AN XML-BASED INTERORGANIZATIONAL KNOWLEDGE MEDIATION SCHEME TO SUPPORT B2B SOLUTIONS

S. Castano¹ V. De Antonellis² S. De Capitani di Vimercati²

(1) *Università di Milano, Milano, Italy*

(2) *Università di Brescia, Brescia, Italy*

Abstract We propose an *Interorganizational Knowledge Mediation Scheme* (IKMS) acting as an information infrastructure for B2B solutions for aggregating exchange information supplied by the companies participating in a trading community in order to support e-business processes. The proposed IKMS is a set of mediated descriptions of exchange information at the global level, defined as *global X-classes*, and relies on a set of techniques for building such global X-classes.

1. Introduction

Nowadays many companies plan to use e-commerce and, in particular, to implement business-to-business (B2B) integration solutions to work more closely with trading partners to improve efficiency and profitability. B2B integration has become increasingly dependent on the Internet and requires companies to be able to share and exchange information between their systems to conduct business together over the Web [18]. However, exchanging information between different information systems is quite complex. Often problems in data exchange result in recoding data into the formats of the different systems, which leads to delays as well as errors in the data. Other problems are related to the cost for data exchange; the number of wrappers that need to be written increases exponentially with the number of systems between which one wishes to transfer information. XML [17] is viewed by many as a way to solve this problem. As a consequence, companies are beginning to use XML to send application data to browsers and to business applications. B2B e-commerce is thus driving a new generation of Internet applications that use XML as common data language for information sharing and exchange between companies [5].

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35658-7_21](https://doi.org/10.1007/978-0-387-35658-7_21)

R. Meersman et al. (eds.), *Semantic Issues in E-Commerce Systems*

© IFIP International Federation for Information Processing 2003

XML has been initially proposed for publishing documents over the Web. XML offers powerful flexibility for defining tags that allow data to be structured and provides a description for that structure. This makes documents more than just a plain text but with a certain degree of intelligence. Therefore, XML is well suited for the interchange of data (XML documents are self-describing, easily parsed and can represent complex data structures) and gives a way of representing company data in an application-independent way. Also, there is a wide variety of tools for parsing and transforming XML documents meaning that applications do not have to perform their own structural validation, reducing so the cost of building applications. Furthermore, the development of B2B standards built on XML will accelerate this adoption and tightly integrate companies around well-defined business processes [16]. For instance, the OASIS organization is targeted to develop interoperable industry specifications based on public standards (XML and SGML) such as ebXML, a global framework for electronic business data exchange [15].

While the adoption of XML can greatly alleviate language incompatibility between disparate applications, a barrier that must be overcome is that B2B applications can only aggregate business processes if there is an ability to mediate between diverse company systems and conceptual contexts, by identifying and organizing the so called interorganizational knowledge. Some initial results for this problem have been proposed for the construction of enterprise ontologies, both in field of international standards [1] and in the field of industrial solutions (www.ontology.org). It is widely recognized that a major challenge that an e-business solution must address to support interorganizational cooperation and business aggregation is the semantic heterogeneity of diverse systems across a trading community, and integrated models and methods to design and develop the interorganizational knowledge are needed.

In this field, we consider an XML-based B2B integration scenario and our contribution is devoted to the development of methods and tools for building an *Interorganizational Knowledge Mediation Scheme* (IKMS). The IKMS acts as an information infrastructure for aggregating exchange information supplied by the companies participating in a trading community in order to support e-business processes. In defining the IKMS, we focus on XML data organization and exploit our previous research work for defining global views of heterogeneous structured datasources [2–3]. The proposed IKMS is a set of mediated integrated descriptions of exchange information at the global level, defined as *global X-classes*. Data integration issues are well known in the area of databases and are becoming more and more important due to the availability of multiple datasources over the Web [11–12]. The IKMS provides

a single point of access to information within and among companies and allows broad access to both structured and unstructured information underlying the global X-classes (e.g., databases or data warehouses, as well as word documents and graphics), for supporting e-service aggregation and cooperation, and for facilitating e-service distribution.

1.1 Related Work

We briefly describe the main contributions [7, 10, 14] in the data integration area.

TSIMMIS is a mediator system being developed by the Stanford database group, in conjunction with IBM [10]. A major focus of the project is the development of tools to speed up the integration process by extracting properties from unstructured or semi-structured sources with no well-defined static schema. The key to the TSIMMIS approach is the *Object Exchange Model* (OEM), in which data (OEM objects) have the structure (object-id, label, type, value). In addition to the OEM, a major contribution of TSIMMIS is the *Mediator Specification Language* (MSL), a high-level declarative language for integrating data-sources. MSL uses prolog-like rules and functions for translating objects. The tail of a rule specifies patterns found in the sources, while the head describes patterns of the top-level objects of integrated views.

HERMES is a mediator system being developed at the University of Maryland [14]. Unlike TSIMMIS, which integrates semi-structured data-sources, HERMES focuses on integrating knowledge bases and reasoning systems. Integration is achieved by hooking up each component system to the semantic model, called the *generalized Annotated Program* (gAP) framework, an extension of logic programming. HERMES defines a rule-based mediator language, implemented within a Mediator Programming Environment. The Mediators are written as prolog-like clauses, with two special predicates: `in()`, executes a select statement on the target data source; `=()`, tries to unify two values.

The Information Manifold (IM) is a decision-logic system for integrating web-based information sources [7]. As in a mediator system, the end-user specifies queries in a declarative way against a static view. Unlike TSIMMIS, however, the IM presents the user with a single global view, called the *World View*, which is a collection of virtual relations and classes that describes the contents of the information sources. The IM uses a relational data model, augmented with class hierarchies. A key feature of the IM model is that classes can be declared to be disjoint, guaranteeing that no object can belong to both.

The three systems described above define the cutting edge of integration research, and all of them seek to provide a framework for schematic integration while maintaining the autonomy of the component systems. Recently, a lot of companies are currently implementing a B2B infrastructure that will allow them to interact with several thousands of business partners that include suppliers, distributors, marketplaces, and financial institutions. A significant part of the architecture is dedicated to the exchange of business documents to support interactions between business partners. XML is viewed by many as a key enabler to encode the information and support interactions between different partners in a way that can be understood by all of them. This view is reflected by today's B2B architectures and products. However, XML itself does not guarantee that different organizations will understand a piece of information. Each partner has to share the same vocabulary and meaning before information can be exchanged. Specifications like *ebXML* [15], *RosettaNet* [16], and commercial organization (e.g., CommerceOne and Ariba) define an abstraction layer on top of the Internet core standards that allows corporations to establish non-negotiated relationships with their business partners over the Internet. It would be almost impossible for any company, regardless of its size, to negotiate the nature of the relationships with hundreds or thousands of partners. For instance, RosettaNet's trading partner agreement includes message formats, sequences of messages, and an implementation framework that specifies the physical means of exchanging message. Despite perceived benefits, in all these cases, the role of XML seems limited to formatting data during interchanges of business documents such as purchase orders, invoices, request for quotes.

1.2 Organization of the paper

The paper is organized as follows. Section 2 illustrates the considered XML-based B2B integration scenario. Section 3 describes the X-formalism used to abstract DTDs. Section 4 introduces global X-classes and describes integration techniques. Section 5 discusses the use of the IKMS in the B2B scenario for querying and browsing in the trading community. Finally, Section 6 presents a few concluding remarks.

2. The XML-based B2B integration scenario

In this section, we characterize the requirements of the considered B2B integration scenario and the proposed responding approach.

The architecture of our system is illustrated in Figure 1. We assume that each company datasource uses a DTD to specify the structure of

Figure 1. Three-tier architecture based on IKMS

information to be exchanged within the trading community. In this way, our approach can support a wide variety of different information models, including relational tables, HTML and XML documents, since DTDs either are directly available or can be easily derived. For instance, an XML document may include a reference to a pre-existing DTD. However, since the DTD is not mandatory, due to the tagging format of XML documents, a minimal implicit DTD can be easily derived by a syntactic check of the document [8]. In case of HTML documents, wrapping tools for generating the XML representation of an HTML document are available, and DTDs can thus be derived from the XML representation [13, 6]. In case of structured databases, a DTD describing the schema of data portion(s) to be exported can be defined. After that each company datasource has defined its DTD, the *IKMS builder* takes these DTDs as input, and builds a set of global *X-classes* which constitute the IKMS. In particular, the proposed approach for building the IKMS is articulated in three steps: (1) **DTD description**, where DTDs associated with datasources are abstracted into a set of *X-classes* according to the proposed X-formalism (see Section 3); (2) **Cluster derivation**, where *X-classes* of different company datasources are grouped on the basis of semantic mappings existing between them (see Section 4.1); (3) **Cluster reconciliation**, where *X-classes* belonging to a same cluster are reconciled into *global X-classes* (see Section 4.2).

```

<!DOCTYPE products [
<!ELEMENT products (computer+,software)>
<!ELEMENT computer (PC)+>
<!ELEMENT software (item)+>
<!ELEMENT PC (name,spec,price)>
<!ELEMENT item (name,note,price?,tohome?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT spec (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT tohome (#PCDATA)>
<!ELEMENT note (#PCDATA)>
<!--ATTLIST tohome
  xmlns:xlink CDATA #FIXED
  "http://www.w3.org/1999/xlink"
  xlink:href CDATA #REQUIRED-->
<!--ATTLIST computer
  brand CDATA #REQUIRED-->
]>

```

(a)

(b)

(c)

Figure 2. An example of DTD and its graphical representation (a) and the graphical representation of two additional DTDs (b)-(c)

The client applications can then pose requests on the global X-classes; the middle-tier handles these requests, shielding them from the complexity involved in dealing with back-end systems and databases.

3. The X-formalism for DTD description

To facilitate the analysis of DTDs associated with different company datasources, we introduce the *X-formalism* according to which DTDs are abstracted at a conceptual level. In the X-formalism, we provide a set of concepts, namely *X-class*, *property*, *referenced X-class*, *link*, *attribute*, and *class structure*, capable of describing in a high level the content of a given DTD. In the following, examples refer to the DTD reported in Figure 2(a).

X-class An X-class represents a set of elements that have a common structure. A structure is defined in terms of names of *properties* and/or names of other *referenced X-classes*. For instance, *products*, *computer*, *software*, *PC*, and *item* are the X-classes associated with the considered DTD. An X-class corresponds to an element declaration with element content in XML, that is, an element that includes other child elements.

Property A property of an X-class describes a characteristic of the elements represented by the X-class. It is characterized by a name and a type (PCDATA, any, or empty). For instance, name and spec are two properties of type PCDATA associated with the X-class PC. A property of an X-class corresponds to an element with PCDATA, any, or empty content in XML.

Referenced X-class A referenced X-class xc' for an X-class xc is an X-class appearing in the structure of xc and describes a relationship between xc and xc' . For instance, item is the referenced X-class associated with the X-class software. A referenced X-class corresponds to an element declared with element content in the structure of a considered element.

Link A link is used to describe a relationship between different documents and is characterized by a name and a type (PCDATA, any, empty, or a class structure – see definition below). For instance, tohome is a link, specified in the X-class item, of type PCDATA. A link corresponds to an *XLink element* [4] in XML.

Attribute An attribute describes additional featuring information of an X-class, property, or link. An attribute is characterized by a name, which must be different from the name of other attributes in the same class/property/link, and is defined on an *attribute type*, which will be specified later on, denoting the set of admissible values for the attribute. For instance, brand is an attribute of the X-class computer whose type is “character data”. An attribute associated with an X-class (property or link, resp.) corresponds to an attribute defined in the attribute list declaration of an element in XML.

Before giving a formal definition of X-class, we introduce a definition of *class structure* as follows. We assume two sets $\mathcal{N}$ and $\mathcal{A}$ of element and attribute names, respectively, are given. Any name $n \in \mathcal{N}$ is a structure. If $s_1, \dots, s_k$ are structures, $(s_1, \dots, s_k)$ is a structure called *sequence structure*. If s_i and s_j are structures, or s_i is a structure and $s_j = \text{PCDATA}$ (or vice versa), $(s_i \mid s_j)$ is a structure called *union structure*. Any structure s so defined may be followed by the cardinality symbols ‘+’, ‘*’, and ‘?’. In the following, $\mathcal{S}$ will denote the set of structures. The cardinality symbols used in the class structure specification represent cardinality constraints associated with properties or (referenced) X-classes. In particular, ‘+’ corresponds to the cardinality constraint (1,n), ‘*’ to (0,n), and ‘?’ to (0,1), respectively. If no symbol is specified, the cardinality constraint (1,1) is taken. Cardinality constraints can also be derived for attributes by taking into account their type. Let $AT = \{\text{CDATA}, \text{ID}, \text{IDREF}, \text{IDREFS}, \text{ENTITY}, \text{ENTITIES}, \text{NMTOKEN}, \text{NMTOKENS}, \text{enumerated}\}$ be

the set of XML attribute types¹. The types `ID`, `CDATA`, `IDREF`, `ENTITY`, `NMTOKEN`, and `enumerated` correspond to the cardinality (1,1) or (0,1), if the attribute is marked as `required` or `implied`, respectively. The types `IDREFS`, `ENTITIES`, and `NMTOKENS` correspond to the cardinality (1,n) or (0,n), if the attribute is marked as `required` or `implied`, respectively. In the discussion, we will use *CARD* to denote the set of cardinality constraints, that is, $CARD = \{(0,1), (1,1), (0,n), (1,n)\}$. A X-class is formally defined as follows.

DEFINITION 1 (X-CLASS) *An X-class xc is a 6-tuple of the form $\langle c, s, P, L, R, A \rangle$, where:*

- $c \in \mathcal{N}$ is the class name;
- $s \in \mathcal{S}$ is the class structure;
- P is a set, possibly empty, of properties of the form (p, t, A) , where $p \in \mathcal{N}$ is a name appearing in s ; $t \in \{\text{PCDATA}, \text{empty}, \text{any}\}$; and A is a set, possibly empty, of triples of the form (a, t, k) , with $a \in \mathcal{A}$, $t \in AT$, and $k \in CARD$, corresponding to the set of attributes of property p .
- L is a set, possibly empty, of links l such that $l \in \mathcal{N} \cup \mathcal{A}$ and l is a class name in s or is an attribute of properties/classes in $(P \cup R)$.
- R is a set, possibly empty, of referenced X-classes r such that $r \in \mathcal{N}$ and r is a class name appearing in s .
- A is a set, possibly empty, of attributes of the form (a, t, k) , where $a \in \mathcal{A}$, $t \in AT$, and $k \in CARD$, corresponding to the set of attributes of c . □

For instance, the X-class $\langle \text{computer}, (\text{PC}+), \{ \}, \{ \}, \{\text{PC}\}, \{(\text{brand}, \text{CDATA}, (1,1))\} \rangle$ contains one reference class `PC` and has an attribute `brand` with type `CDATA` and cardinality (1,1).

Intuitively, each concept above-mentioned can be graphically represented as a node in a graph. In particular, each X-class, property, and attribute is represented as a rectangular, oval, and double oval node, respectively. A link is represented as an oval (or rectangular, depending on its type) labeled with the name of the link and connected by a double directed arc to the appropriate X-class node. There is an arcs between two nodes if there is a containment relationship between them.

¹We use `enumerated` to represent any possible list of names from which the value of an attribute must be taken.

A union structure is represented as an or-labeled dashed arc crossing the arcs entering the class/property nodes involved in the union. A sequence structure with a cardinality constraint applying to the whole structure is represented as an and-labeled arc crossing the arcs entering the involved class/property nodes. Finally, cardinality constraints associated with properties, links, attributes, and referenced classes are explicitly represented, if different from (1,1), as labels attached to arcs; (1,1) is implicitly taken as a default. Figure 2 illustrates the graphical representation of three DTDs.

4. Building the IKMS

To support mediation among multiple and heterogeneous company datasources, we exploit the knowledge provided by X-classes to construct a set of *global X-classes* that constitute the IKMS. Global X-classes provide a reconciled description and a set of references to X-classes of different datasources that describe common elements.

The construction of global X-classes involves problems related to the identification of *clusters*, that is, sets, of semantically related X-classes defined in different company datasources. To this purpose, we rely on reference ontology $\mathcal{O}$ for the domain under consideration [3]. The ontology is useful to derive clusters of X-classes that describe the same concept (*cluster derivation* step), and construct global X-classes from clusters by finding a reconciled representation for properties and attributes (*cluster reconciliation* step).

According to the definition of X-class introduced in the previous section, we are now ready to formally define the concept of global X-class.

DEFINITION 2 (GLOBAL X-CLASS) *Given a cluster cl of X-classes, the global X-class, denoted g_{cl} , obtained from the X-classes in cl is a 6-tuple of the form $\langle gc, GP, GL, GR, GA, cl \rangle$, where (1) gc is the global class name derived from the names of the X-classes in cl ; (2) GP is a set, possibly empty, of properties obtained from the reconciliation of properties of the X-classes in cl ; (3) GL is a set, possibly empty, of links obtained from the reconciliation of links of the X-classes in cl ; (4) GR is a set, possibly empty, of referenced global X-classes obtained from the reconciliation of referenced classes of the X-classes in cl ; (5) GA is a set, possibly empty, of attributes obtained from the reconciliation of attributes of the X-classes in cl ; and (6) cl is the cluster from which the global X-class g_{cl} is derived. $\square$*

To illustrate the cluster derivation and reconciliation steps in more details, suppose that we want to create a simple B2B portal for companies

operating in the electronic selling of computers marketplace. For simplicity, we consider only the DTDs of three company datasources, represented in Figure 2. In the following, we refer to these three datasources as S_1 , S_2 , and S_3 , respectively.

4.1 Cluster derivation

Clusters are identified on the basis of semantic mappings among X-classes of different datasources. Semantic mappings are discovered by comparing X-classes using the knowledge provided by the reference ontology $\mathcal{O}$ [9].

The ontology is exploited for assessing the level of semantic relationship between X-classes, called *class affinity* [3]. The ontology-based affinity evaluation process is automatically supported by procedures that we have developed in the framework of the ARTEMIS tool environment [2]. By exploiting the semantic relationship between X-classes, *semantic mappings* can be established between X-classes. More precisely, given two X-classes $xc_1 = (c_1, s_1, P_1, L_1, R_1, A_1)$ and $xc_2 = (c_2, s_2, P_2, L_2, R_2, A_2)$, a *semantic mapping*, denoted $xc_1 \leftrightarrow xc_2$, is established if their names have affinity, that is, $c_1 \sim c_2$. For instance, a semantic mapping can be established among X-classes *computer* of datasource S_1 and *machine* of datasource S_2 .

By using semantic mappings, we define clusters of X-classes as follows. For any X-class xc , the cluster for xc , denoted $[xc]$, contains all X-classes having a semantic mapping with xc . Formally, $[xc] = \{xc' \leftrightarrow xc\}$. In particular, *well-formed* clusters, denoted $[]^{wf}$, contain at least one X-class from each company datasource. For instance, a well-formed cluster for datasources in Figure 2 is $[S_1.computer]^{wf} = \{S_1.computer, S_2.machine, S_3.computers\}$.

4.2 Cluster reconciliation

Given a cluster $[xc]$ of X-classes, its representative global X-class $g_{[xc]}$ is derived by reconciling the properties, links, referenced classes, and attributes of the X-classes in $[xc]$ that have a semantic mapping. To this purpose, a set of *reconciliation rules* is used. Reconciliation rules establish how to derive global features by mediating names, types, and cardinality constraints of X-classes in a cluster. Here, the term feature is used to denote a property, link, referenced class, or attribute of a given X-class.

Name reconciliation. The mediated name of two features f_1 and f_2 can coincide with the name of one of them, or can be one of their

	PCDATA	empty	any $s \in \mathcal{S}$
PCDATA	PCDATA	PCDATA	any ($s PCDATA$)
empty	PCDATA	empty	any s
any	any	any	any any
$s \in \mathcal{S}$	($s PCDATA$)	s	any rules

	CDATA	ID	IDREF(S)	ENTITY(IES)	NMTOKEN(S)
CDATA	CDATA	CDATA	CDATA	CDATA	NMTOKEN(S)
ID	CDATA	ID	CDATA	CDATA	NMTOKEN(S)
IDREF(S)	CDATA	CDATA	IDREF(S)	CDATA	NMTOKEN(S)
ENTITY(IES)	CDATA	CDATA	CDATA	ENTITY(IES)	NMTOKEN(S)
NMTOKEN(S)	NMTOKEN(S)	NMTOKEN(S)	NMTOKEN(S)	NMTOKEN(S)	NMTOKEN(S)

Figure 3. Rules for type reconciliation

hypernyms or synonyms, selected by exploiting the relationships in the ontology $\mathcal{O}$.

Type reconciliation. The mediated type of two features f_1 and f_2 coincides with the type of f_1 (or f_2), if they have the same type. Otherwise, the selected type is the less restrictive type (e.g., NMTOKEN(S), if f_1 and f_2 are two attributes) between all the possible ones. Figure 3 illustrates how to combine the different types. Rows and columns correspond to types to be reconciled, and the entries denote the reconciled type.

Cardinality reconciliation. The mediated cardinality of two features f_1 and f_2 is defined as the less restrictive cardinality. That is, the minimum (maximum, resp.) cardinality coincides with the minimum (maximum, resp.) value associated with f_1 and f_2 .

Global features of $g_{[xc]}$ obtained by reconciling features of each X-class in $[xc]$, are considered as features characterizing the global X-class. Finally, to be exploited for query purposes, a set of *references* is maintained for each global X-class. Besides the reference to the cluster from which a global X-class is derived, each global feature has associated the corresponding set of features of the X-classes in $[xc]$ from which it has been derived.

With respect to the structure of global X-classes, we only admit sequence structures. In other words, the structures of the X-classes belonging to a given cluster are flattened. The reason for that is simplicity. We are currently studying techniques able to unify the “local” structures

```

<!ELEMENT goods (machine+,software,internet)>
<!ELEMENT machine (notebook+,desktop+,other)>
<!ELEMENT software (item+)>
<!ELEMENT internet (book+)>
<!ELEMENT notebook (name,model,spec,price,image?)>
<!ELEMENT desktop (name,model,spec,price)>
<!ELEMENT other (battery+,webcam+)>
<!ELEMENT item (name,note,price?,tohome?)>
<!ELEMENT book (title,author+,price)>
<!ELEMENT battery (name,price,warranty?)>
<!ELEMENT webcam (model,price,warranty?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT image (EMPTY)>
<!ELEMENT spec (#PCDATA)>
<!ELEMENT warranty (#PCDATA)>
<!ELEMENT note (#PCDATA)>
<!ELEMENT tohome (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!--ATTLIST tohome
xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
xlink:href CDATA #REQUIRED-->
<!--ATTLIST image
xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
xlink:href CDATA #REQUIRED-->
<!--ATTLIST machine brand CDATA #REQUIRED-->
<!--REF products {S1.products,S2.goods}>
<!--REF machine {S1.computer,S2.machine,S3.computers}>
<!--REF software {S1.software}>
<!--REF internet {S2.internet}>
<!--REF item {S1.item}>
<!--REF notebook {S1.PC,S2.notebook,S3.laptop}>
<!--REF desktop {S1.PC,S2.desktop}>
<!--REF other {S3.other}>
<!--REF book {S2.book}>
<!--REF battery {S3.battery}>
<!--REF webcam {S3.webcam}>
<!--REF item/name {S1.item/name}>
<!--REF item/note {S3.item/note}>
<!--REF item/price {S3.item/price}>
<!--REF item/tohome {S3.item/tohome}>
<!--REF notebook/model {S2.notebook/model,
S3.laptop/model}>
<!--REF notebook/name {S1.PC/name}>
<!--REF notebook/spec {S1.PC/spec}>
<!--REF notebook/image {S3.laptop/image}>
<!--REF notebook/price {S1.PC/price,S2.notebook/price,
S3.laptop/price}>
<!--REF desktop/model {S2.desktop/model}>
<!--REF desktop/name {S1.PC/name}>
<!--REF desktop/spec {S1.PC/spec}>
<!--REF desktop/price {S1.PC/price,S2.desktop/price}>
<!--REF book/title {S2.book/title}>
<!--REF book/author {S2.book/author}>
<!--REF book/price {S2.book/price}>
<!--REF battery/name {S3.battery/name}>
<!--REF battery/price {S3.battery/price}>
<!--REF battery/warranty {S3.battery/warranty}>
<!--REF webcam/model {S3.webcam/model}>
<!--REF webcam/price {S3.webcam/price}>
<!--REF webcam/warranty {S3.webcam/warranty}>
<!--REF machine/@brand {S2.machine/@label}>
<!--REF item/tohome/@href {S1.item/tohome/@href}>
<!--REF notebook/image/@href {S2.notebook/image/@href}>

```

Figure 4. XML definition of the global X-classes derived from DTDs in Figure 2

in a more sophisticated way. Figure 4 shows the XML definition of the global X-classes derived from DTDs in Figure 2. The construct REF is introduced in the XML syntax to properly represent references maintained for global X-classes. The syntax of the `<!--REF NAME {set_of_references}>` is “`<!--REF NAME {set_of_references}>`”.

5. Using the IKMS for information search in the B2B scenario

The IKMS provides a single point of access to interorganizational information within the trading community and supports different modes of searching the information of interest for performing e-business processes, such as identification of new trading partners or identification of marketplaces of similar business. For instance, in our computer selling B2B scenario, a company can be interested in ordering a bunch of notebooks and laptops, and asks selling companies of the trading community for their offers. All selling companies make their best offers for the requested order; furthermore they can exploit the IKMS to set up the supply chain to fulfill their proposed offer on time and in budget. At the same time, such companies can exploit the IKMS for finding financial partners too. Indeed, the IKMS global X-classes can be exploited for both *analysing* (browsing) and *querying* the interorganizational in-

formation of the trading community, spanning multiple company datasources. Global X-classes can be used as a basis to pose queries over the information space with the following advantages: *i)* only one query on the global X-classes of interest has to be specified instead of multiple queries on each involved datasource; *ii)* location, vocabulary, and content of each company datasource have not to be known in advance; *iii)* the integrated result of a query is automatically constructed by exploiting the sets of references to the original company datasources associated with the global X-classes. These global X-classes make possible both a *keyword-based* and a *SQL-based* query modalities.

Keyword-based queries. It is possible to locate all company datasources concerned with a particular topic by specifying the keywords of interest chosen among the names of the available global X-classes. In case of global X-classes, the terminology is strictly related to that of the DTDs, but can be properly controlled by exploiting related terms and corresponding relationships in the underlying ontology. For instance, if a company of the community wants to order a bunch of desktops and notebooks, to acquire information of available items a query like the following: Find companies selling notebook OR desktop, is formulated in form of a keyword-based query over the IKMS: "NOTEBOOK or DESKTOP".

Terms specified in this query are matched against global X-classes and addresses of company datasources corresponding to DTDs of Figure 2 are returned back to the requesting company.

SQL-based queries. To retrieve information verifying a query specified on properties of a global X-class. For instance, it is possible to pose a query like the following: Find information about notebooks "Toshiba" at a price less than \$ 4,000, by specifying the following SQL-based query on model and price properties:

```
Goods/machine/notebook/model = "Toshiba"  
AND  
Goods/machine/notebook/price < $ 4,000
```

This query returns information from the community datasources whose classes are contained in the cluster associated with the global X-classes specified in the query. A global query involving one or more global X-classes can be processed by first determining the community datasources and their DTDs for answering the query and, then, by reconstructing the integrated result out of retrieved information. The integrated result is obtained by exploiting references and *map-*

ping/conversion functions defined for global X-classes [3]. An SQL-based query is thus transformed into subqueries over involved company datasources, determined by exploiting mapping rules associated with global X-classes specified in the query. AND clauses between classes are specified in a subquery only for company datasources containing X-classes for all patterns involved in the query.

6. Concluding remarks

We have presented an approach to the construction of an Interorganizational Knowledge Mediation Scheme acting as an information infrastructure for B2B solutions for a trading community, such as integrated marketplaces and B2B portals. The work reported represents only a starting point and leaves space for further developments. Issues to be investigated are related to the: *i)* refinement and experimentation of the framework; *ii)* definition of strategies for global X-classes maintenance; *iii)* development of non-trivial ways for unifying X-classes' structures; *iv)* indexing of global X-classes for query optimization.

References

- [1] P.C. Benjamin, C.P. Menzel, R.J. Mayer, F. Fillion, M.T. Futrell, P.S. de-Vitte, and M. Lingineni. Information Integration for Concurrent Engineering (IICE). Technical report, IDEF5 Method Development Team, 1994. <http://www.idef.com/idef5.html>.
- [2] S. Castano and V. De Antonellis. A Schema Analysis and Reconciliation Tool Environment for Heterogeneous Databases. In *Proc. of IDEAS'99 Int. Database Engineering and Applications Symposium*, Montreal, Canada, August 1999.
- [3] S. Castano, V. De Antonellis, and S. De Capitani di Vimercati. Global Viewing of Heterogeneous Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2), March/April 2001.
- [4] S. DeRose, E. Maler, D. Orchard, and B. Trafford. *XML Linking Language (XLink)*. World Wide Web Consortium (W3C), July 2000. <http://www.w3.org/TR/xlink>.
- [5] R.J. Glushko, J.M. Tenenbaum, and B. Meltzer. An XML Framework for Agent-Based E-commerce. *Communications of the ACM*, 42(3):106–, March 1999.
- [6] W. Han L. Liu, C. Pu. An XML-Enabled Data Extraction Tool for Web Sources. *Information Systems - Special Issue on Data Extraction, Cleaning, and Reconciliation*, 2001. to appear.
- [7] A. Levy, A. Rajaraman, and J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proc. of the 22nd VLDB Conference*, 1996.
- [8] G. Mecca, P. Merialdo, and P. Atzeni. Araneus in the Era of XML. *IEEE Data Engineering Bulletin, Special Issue on XML*, 22(3):19–26, September 1999.
- [9] A.G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, November 1995.

- [10] G. Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassolos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems*, 8(2):117–132, March 1997.
- [11] C. Parent and S. Spaccapietra. Database Integration : an Overview of Issues and Approaches. *Communications of the ACM*, 41(5):166–178, May 1998.
- [12] E. Rahm and P.A. Bernstein. On Matching Schemas Automatically. Technical Report MSR-TR-2001-17, Microsoft Research, February 2001. http://research.microsoft.com/scripts/pubs/view.asp?TR_ID=MSR-TR-2001-17.
- [13] Y.K. Ng S.J. Lim. A Heuristic Approach for Converting html Documents to XML Documents. *Computational Logic*, pages 1182–1196, 2000.
- [14] V.S. Subrahmanian, S. Adali, A. Brink, R. Emery, J.J Lu, A. Rajput, T.J. Rogers, R. Ross, and C. Ward. HERMES: Heterogeneous Reasoning and Mediator System. <http://www.cs.umd.edu/projects/publications/abstract/hermes.html>.
- [15] The ebXML Consortium. UN/CEFACT, OASIS. <http://www.ebxml.org>.
- [16] The RosettaNet Consortium. <http://www.rosettanet.org>.
- [17] World Wide Web Consortium. *Extensible Markup Language (XML) Version 1.0*, February 1998. <http://www.w3.org/TR/REC-xml>.
- [18] J. Yand and M.P. Papazoglou. Interoperation Support for Electronic Business. *Communications of the ACM*, 43(6):39–47, June 2000.