

WHITEPAPER

B³

SEMANTIC **B2B** BROKER

ontoprise GmbH vision and
emerging products document

December 2000



ontoprise
Semantics for the Web

CONTENT

1. Introduction	3
2. Example Scenario: Product Catalogues	3
3. Current Approaches	4
3.1 EDIFACT	4
3.2 XML	5
4. Ontology-based Approaches	8
5. The ontoprise B ³ B2B Broker	11
5.1 Basic tools	11
5.2 Advanced tools	11
5.2.1 Mapping XML documents	11
5.3 Mapping RDF documents	13
6. Conclusion	13

ontoprise

ontoprise GmbH
Haid-und-Neu-Strasse 7
D- 76131 Karlsruhe

Tel.: +49-(0)721-6635933
Fax.: +49-(0)721-6635934
eMail: info@ontoprise.de
www.ontoprise.de

1 Introduction

The development of the World Wide Web changes the business relationships between enterprises. Already the "First Generation Web" that puts emphasis on handling document *structures* had a significant impact on the way enterprises exchange information among each other: HTML or XML make business information exchangeable in a platform independent way. However, since HTML or XML only describe the *structure* of documents in a machine-processable way, this "First Generation Web" does not provide the support that is needed in the end. Rather, the notion of the "Semantic Web" as coined by Tim Berners-Lee paves the way to make the *semantics* of documents or more generally speaking the semantics of business information sources machine-processable. 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 products of ontoprise will provide advanced support for developing and managing these new kinds of semantic B2B-solutions.

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 (cf Figure 1): 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.

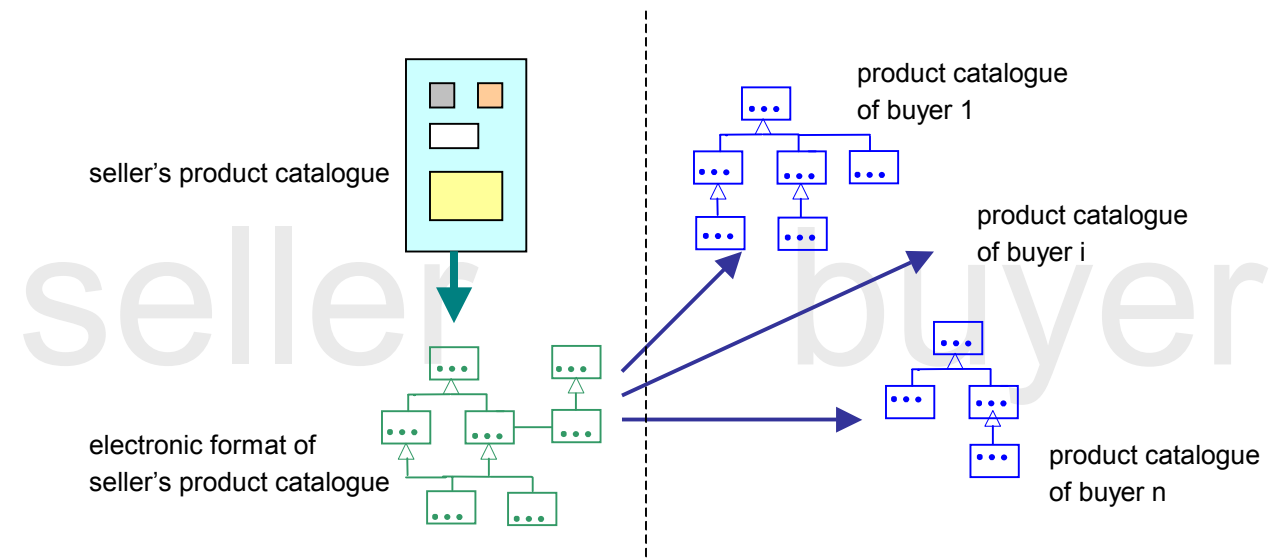
Subsequently, we will use the product catalogue of a computer dealer for further illustration. Among others this catalogue contains data about printers. A printer is described by its name, its price, its features, its producer and its type, i.e. whether it is a laser printer, an ink printer or a matrix printer. Our dealer is located in 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, its catalogue is structured in

another way: 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 transformations.

Figure 1 Product Catalogue Translation Steps



3 Current Approaches

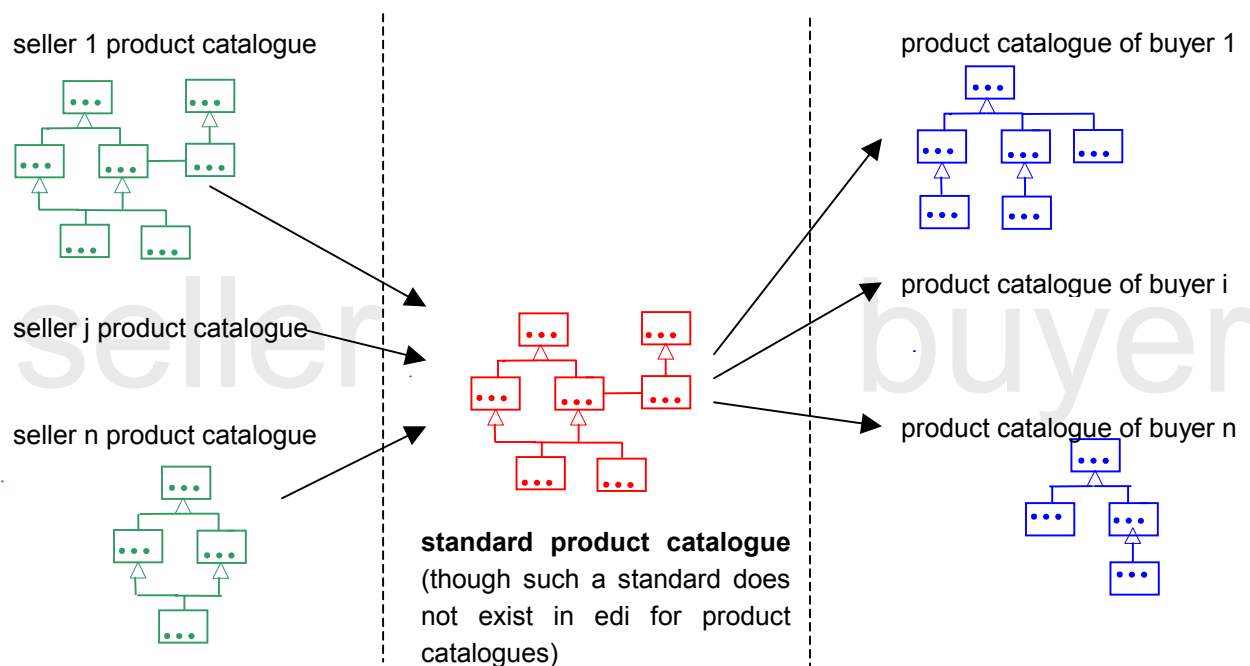
3.1 EDIFACT

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 (compare Figure 2). 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, EDI is not the final solution for data exchange due to the following reasons:

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

Figure 2 Product Catalogue Translation Steps based on a Standard



3.2 XML

The emerging standard for data exchange is XML. 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 3 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 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 understandable (and readable) 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.

Figure 3 XML Descriptions of a printer

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

To handle the heterogeneity of XML sources within B2B-applications, one can exploit XSLT 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 she 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, 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 an integrated offer for a computer, a screen etc., the price of the mouse may only be 1/100 of the price of the entire package.

- 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 come from different XML sources.
- Query languages like XQL and XML-QL allow to query the document structure of the XML document, but not its semantic content. For instance such languages do not naturally allow to pose 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-catalog format, such as BMEcat; here the XML-structure of the catalog only offers very general means to provide information about possible relationships between the different articles contained in the catalog, 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 kind 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 B³, the Semantic B2B Broker of ontoprise: we will subsequently first describe the logical architecture (see Section 4) and then describe the specific tool-set of the B2B Broker (see Section 5).

4 Ontology-based Approaches

Based on the analysis of the strengths and weaknesses of XML-based applications, ontoprise developed a new level of solutions for B2B-applications: the ontology-based Semantic B2B Broker that will be described on a tool level in section 5. 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 4).

In a business context, an *ontology* is defined as a "formal, explicit specification of a shared conceptualization" (Tom Gruber, 1993). I.e. 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 light-weight 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 light-weight ones to full-fledged heavy-weight ones. Thus the type of ontology can be adjusted to the specific needs of the B2B-application at hand.

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, the ontology engineering environment OntoEdit supports the language-independent modeling of ontologies. Dependent on the given context, OntoEdit provides means for subsequently generating ontologies in the most convenient representation language, e.g. RDFS (RDF-Schema) (<http://www.w3.org>), OIL (Ontology Inference Layer), (<http://www.ontoknowledge.org>), DAML (DARPA Agent Markup Language) (<http://www.daml.org>), or Frame Logic. Thus, ontologies as handled within the Semantic B2B-Broker are compatible with all internationally relevant ontology specification languages as well as with new emerging language proposals that will be based on W3C-standards.

Within the Semantic B2B-Broker different kinds of settings may be distinguished:

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. 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 Figure 4.

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 automatically be 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 Figure 5.

Figure 4: Top-Down setting

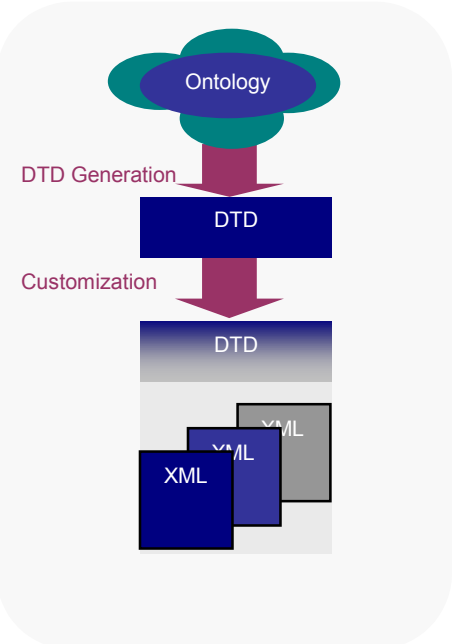


Figure 5 : View setting

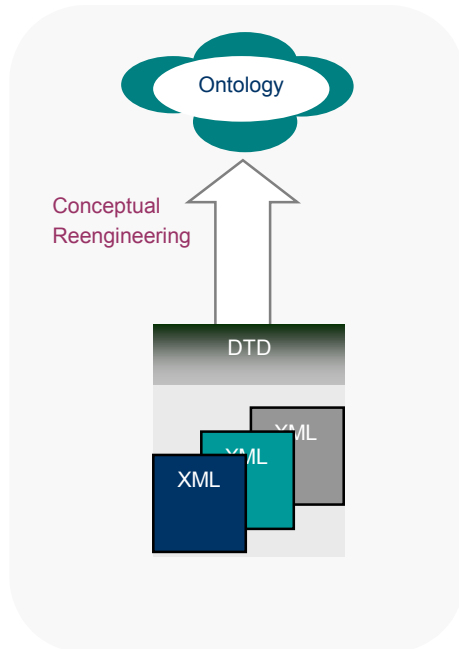


Bottom-Up setting:

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. Figure 6). 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.

Figure 6: Bottom-Up setting



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 define a conceptual specification of the relations 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.
- + Heavy-weight 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 concept and relations of the ontologies.

OntoEdit is an editor to create and modify ontologies graphically. Concepts may be arranged in an 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, complexer axioms have to be given in F-Logic. 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, OIL, F-Logic. RDFS and OIL are formats especially developed for the Semantic Web and will therefore play a major role in B2B applications in future.

OntoFact is a tool that automatically generates forms for the attributes and relations of instances out of an ontology. Thus facts may be added, modified or deleted.

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 three different ways. First such a document may be processed by a program called crawler which extracts the information and by this way fills the knowledge base with facts. Second it may be used as a free form for facts, i.e. changing the name "Peterson" to "Miller" in an annotated document does not destroy the annotation and may thus be used as an additional fact source. Third annotated documents may be used as a template to acquire information in similarly structured documents. For this goal OntoAnnotate serves as a definition tool for our tool OntoWrapper.

5.2 Advanced tools

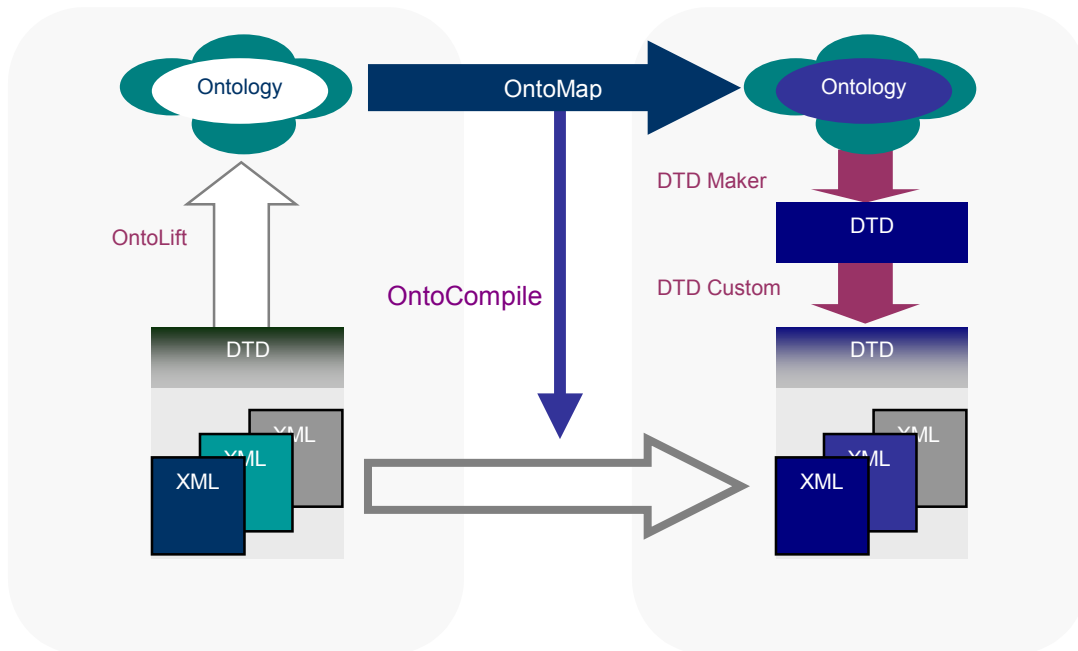
While the basic tools cover the creation and manipulation of ontologies, provide query mechanisms and provide tools for annotating documents the advanced tools especially support the mapping problem which occurs in many places in B2B applications on a semantic and thus abstract level (see Figure 7).

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 of 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 adequate to

describe such transformations. Additionally ontologies enable the generation of DTDs representing different views on the data.

Figure 7 : Transforming XML Documents



Looking at the right part of Figure 7 in a top-down setting DTDMaker 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:

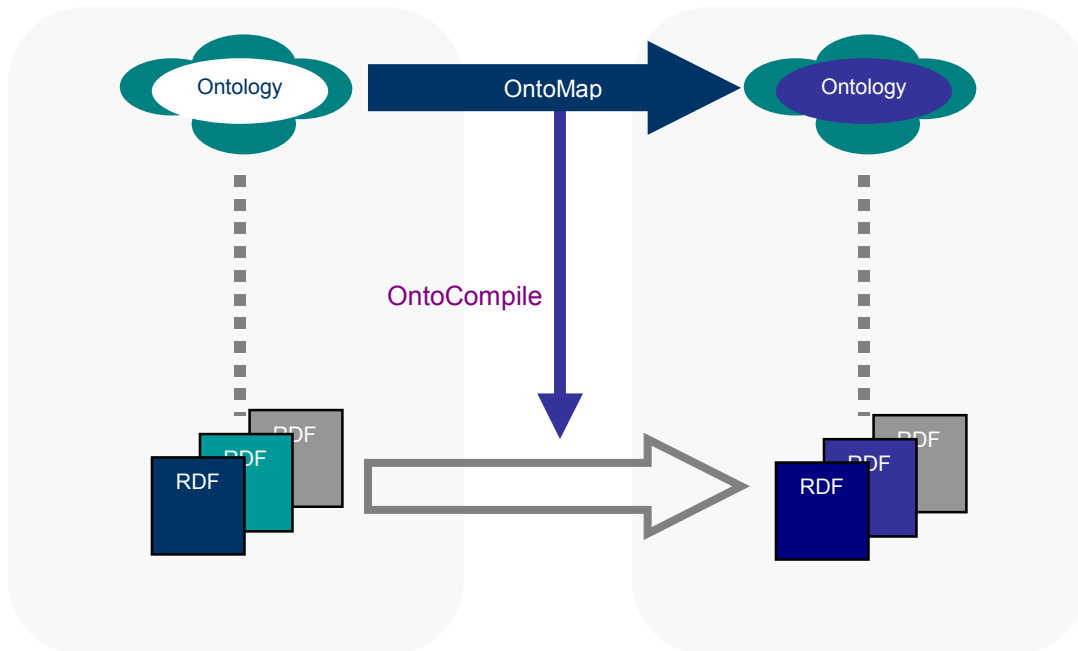
OntoLift provides means to identify concepts and relationships from an XML-DTD or an XML-schema in order to derive an ontology of the given application domain. Thus OntoLift has to focus on the reengineering of conceptual models encoded in XML DTDs. (ii) OntoMap offers a collection of techniques for 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. (iii) 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 OntoInfer is able to interpret and execute these rules. Therefore, even without step (iii) the transformation is already operational and may thus be evaluated and tested.

5.3 Mapping RDF documents

Life becomes much easier if we are already on the semantic level. This is the case if we use RDF-S to describe an ontology and we want to transform corresponding RDF documents. Thus the step to lift a weak description (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 OntoInfer 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.

Figure 8 : Transforming RDF Documents



6 Conclusion

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