (DTDs) or XML-schema specifications (cf. [Thompson et al. 00], [Biron, Malhorta 00]) can be used as the contractual basis for communication. Thus, these data descriptions may be checked for structural validity. As such, XML is *the* new standard format for data exchange in a lot of new B2B-applications.

When comparing an XML-based approach with an EDIFACT-based approach, one can easily identify a variety of advantages for XML:

- XML is readable (and understandable) for non-experts,
- XML is well integrated with other document/data exchange standards,
- DTDs and XML-schema allow to validate the syntactic structure of XML documents,
- DTDs and XML-schema enable the generation of tools like form editors etc.,
- XML is quite robust with respect to some types of modifications,
- XML is a format designed for the web and thus will be spread very fast and wide.

```
<article>
  <articleid>a-5634</articleid>
  <category>printer</category>
  <name>hp81</name>
  <price>500 $</price>
  <producer>hp</producer>
  <features>2000 resolution</features>
….<type>laser</type>
</article>
```

```
<article>
  <articleid>a-5634</articleid>
  <category>laserprinter</category>
  <name>hp81</name>
  <price>625 E</price>
  <producer>hp</producer>
  <features>2000 resolution</features>

</article>
```

Figure 2: XML Descriptions of a printer

To handle the heterogeneity of XML sources within B2B-applications, one can exploit XSLT [Clark 99], which provides means for specifying *syntactical* transformations of XML documents. To come up with these transformations one has to start with a kind of reengineering of the semantics of the application domain that is hidden in the XML-tags (the programmer must know the semantics before he can write XSLT rules). Since these XSLT rules are based on the syntactical structure of XML documents, capturing the semantics of a particular domain is hardly possible at all.

Though the XML format provides the above-mentioned features, XML-based B2B applications still have several severe disadvantages:

- There is no commitment to a domain specific vocabulary. Thus sellers and buyers have to agree implicitly on the meaning of tags.
- The validation of XML documents using DTDs is only a syntactic validation, i.e. the labeled tree structure is checked. No semantic constraints are enforced, like e.g. the following constraint: „if the catalogue contains a bundled computer system, this system integrates a mouse of a certain type.“
- There are no complex modeling primitives available for capturing the semantics of aspects like multi-valued attributes, is-a hierarchies or part-of hierarchies etc. The lack of an inheritance concept has the consequence that information has to be described redundantly at different locations within an XML document, which makes the maintenance more difficult and error prone.
- Transformations based on DTDs and languages like XSLT are purely syntactical transformations. Furthermore, there do not exist handy means for integrating and transforming information that comes from different XML sources.
- Query languages like XQL [Robie et al. 98] and XML-QL [Deutsch et al. 99] allow querying the XML-document structure , but not its semantic content. For instance such languages do not naturally allow posing queries like: “Give me all printers for which cartridges are available that last for more than 5000 pages.” This has the consequence that transformations based on such semantic contents need a lot of programming in languages like XSLT.

Let us illustrate that situation with an example. Assume a standard XML-catalogue format, such as BMEcat [Hümpel et al. 99]; here the XML-structure of the catalogue only offers very general means to provide information about possible relationships between the different articles contained in the catalogue, e.g. which print cartridges fit into a special class of printers. This information is stored in FEATURE elements with FNAME and FVALUE subelements that may contain any user-defined features. The above query could look like this in XQL (assuming a taxonomy of cartridges and printers with appropriate relations between them):

```
//ARTICLE[ARTCLE_DETAILS/SEGMENT=”Printer” $and$
   $x:=SUPPLIER_AID $and$
```

```
{//ARTICLE[ARTCLE_DETAILS/SEGMENT=”Cartridge” $and$
    ARTICLE_FEATURES/FEATURE[FNAME=”forUseWith”
            $and$ FVALUE=$x] $and$
    ARTICLE_FEATURES/FEATURE[FNAME=”capacity”
             $and$ FVALUE>5000
             $and$ FUNIT=”pages”]
] } ]
```

One can easily see that these kinds of queries are overloaded with a lot of syntactical aspects that hide the semantics of the query.

These semantic aspects require the establishment of a technological infrastructure that provides methods and tools to handle the semantic heterogeneity aspects in an appropriate and efficient way. Such an infrastructure is offered by B^3, the Semantic B2B Broker of *ontoprise* GmbH: we will subsequently describe the logical architecture (Section 4) and the specific tool-set of the B2B Broker (Section 5).

4 Ontology-based Approaches

Based on the analysis of the strengths and weaknesses of XML-based applications, *ontoprise* develops a new level of solutions for B2B-applications: the ontology-based Semantic B2B Broker. In essence, the ontology-based approach supersedes the syntactic XML-transformations with semantic mappings between ontologies, i.e., conceptual models of the respective domains (cf. Figure 3).

An *ontology* is defined as a "formal, explicit specification of a shared conceptualization" [Gruber 93]. In a business context an ontology talks about business objects, their properties, and relationships between these objects. Thus, ontologies provide a shared vocabulary of business terms that may be used within respective B2B-applications.

Ontologies may come in different flavors: “light-weight” ontologies rely on a small set of modeling primitives: concepts come with properties and relationships and are embedded in an is-a hierarchy. Such lightweight ontologies are comparable to conceptual database schemata. On the other end of the spectrum we have “heavy-weight” ontologies that come in addition with various types of rules to capture more semantic aspects of a given domain. The ontology-based B2B Broker of *ontoprise* supports the whole range of ontology types, from lightweight ones to full-fledged heavyweight ones. Thus the type of ontology can be adjusted to the specific needs of the B2B-application at hand.

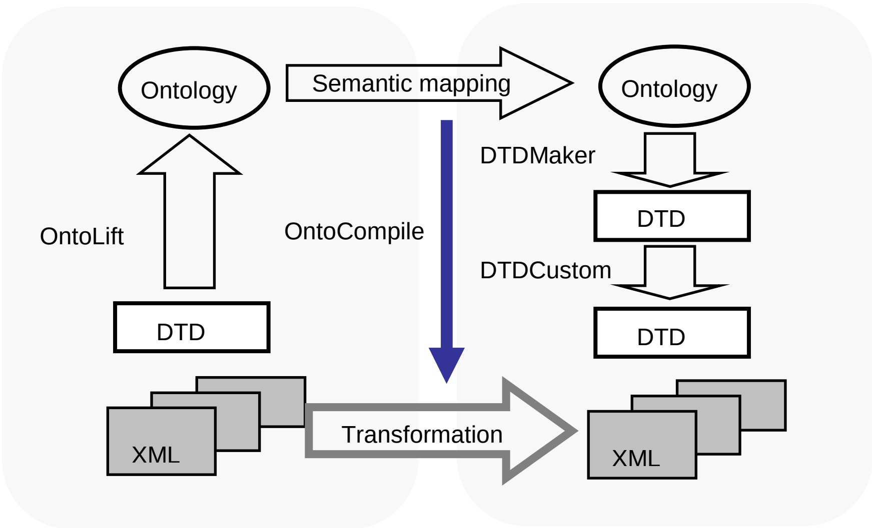

Figure 3: Transforming XML-Documents

Such an approach only makes sense if ontologies are smoothly integrated with other existing or emerging sources on the Web or an enterprise's intranet. Therefore tools should provide means for generating, handling or understanding ontologies in current web-based conceptual representation language, e.g. RDFS (RDF-Schema [Brickley, Guha 99]), OIL (Ontology Inference Layer, http://www.ontoknowledge.org), or the languages from the DAML-program (DARPA Agent Markup Language, http://www.daml.org). The presented Semantic B2B-Broker is compatible with these internationally relevant ontology specification languages as well as with new emerging language proposals that will be based on W3C-standards.

Due to the heterogeneous nature of the (semantic) web as depicted in the catalogue scenario, above, different kinds of settings may be distinguished that are all handled by the B^3 Semantic B2B-Broker.

4.1 Top-Down Setting

The top-down setting may be used in contexts in which a schematic description of the domain at hand is given or may be developed in the first place. This setting is typically applicable in situations in which a consortium or a committee has to define e.g. the structure of a new product catalogue or the structure of business objects to be exchanged between enterprises. Since syntactical structures, e.g. the structure of XML documents, are secondary to the real decisions concerning the relevant business objects, their properties and their relationships, such a committee can first construct an ontology to provide a shared understanding of the given task and domain. From such a conceptual model the Semantic B2B-Broker can then automatically generate corresponding XML-DTDs that reflect the conceptual model [Erdmann, Studer 01]. Thus, XML documents may be authored the tags of which are underpinned with a clearly defined semantics – as defined by the given ontology. In this way, “ontology-aware” applications may be developed that “understand” the rich semantics of these XML-documents. The top-down setting corresponds to the right part of Figure 3.

4.2 View Setting

The view setting may be used to realize B2B-solutions in which different stakeholders have to be provided with different views on the same information source, e.g. a product catalogue. In such situations the ontology of the information source may be semantically mapped to different target ontologies reflecting the stakeholders’ conceptual views of the original source. From each such target ontology a respective XML-DTD may then be automatically generated. Thus, instead of defining views on a syntactic XML level, views may be defined on a conceptual level. The view setting extends the top-down setting with the top-level semantic mapping as shown in the upper part of Figure 3.

4.3 Bottom-up Setting

In the bottom-up setting the situation is somewhat more difficult, because here in general two XML-DTDs and two conceptual models are involved that have to be related to each other (cf. left part of Figure 3). The bottom-up approach is relevant in situations in which e.g. several XML-based standards for B2B-applications already exist. As said, conceptual models are independent of the representation, thus the bottom-up setting may be applied to EDI sources as well. Thus EDI messages become potentially compatible with the XML world.

Within the bottom-up setting three steps must be conducted:

The conceptual reengineering step provides means to identify concepts, their properties and relationships from XML-DTDs or XML schemata. This reengineering step exploits a collection of heuristics that allow the identification of the conceptual structure underlying a given DTD. Using such heuristics suggestions for structuring the respective ontology are generated which can then be evaluated by the ontology builder.

Once the structures of the ontologies have been identified, techniques are offered within the Semantic B2B-Broker for the semi-automatic creation of mapping rules on that conceptual level. To facilitate these mappings top-level or meta-ontologies can provide an embracing vocabulary to simplify the mapping generation.

From those conceptual mapping rules, which establish a semantic correspondence between the syntactic DTD structures, syntactic transformations of the DTDs can be generated and specified in XSLT. The transformation rules have to be generated just once and are valid as long as the DTDs or XML-schemata are not changed.

The advantages of the described ontology-based B2B-solution may be summarized as follows:

- Ontologies provide a conceptual model of a B2B-application and define a shared vocabulary for the different involved stakeholders.
- The semantic mappings explicitly define the conceptual relationships that exist between different information sources that interact within a B2B-application.
- The generation of XML-DTDs from ontologies associates a precise meaning to the XML tags and the XML document structures. Thus XML modeling primitives are underpinned with a clear semantics.
- Inheritance as supported within ontologies avoids the redundancy as known from XML specifications.
- Heavyweight ontologies provide further means for deriving new facts and relationships from the explicitly given ones: thus queries are e.g. answered with more complete results when compared to the XML-based query results.

5 The *ontoprise* B³ B2B BROKER

5.1 Basic Tools

ontoprise offers various tools for the creation and manipulation of ontologies as well as of facts instantiating the concepts and relations of the ontologies.

OntoEdit is an editor to create and modify ontologies graphically. Concepts may be arranged in an is-a-hierarchy. For the definition of attributes and relationships appropriate forms are offered. OntoEdit is multilingual, i.e. concept names may be given in different languages and the user may switch between these languages. OntoEdit supports the definition of axioms and constraints. Often used axioms like for instance transitivity of binary relations may be defined by a mouse click; more complex axioms have to be given in F-logic [Kifer et al. 95], an object oriented and logic-based language from the deductive database community. A future version of OntoEdit will contain a graphical editor for axioms, which does no longer require any F-logic know-how from the user. OntoEdit is able to read and write different representation formats like RDFS, OXML (an XML representation of ontologies), DAML+OIL, or F-logic. RDFS and DAML+OIL are formats especially developed for the Semantic Web and will therefore play a major role in B2B applications in future.

OntoAnnotate allows the user to add meta information to a document (HTML, Word, PDF). For instance if the word “Peterson” appears in a document, a user marks this word in the document and attaches the attribute “last name” of “employee” to it (graphically by browsing in the ontology). Such annotated documents may then be used in different ways in Semantic Web applications. Firstly, such a document may be processed by a program called crawler, which extracts the information and in this way fills a corporate knowledge base with facts, which may serve as an organizational memory. Secondly, it may be transmitted to a B2B partner who can process the semantically founded annotations in the document automatically in addition to the textual, human readable content.

OntoBroker is the middleware run time system, which administers the knowledge base containing ontologies and facts. OntoBroker integrates a powerful reasoning mechanism (inference engine), which delivers answers to queries to the knowledge base. During this processing rules are evaluated which describe (domain specific) relationships between concepts and instances of concepts. Even, mapping relationships between different ontologies can be defined by rules. The inference engine then operationalizes these relationships, i.e. it maps instances of one ontology to instances of another ontology, i.e. it translates from one conceptual schema into another. In the given scenario it implements a transformation of a product catalogue described in terms of one ontology to a product catalogue described in terms of another ontology. Thus, translation on the conceptual level is achieved.

5.2 Advanced Tools

While the basic tools cover the creation and manipulation of ontologies, provide query mechanisms, and provide tools for annotating documents the advanced tools especially support the mapping problem (which occurs in many places in B2B applications) on a semantic and thus abstract level (cf. Figure 3).

5.2.1 Mapping XML Documents

Though XML provides a readable and understandable basic format to describe documents, it is focused on the representation of documents and not on the description of data, their relationships etc. For this purpose it lacks comfortable modeling primitives with a well-defined semantics. For mappings, which require the exploration of the data in a way independent of the document structure, transformations with XSLT are inadequate. Instead semantic mapping definitions based on conceptual models are needed to describe such transformations. Additionally, ontologies enable the generation of DTDs representing different views on the data.

Looking at the right part of Figure 3 in a top-down setting DTDMaker [Erdmann, Studer 01] offers the automatic generation of a set of XML-DTDs and XML-Schemas from an ontology. The resulting DTDs may be customized (DTDCustom), i.e. parts of an XML structure that are not needed for a specific application may be deleted. This top-down approach is especially appropriate if a conceptual model already exists, e.g., as an ontology, an object-oriented schema or an entity relationship schema.

In a bottom-up setting, i.e. in a situation where a given seller product catalogue has to be mapped to a given buyer product catalogue, the following steps have to be performed:

- OntoLift provides means to identify concepts and relationships from an XML-DTD or an XML-schema in order to derive an ontology of the given application domain. Thus OntoLift has to focus on the reengineering of conceptual models encoded in XML DTDs.
- OntoMap offers a collection of techniques for the semi-automated creation of mapping rules on the conceptual level. These techniques come with a graphical GUI for evaluating and applying the generated rules.
- Finally OntoCompile will be able to create a set of XSLT transformations, which perform transformations between XML documents with the required performance.

The rules that are used to transform the ontologies are directly operational since OntoBroker is able to interpret and execute these rules, i.e. XML instances from one DTD may be processed and translated into instances according to another DTD via the conceptual level. Therefore, even without the third step the transformation is already operational and may thus be evaluated and tested.

5.2.2 Mapping RDF Documents

Life becomes much easier if we are already on the semantic level. This is the case if we use a language like RDFS [Brickley, Guha 99] to describe an ontology and a semantically well defined representation language for ground facts, like RDF [Lassila, Swick 99]. If we want to transform RDF documents, the first step to lift a weak description (e.g. a DTD) to a conceptual level may be dropped and the mapping of one ontology to another may be defined on the semantic level directly. Again OntoMap plays a crucial role in this process. OntoMap will be able to apply heuristics for this mapping process and to provide a GUI, which allows the manual creation and manipulation of such mappings. As in the XML case, with OntoBroker these mappings are immediately operational and may thus be tested. In a further process these mappings may be compiled into a running program, which enables high performance transformations of RDF documents.

6 Related Approaches and Conclusion

Basically there are two different classes of related approaches. In one class methods reside which form a general framework for mappings in the B2B or B2C area. They are based on representation languages and formats and provide methods and tools to map between them. The other class is characterized by sets of predefined models and formats for different application domains.

BizTalk [Microsoft 00] is a framework based on XML for application integration and electronic commerce. It provides tools to generate XML schemas and define mappings between schemas. These mappings are then operationalized using XSLT. BizTalk supports different formats and protocols like EDI, EDIFACT and different Web protocols. It is an open architecture, which allows developers to build adapters to their products. It contains a server, which receives and distributes orders. Clearly BizTalk falls in the first class mentioned. Though, it provides a general toolset for the definition and mapping of business data it is in contrast to our approach based on a low semantic level of representation, namely pure XML without further conventions. As a consequence XSLT is used for mappings. Though, the mapping between documents is supported by graphical tools these mappings are developed in a “programming” like process based on the document structure instead of defining the mappings on an abstract semantic level based on the meaning of the documents. Nevertheless the BizTalk server would be an ideal completion for our approach. While we add the semantic level and tools on an abstract user-friendly level, the server would process our mappings and handle transactions. Additionally the server would provide access to other applications and formats.

Special catalogue formats like the Open Catalog Format (OCF, http://www.martsoft.com/ocp) or BMEcat [Hümpel et al. 99] specify proprietary XML document structures for representing catalogue data. These proprietary formats are incompatible with each other, i.e. no BMEcat-application can process OCF-data and vice versa. Both approaches, nevertheless, contain one aspect that is a little bit closer to a semantic level as presented in this paper, i.e. the product hierarchy of OCF and the feature system of BMEcat and its associated product classification system eCl@ss (http://www.eclass.de) can be interpreted as "light weight ontologies". The diversity of syntaxes, on the other hand, hinders easy interoperability of different systems and approaches -- a main problem of the electronic markets, today.

Commerce XML (cXML) is being developed by Ariba. It defines a collection of DTDs for business objects and a basic communication model to exchange such business objects. The exchange formats like *order* are based on predefined DTDs. In contrast to the previous approaches it is a normative approach and falls into the second class mentioned. Thus, it is assumed that everybody presents its data in one of the predefined business objects and exchanges its data in a predefined exchange format. Thus, there is no need for a sophisticated mapping approach. This approach works very well as long as there is no need for more flexibility in the business object formats or in the exchange formats. The EDI history has shown that this does not work in general and that the need for more flexibility will arise. As soon as this need comes up this approach lacks of the semantic level to describe add-ons and the mappings between (then incompatible) objects.

B2B applications in general have to relate data that is exchanged between suppliers and buyers on a semantic level. Only then, all stakeholders will interpret exchanged data in the same way. Whereas XML provides a significant step in standardizing the syntactic structure of this data, ontologies provide the required semantic means to characterize the exchanged data on a conceptual and thus semantic level. The B2B Broker offers exactly the functionalities that are needed to provide the semantic underpinning for the exchanged XML data.

References

[Berners-Lee 99] T. Berners-Lee: Weaving the Web. The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. Harper, San Francisco, CA, 1999.

[Biron, Malhorta 00] P.V. Biron, A. Malhotra (eds.): XML Schema Part 2: Datatypes. W3C Candidate Recommendation. 24. October 2000.

[Brickley, Guha 99] D. Brickley, R.V. Guha: Resource Description Framework (RDF) Schema Specification. W3C Candidate Recommendation, 27. March 2000.

[Bray et al. 98] T. Bray, J. Paoli, C.M. Sperberg-McQueen (eds.): Extensible Markup Language (XML) 1.0. W3C Recommendation, 10. February 1998.

[Clark 99] J. Clark (ed.): XSL Transformations (XSLT) Version 1.0. W3C Recommendation, 16. November 1999.

[Decker et al. 00] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, I. Horrocks: The Semantic Web: The Roles of XML and RDF. in: IEEE Internet Computing, September/October 2000. pp. 63-74.

[Deutsch et al. 99] A. Deutsch, M. Fernández, D. Florescu, A. Levy, D. Suciu: A Query Language for XML. in: World Wide Web Conference (WWW-8), 1999.

[Erdmann, Studer 01] M. Erdmann, R. Studer: How to structure and access XML documents with ontologies. in: Data and Knowledge Engineering, Volume 36, Issue 3, Elsevier, March 2001. pp. 317-335.

[Fensel et al. 00] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, M. Klein: OIL in a Nutshell. in: Proceedings of the 12th International Workshop on Knowledge Engineering and Knowledge Management (EKAW 2000), Juan-les-Pins, France, October, 2000.

[Gruber 93] T.R. Gruber: A Translation Approach to Portable Ontology Specifications. in: Knowledge Acquisition. 5(2), Academic Press, 1993. pp. 199-220.

[Hümpel et al. 99] C. Hümpel, T. Renner, V. Schmitz: Spezifikation BMEcat. Version 1.01. Bundesverband Materialwirtschaft, Einkauf und Logistik e.V.; Fraunhofer IAO, Stuttgart; Universität Essen, BLI, 1999.

[Kifer et al. 95] M. Kifer, G. Lausen, J. Wu: Logical Foundations of Object-Oriented and Frame-Based Languages, Journal of the ACM, 42, 1995.

[Lassila, Swick 99] O. Lassila, R.R. Swick: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 22. February 1999.

[Microsoft 00] Microsoft Corp.: BizTalk Framework 2.0: Document and Message Specification. December 2000

[Robie et al. 98] J. Robie, J. Lapp, D. Schach: XML Query Language (XQL). in: Proceedings of the W3C Query Language Workshop (QL-98), Boston, December 1998.

[Thompson et al. 00] H.S. Thompson, D. Beech, M. Maloney, N. Mendelsohn (eds.): XML Schema Part 1: Structures. W3C Candidate Recommendation. 24. October 2000.