Semantic technologies and e-business

Ivan Bedini, Georges Gardarin, Benjamin Nguyen

Semantic Technologies and e-business

Ivan Bedini
Orange Labs
France

Georges Gardarin
University of Versailles
France

Benjamin Nguyen
University of Versailles
France

ABSTRACT

In this chapter, we study what semantic technologies can bring to the e-business domain and how they can be applied to it. After an overview of the goals to be achieved by e-business applications we detail a large panel of existing e-business standards, with a specific focus on B2B (Business to Business) and their current modus operandi. Furthermore we also present some of the most relevant e-business ontologies. We then argue that the use of semantic technologies will simplify the automatic management of many e-business partnerships. However the construction of ontologies brings a new level of complexity that might be facilitated by automating the great part of the generation process. For this we have developed the Janus system, which is a prototype to help with the automatic derivation of ontologies from XML Schemas, the de-facto format adopted in e-business standard applications. Differently from existing systems it permits to retrieve automatically conceptual knowledge from large XML corpus sources and is based on the use of the Semantic Data Model for Ontology (SDMO) whose advantages are presented in this chapter.

INTRODUCTION

Computer mediated networks play a central role in the evolution of Information Systems. For example the sales application must interface with the inventory application or the inventory application must connect to the supplier's application, or the simple mobile calendar must synchronize with the professional calendar; all the time, applications require efficient and effortless integration with others. Nevertheless the integration of enterprises applications still remains harder than it really should be. Enterprises are typically composed of several applications that are custom built, acquired from third parties or a combination of both. Moreover it is not uncommon to find an enterprise whose information is segmented between different instances of enterprise software and countless departmental solutions. In consequence, the integration of these application systems becomes a real challenge that requires considerable human effort, especially if the final goal is to connect applications belonging to different enterprises. This last use case refers to what is also called Business to Business (or simply B2B).

Communication between applications is mainly governed by standard protocols and standardized content, as shown in the European e-business report (E-Business W@tch, 2007) among different solutions applicable to e-business, at least three enterprises out of four that implement business exchanges with partners, declare implementing applications standards solutions based on these two technologies (in Europe). The advent of XML along with Web

Services, and more generically with the Service Oriented Architecture (SOA), has contributed greatly to the development of such standards-based integration solutions. But the large adoption of these technologies entails a new fragmentation in applications development. As a result standardisation addresses only parts of the integration challenge. The frequent claim that XML is the lingua franca for system integration is somewhat misleading; indeed this statement does not imply common semantics and its adoption has led to the creation of countless dialects and languages which cannot be understood and integrated directly by machines. This problem is reflected in the many existing B2B standards that we present in this Chapter. The analysis we provide is based on the observation of more than 40 of them.

Following this approach, professional exchange integration scenarios are based on a complete transformation of business messages at design time. Although this model works and businesses are able to exchange messages electronically, the effort to produce these standards appears too high. Moreover, it would be impossible to write a standard specification for every possible business communication. Especially for (smaller) firms who are unable to contribute to standardization. For this reason Semantic Web-related technologies are well suited to integrate the e-business architecture in order to fulfil the standardization approach and achieve the needed flexibility.

Another aspect that we tackle in the Chapter is the automatic construction of top-level domain ontologies. As asserted by Euzenat and Shvaiko (Euzenat *et al.*, 2007), the importance of the generation of such kind of knowledge is fundamental for the improvement of the alignment and thus integration problem. However most solutions implicitly assume that a reference knowledge exists in compatible format and semantics, but actually it is often inadequate for the application domain or difficult to find, if it even exists at all.

To give a point of comparison, we also present the most adopted approach to e-business data integration. Through this analysis we point out the current architecture limitations and explain why ontologies are a better approach which leads to a gain in flexibility and dynamicity. In this sense we provide an overview of schema matching and ontology alignment solutions and we point out one of the current limitations to their broad adoption and provide a system that facilitates, by automation, the transformation from the current model to the "next one": from XML to OWL.

The overall outline of the Chapter is as follows: the first section introduces current e-business approaches to data integration and we follow with the presentation of more than 40 existing standards for the B2B and B2C domains. Following this introduction we focus on Semantic Web related technologies applied to the e-business domain. In the survey we detail some of the most relevant works related to product classification and we continue with a section focusing on schema matching and ontology alignment solutions. The last section provides the description of a system we have implemented to fulfil some of the current shortcomings. We conclude with what we think to be the most important issues to be developed and provide some directions to follow.

E-BUSINESS SEMANTICS DESIGN

Three main patterns to achieve messages exchanges

To understand how the integration of messages in e-business exchanges works let us consider a common transaction among a buyer and its supplier. Figure 1 shows the two parties with an

internal interface used by their "domestic" applications. These interfaces reflect exactly internal data requirements at semantic and structural level and applications are designed or adapted using these interfaces. As we argue below most businesses already use a different format, most often a standard based solution, for their external connections, that we call *external interface*. This interface organizes the internal data necessary to the exchange and produces a first conversion handled by each party to reflect their own application data input/output. If these first conversions do not correspond exactly, another conversion is required, this time defined accordingly by both parties.

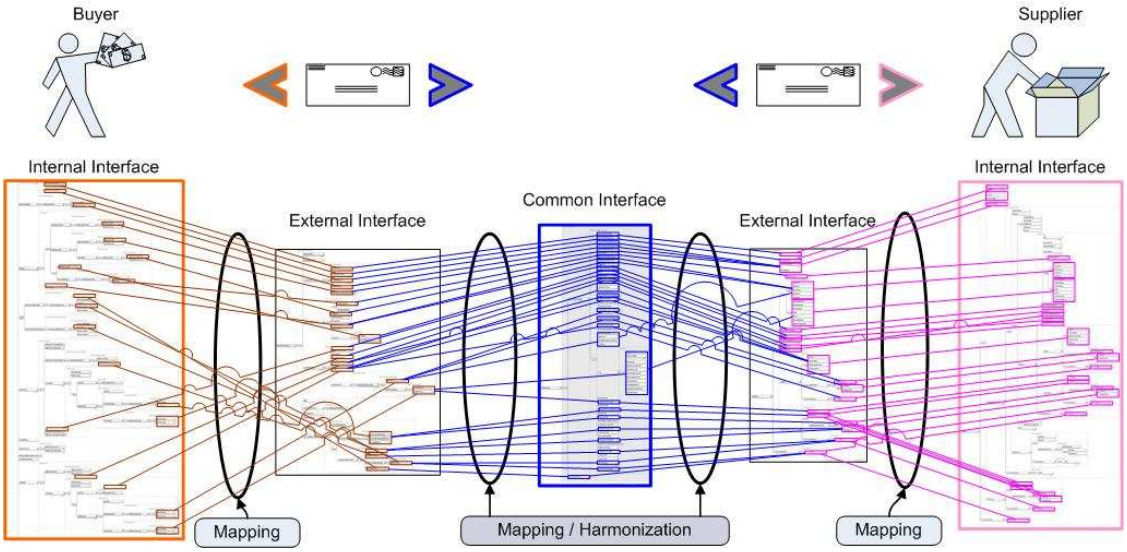

Figure 1 – Representation of message transformation scenario

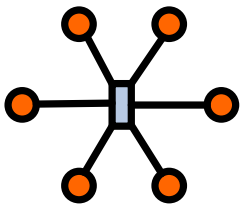

Figure 2 – Message content definition adopting standards

We define this approach to e-business exchanges as the *adoption of standards pattern* (mutualisation). Here business requirements are provided by a collegial work defined in a specific consortium. The realization is a common preliminary effort that involves several parties, mainly experts of the specific process and/or the whole domain. It has the advantage of being a standard and thus of guaranteeing a certain level of compatibility, durability and reuse of past experiences and knowledge. The resulting definition of business data is a static knowledge representation that can be changed only with further common effort. Negative points are that it requires a tremendous standardization effort and quite often several standards coexist for the same requirements. Figure 3 illustrates how this business exchange pattern centralises efforts and makes this approach more profitable with respect to others, but only in a theoretical perspective because it can become complex when more standards come into the arena.

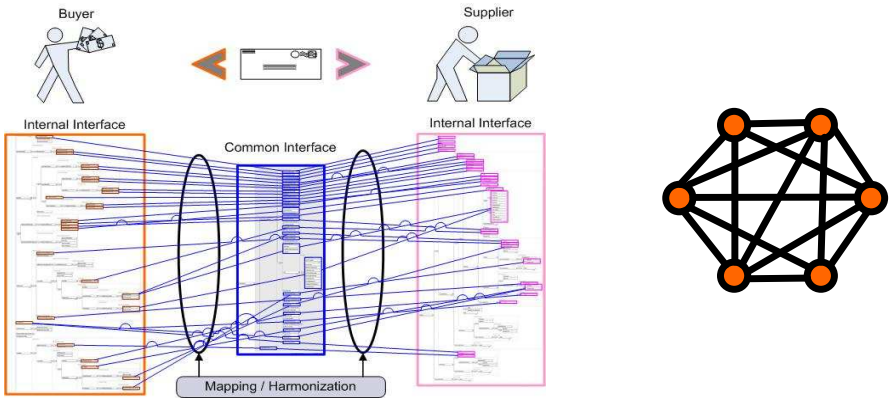

Figure 3 – Message content definition in ad hoc solution

Alternatively consider the *ad-hoc* or *point-to-point approach*, where external interfaces and the corresponding mappings are defined multilaterally during the design time phase of the collaboration in order to respect the information to exchange. This system shows some kind of "flexibility", in the sense that it does not present specific constraints: a new design is made every time. This flexibility on the other hand clearly shows a low degree of reusability and integration with new partners. The left hand side of Figure 3 shows the mapping between interfaces of two companies, while the right hand side of the picture highlights what happens when a company has more business relationships to set up. Interfaces defined by this approach are rarely compliant among different connections. Therefore the number of conversion needed to have a fully meshed point-to-point connections between *n* companies is *n(n-1)*. i.e. for 10 applications to be fully integrated point-to-point, 90 conversions could be necessary.

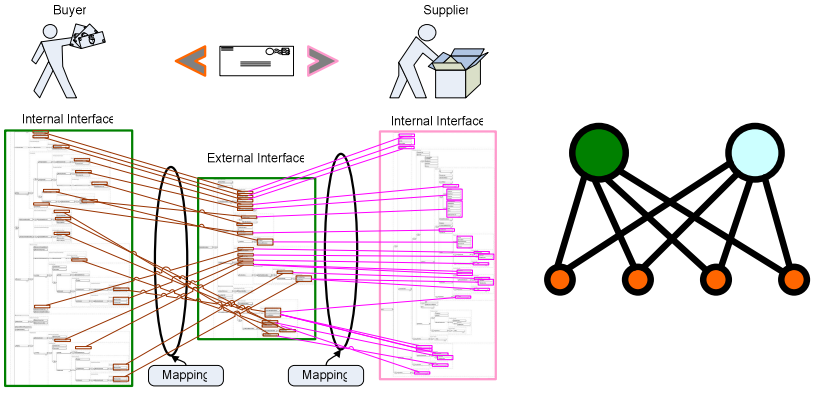

Figure 4 – Message content definition according a proprietary solution

Another pattern is the *proprietary data model*; in this case external interfaces are decided unilaterally. Typically this approach covers business collaborations with a main contractor in cooperation with small businesses, such as a big retail group and its suppliers. In this case it is simpler for the big company to take entire charge of the business requirements design, trying to adopt the larger predictable requirement, because it often has the more complex system to manage and to make interoperable with internal processes, while a little company uses a smaller

information system. Setting up such a solution is faster and does not require the complex harmonization phase, but on the other hand partners who do not adopt the same solution are forced to develop a new application layer to join the business collaboration. Figure 4 depicts this business collaboration pattern and draws attention to the fact that there is a party that is forced to produce mappings and application layers for each new collaboration.

e-business Standards

Enterprises do not currently publish their interfaces formally in public repositories, which made it difficult to produce an explicit base of reusable documents. However as shown in the European e-business report (E-Business W@tch, 2007) at least three enterprises out of four that implement business exchanges with partners, declare implementing applications based on e-business standards solutions (in Europe). Another conclusion drawn by this report is that the difficulty with e-business and e-government development is that they mainly work vertically by producing connexions among enterprises belonging to the same business area. Indeed while interoperability within industries, such as the financial industry, is intended to enable efficient e-business (with *The Single Euro Payments Area* – SEPA as an example), interoperability between *all* industry sectors for e-business, i.e. between financial institutions and their clients from other industries, is not optimal. Corporations' expectations and financial institutions' demand for value-added services will, however, continue to rise. This means that the interfaces between them are becoming increasingly important. These interfaces have not yet been implemented in their final form, and most of them have not even been defined in detail yet (in terms of standards). Here developments in standardization can take place to reduce interoperability problems and to benefit from world wide experiences, but it is hopeless to standardize any possible business collaboration. Moreover the problem of finding, reusing, harmonizing and adapting the different standard components is not trivial: until now it has been common practice, including among standardization organizations, to simply publish business data on a Web page using directories or even flat files!

Table 1 presents a list of 37 e-business standards, mainly targeting the B2B area. The data provided by this set of standards is a considerable corpus that gives us a broad view about current practices. The table lists: the name of the standard body or consortium; column two lists the business areas that the standard covers; the alliances column informs about declared compatibility coalitions, already active or expected to come; the fourth column summarizes what kind of business content is produced by each standard body; the following column details the formalization of published standards; the standards' downloads column provides the information of their availability and adoption (public, under a payment, or only for member of the consortium); the last column just provides a link. The table does not say if the consortium also provides a specific implementation framework.

We have not inserted in this list the standard bodies that have been *a priori* excluded because they are designed for too specific use case. Examples of the overly specific working groups are: EDItEUR (the international group for electronic commerce in the book and serials sectors), BISG (Book Industry Study Group) and EPISTLE (the European Process Industries STEP Technical Liaison Executive), PRODML (Production Mark-up Language and WITSML (Wellsite Information Transfer Standard Mark-up Language).

As we can see, a lot of business data is defined by standard bodies: a dictionary of core components, whole messages, business processes, Web Service descriptions, code lists and EDIFACT messages. In this chapter, only core components, often called *Data Dictionary*, and

messages have had our attention and were analysed in detail. Our study shows that XML Schema is the most widely supported formalism adopted by consortiums and at present it is the *de-facto* standard document format. It has overtaken other formats like the "old" EDIFACT and, at least for the moment, the "new" RDF/OWL format. cXML[i] is the only standard to provides simply a DTD, and *not a single* RDF/OWL format is officially produced by any consortium.

A growing number of standard bodies are currently adopting the ebXML (e-business XML) design as basis for their own standards and are aligning their business components to the Core Components Library (CCL). Among them we can cite: OASIS Universal Business Language (UBL), Open Applications Group (OAG), EAN-UCC, SWIFT, ANSI ASC X12 and CIDX.

ebXML is a joint effort of OASIS and UN/CEFACT that aims to develop a complete framework for e-business. The library is prevalently developed by the UN/CEFACT standard body that counts 15 specific working groups, each one representing a business area such as Supply Chain, Transport Domain, Customs, Finance, Construction, Insurance, Healthcare, Agriculture and e-Gov. Another specialised group provides a synchronization of the documentation and specifications proposed by each group. It finalizes the work with a harmonized library of the so called CCL, which are the basic components to build B2B messages. Others groups also define standard business processes and technical implementations. The CCL is drawn on the UN/CEFACT Core Component Technical Specification (UN/CEFACT TMG, 2003) that provides a simple and powerful UML based data model, to define reusable structure and semantic content of business messages.

Concerning data presentation, almost all organizations provide a package containing several documents. It includes specifications, graphics, examples, guidelines, implementation tutorials and XSD files. Generally XSD files are numerous, at least one for each specific business message, one for grouping common core components, others for grouping common data type definitions and code lists. Only few of them provide a specific repository with a detailed view and discovery system of data components.

B2B Standards' Semantics

In order to understand if XML Schema standards can be processed by semantic engines we have developed an automaton that extracts all XSD tags and retrieves the words from them. The automaton uses WordNet (Miller, 1995) to verify that tags are compound words that can be converted to real words. Once processed, our corpus source is composed of a collection of 26 B2B standards, composed of over 3000 XSD files with more than 170.000 named tags. We feel that this is largely enough in order to have significant information about B2B business message description practices and semantics. Our results depicted in Figure 6 show that 71% of tags are composed of words recognized by the dictionary, 14% contain abbreviations that can be related to dictionary words, and only 15% of total tags contain unknown words. From the pie-chart we observe that Mismo is the more prolific standard body, a few others provide between 5 and 10 % each and around 30 % is shared between the remaining standards. Finally we found that the whole set of tags is built with only ~3300 different words, that we call the *e-business vocabulary*. Moreover we have observed that at semantic level, past a given point, adding more standards into the process does not change much. This is proven by the experiment we conducted and results shown in Figure 5. We can see that for both pictures, the line indicating the percentage of words added by each standard is high only during the first few iterations; afterwards we have only about 5% of the extracted words that are added to the vocabulary.

We conclude that this corpus can be considered as a basis for a deeper semantic approach in order to generate the domain ontology. In sections below we provide reasons for using a semantic approach for the e-business domain and we continue with a contribution to the automation of the generation of an ontology from XML Schemas.

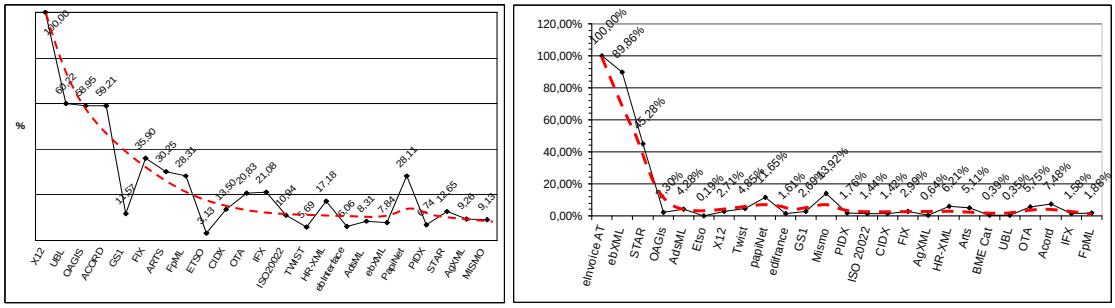

Figure 5 – e-business vocabulary generation

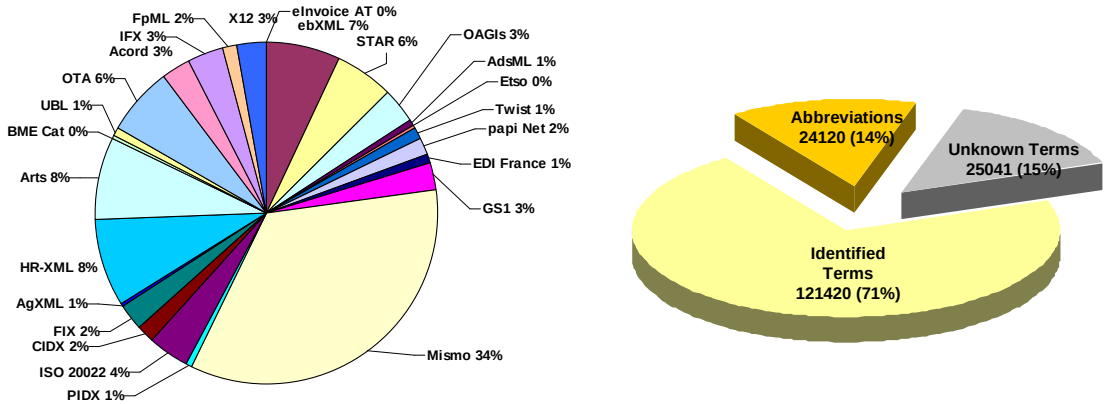

Figure 6 – Standard XML Schemas extraction figures

	Standard Body		Business Area	Alliances	What	Published Formats	Standards' Downloads	Web Site
1	ACORD	Association for Cooperative Operations Research and Development	Insurance, reinsurance and related financial service	ASC-X12, XBRL, HR-XML, eEG7, CSIO	Dictionary, messages	EDIFACT, XML Schema, WSDL	registration	www.acord.org
2	AdsML	Advertising Standards	Advertising, Graphics communication		Dictionary, messages	XML Schema	free	www.adsml.org
3	AgXML	Agricolture XML	Agriculture supply chain	ebXML, CIDX, RAPID	Dictionary, messages	XML Schema	membership fees	www.agxml.org
4	AIAG	Automotive Industry Action Group	Automotive industry				membership fees	www.aiag.org
5	ARTS	Association for Retail Technology Standards	Retail		Dictionary, Relational Data Model	XML Schema	payment (exept for schemas)	www.nrf-arts.org
6	ASC X12	The Accredited Standards Committee	Cross industry		Dictionary, messages, EDIfact messages, BP	EDI X12, XML Schema	registration	www.x12.org/
7	BMECat	Federal Association for Material Management, Purchasing and Logistics	Electronic		Dictionary, Classification schemas, Product Configuration, price formulas	XML Schema and DTD	registration	www.bmecat.org
8	ChemITC	American Chemistry Council's Chemical Information Technology Center	Chemical					www.americanchemistry.com/s_chemITC/
9	CIDX	Chemical Industry Data Exchange	Chemical	ebXML, RAPID, OAGi, ChemITC	Dictionary, Business Processes, WSDL, RFID codes, messages	XML Schema	free	www.cidx.org
10	CSIO	Centre for Studies in Insurance Operations	Insurance, reinsurance and related financial service					www.csio.com/
11	ebInterface		Invoice		Invoice Document	XML Schema	free	www.ebinterface.at/
12	EbIX	European forum for energy Business Information eXchange	Energy				free	www.ebix.org
13	ebXML	e-business XML	Multi area. 15 business area represented. One WG with harmonisation purposes and one for BP definition	ISO	Dictionary, Messages, code lists, EDIFACT, methodologies	XML Schema and UML, EDIFACT, Spreadsheet	free	www.unece.org/cefact/
14	eEg7	E-business Standards for the European Insurance Industry	Insurance, reinsurance and related financial service					www.eeg7.org/

15	Energistics		Energy		Dictionary		registration	www.energistics.org
16	ETSO	European Transmission System Operators	Specific electric transaction	ebXML	Dictionary	XML Schema	free	www.etso-net.org
17	FIX	Financial Information eXchange	Banks, broker-dealers, exchanges and institutional investors	SWIFT (ISO 20022), FpML	Framework with message protocol, message definition, codes and Dictionary	XML Schema	registration	fixprotocol.org
18	FpML	Financial Product Markup Language	Financial	FIX, FIXML	Dictionary, Business Processes, architecture	XML Based	registration	www.fpml.org/
19	GS1	Global Standards	Supply chain for Healthcare, Defence, Transport & Logistics	ebXML	Dictionary, Business Processes, Messages, SOAP Messages…	XML Based	free	www.gs1.org/
20	HL7	Health Level 7	Health				free	www.hl7.org
21	HR-XML	Human Resources XML	Human Resource	ACORD	Dictionary	XML Schema	free	www.hr-xml.org
22	IFX	Interactive Financial eXchange (IFX) Forum	Financial		Dictionary, Messages, Web Services	XML Schema, WSDL	registration	www.ifxforum.org/
23	ISO 20022	ISO 20022 Universal financial industry message scheme	Financial	IFX, OAGi, TWIST	Dictionary	XML Schema, UML	payment	www.iso20022.org/
24	MDDL	Market Data Definition Language	Financial		Specific XML framework		registration	www.mddl.org/
25	MISMO	Mortgage Industry Standards Maintenance Organization	Residential, commercial, eMortgage	IFX, ACORD, ASC X12	Dictionary	XML Schema	free	www.mismo.org
26	NAESB	North American Energy Standards Board	Energy (Gas, electric)				membership fees	www.naesb.org/
27	OAGi	Open Application Group integration Standard	Cross industry	ebXML	Dictionary, Web Services, Messages	XML Schema, WSDL	registration	oagi.org
28	Odette		Automotive industry				membership fees	www.odette.org
29	OTA	Open Travel Alliance	Turist		Dictionary, codes, messages	XML, Spreadsheet	registration	www.opentravel.org/
30	PapiNet	Paper Industry Network	Paper Industry		Dictionary, messages	XML Schema	free	www.papinet.org/
31	PIDX	Petroleum Industry Data Exchange	Energy (petroleum industry)	ebXML	Dictionary, Web Services, Bar codes, EDI messages, Business Process	XML, WSDL, EDIFACT	free	www.pidx.org
32	RAPID		Agricolture	CIDX	Dictionary, Messages, Code lists, Bar codes	XML Schema, EDIFACT	free	www.rapidnet.org/
33	RosettaNet		Supply Chain Management, IT, Telecommunication	GS1 US, ebXML	Dictionary, Business Processes	DTD, EDIFACT, XML Schema	registration	www.rosettanet.org

34	STAR	Standards for Technology in Automotive Retail	Automotive industry	OAGi, ebXML	Dictionary, messages, Web Services	XML Schema, UML, WSDL	free	www.starstandard.org
35	TWIST	Transaction Workflow Innovation Standards Team	Supply chain, payment	FpML, FIX, SWIFT	Dictionary, Business Process	XML Schema	free	www.twiststandards.org/
36	UBL	Universal Business Language	Invoicing, ordering	ebXML	Dictionary, messages, Business Processes	XML Schema, UML, ebBP	free	www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl
37	XBRL	eXtensible Business Reporting Language	Reporting, accounting	UN/CEFACT, CIDX	Dictionary, messages, formulas	XML	free	www.xbrl.org/

Table 1 – B2B Standards

WHY CREATE E-BUSINESS ONTOLOGIES?

Current methods of business collaborations and relative architectures exhibit a common characteristic of business data design: *they are always pre-formatted to strict and precise structures and semantics*. These methods have the advantage of allowing error-safe execution management but to the cost of a strong initial effort. We define this approach as the ***deterministic method***, although no module exists yet to resolve ambiguous situations due to similar, though different design. Since the Semantic Web Vision (Berners-Lee *et al.*, 2001) which is all about machines being able to locate and process information on the World Wide Web without the need for human intervention, the next step to transform a deterministic method to a more dynamic and automated method, should be the adoption of semantic related technologies. However it is known that adding new tools adds new complexities and new learning curves, so there needs to be a concrete business benefit to justify the cost of implementation. Throughout this section we argue why ontologies should be introduced to the e-business domain.

Firstly we observe that e-business provides an interesting use case for semantic applications because by its nature it illustrates the problem of different designs and ways of structuring the same set of concepts producing data heterogeneity problems. The deterministic approach prevents any possible automation of data interpretation because machines are only called to execute code and no data description is available for handling reasoning and inferences at run time, even for simple mismatches. This is the consequence of an approach completely designed for human understanding. Reasoning on this kind of data is impossible because of the intrinsic limits of its definition.

How can we combine dissimilarities of semantics, information details, structure and also cultural approaches in a comprehensive model? How can machines communicate between themselves reducing human effort?

As we already mentioned the Semantic Web, and particularly ontologies, seem to achieve good results within the last years. Several people have addressed the specific adoption of such technologies for the e-business domain. Dieter Fensel in his book, *Ontologies: Silver bullet for knowledge management and electronic commerce* (Fensel, 2001b), outlines the key differences between ontologies and databases schemas which are more close to a “physical data model”. Moreover he argues that the language for defining ontologies is syntactically and semantically richer, by its own nature the ontology requires a consensus among several parties and as such it is more similar to a domain theory rather than a data container.

The document *Best Practices and Guidelines* (Leger, 2002) focuses on applications of Semantic Web for electronic commerce on the Internet, and defines a specific list of potential benefits from its adoption. For instance, it details the development of efficient and profitable Internet solutions, a meaningfully share of information, that provide a good basis to argue the benefit of the integration of semantic technologies. At the same time, the authors identify critical issues and research priorities to transform these potentials into real benefits.

In the paper *Potential Advantages of Semantic Web for Internet Commerce*, (Zhao, 2003) the author provides a comprehensive list of twelve points on the potential benefits of adopting Semantic Web in the domain. Among these twelve categories let us stress the possible improvement in the integration of applications, information management, filtering of information, the composition of complex systems, a more flexible standard vocabulary, and *serendipity* (unexpected benefits).

Antony B. Coates in his talk (Coates, 2007) is more pessimistic and argues that the Semantic Web vision still remains a long term goal, and this is the reason why businesses and standard

bodies still hesitate to introduce it. However he adds some factual reasons linked to the limitations of current data models and how ontologies can already improve them in the short term. For instance UML (Unified Modelling Language) is the most widely used modelling technique in the domain. Indeed UML is intended as a general modelling approach because it does not only propose data modelling, but also use cases, process flows, state diagrams and also has an XML interchange format (XMI). However the interchange format has numerous versions and different tools either use different versions, or use the same version in different ways (too much flexibility in the format?). In consequence, interoperability is in fact rather difficult. Another relevant limitation of UML is that for object-oriented reasons in some cases it requires adding extra classes, which is fine for technical users but it is irrelevant and unnecessary in a model designed to be used by business experts. This makes diagrams more complex and confusing than they need to be. Take as an example, illustrated in Figure 7, an intended business model like "vendor sells to company or government", where UML forces the creation of common "purchaser" parent class. OWL adds simplicity, when representing the same model, and allows us to say that a Vendor sells to a "Company or Government", without introducing a named parent class

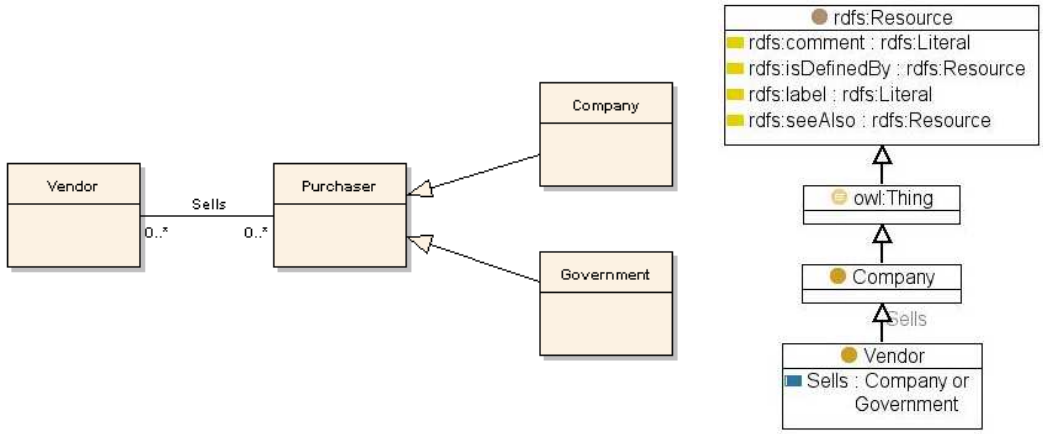

Figure 7 – Example of UML class diagram and correspondent OWL modelization

Also the UML tools' support for objects/instances (e.g. "a particular car, a particular person") is much weaker than RDF/OWL tools, and not really usable for constructing business context models referencing particular countries, business areas, etc. Moreover when merging models, RDF/OWL assertions are preserved and also enable detection of inconsistencies, while the UML merging operation is completely a human task.

In (Anicic, 2005) the author defines an architecture based on Semantic Web technologies to investigate enterprise application integration (EAI). As an example both enterprise applications implement two correlated but independent standards for messages exchanges. One is Standards in Automotive Retail (STAR) and the second is the Automotive Industry Action Group (AIAG) and both base their interface on a more "horizontal" standard defined by the Open Application Group (OAG). Their study shows that ontologies and reasoners improve the integration of message exchanges between companies. Conversely, in their implementation the integration still requires human intervention, since identification and resolution of semantic and syntactic similarities, is done by hand.

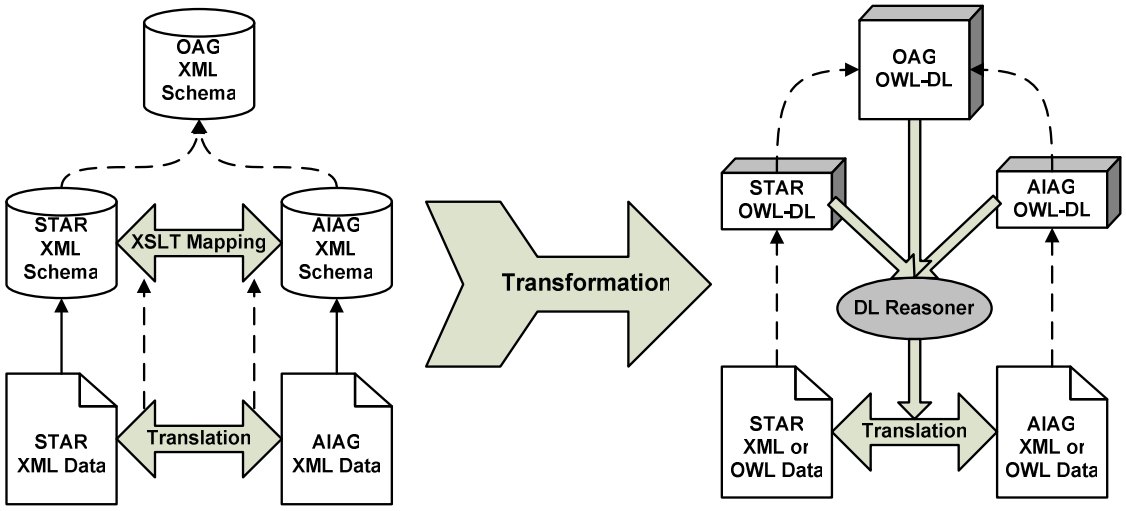

Figure 8 – Traditional and Semantic Web-based EAI Standards Architectures

This experience and similarly the architecture presented in the B2Boom work (Kajan, 2009), show how the semantic mediator improves interoperability problems between worldwide enterprise applications. However the problem is still strongly related to the ontology matching/alignment problem, and the need for a specific domain ontology which becomes the new core question.

The Canonical Data Model

The book *Enterprise Integration Patterns* by Gregor Hohpe (Hohpe, 2003) clearly formalizes problems with application integration. He provides an exhaustive list composed of 65 enterprise integration patterns to be considered when building a system able to manage the whole process of electronic business exchange. His approach is based on a messaging system. Focusing on those patterns for data integration, Hohpe suggests different approaches to resolve the problem. One is to share the same basis of data like using a shared database or adopting the same base of documents between applications, but these patterns can be at most adopted within a single company. A second approach is to build a messaging system that translates business documents, called *message translator*, which is similar to the *point-to-point* approach presented above. Yet in the same approach a complementary pattern suggests using a *message mapper* which tries to conceptualize messages as business objects and thus more independent of application data. By doing so, he adds a pattern including a *Canonical Data Model* in order to minimize dependencies from different data formats. In this approach the Canonical Data Model provides an additional level of indirection between applications' individual format, similar to a pivotal format, like a "lingua franca" for information systems. This approach is somewhat a mix of the **proprietary approach** with the **adoption of standard** approach seen above. In fact this approach is used by many industry specific consortia (like PIDX for the petroleum industry, or XBIT for the book industry) that produce a formal model specific to their use that must be adopted by all collaborating partners.

In our approach we suggest adopting an ontology when building the specific B2B messages canonical data model. More than a pivotal format, we want to construct *reference background knowledge* to improve application integration on the basis of a *message mapper* pattern. This

approach is quite different from other experiences in the e-business domain, such as those provided by Corcho *et al.* (Corcho, 2001) and by Hepp (Hepp, 2006), because it targets message definition rather than a thesaurus like the eCl@ss ontology, since a message is not a well defined hierarchical set of products. This means that matching messages is a more complex operation because each message meets a specific action, which is not always the same for different standards. In other words, in a heterogeneous environment we are not able to say beforehand if the sending application has messages that correspond exactly to the receiver application messages, in a one-to-one association, but we can make the hypothesis that the sender application manages some "concepts" that are similar to those of the receiver application. In this context we consider a new pattern based on a canonical data model developed as ontology that aims to correlate these messages with common concepts. A procedure that performs such pattern is shown in Figure 9 and is as follows: 1) detect what concepts the message conveys; 2) match them with the canonical model; 3) find corresponding concepts in the target application data model; 4) chose the messages that fit the requirement best and finally; 5) translate.

However one main problem here is the Canonical Data Model generation, which corresponds to the development of a domain ontology, or at least a reference ontology common to the whole B2B domain. The difficulty is that the classical development of this ontology is typically entirely based on strong human participation, which is a long task, really similar to the realization of a big standard and delves into a static knowledge representation. In the B2B context, where business partners can join a collaboration on the fly, the Canonical Data Model should be able to integrate new knowledge on the fly as well. In the following section we trace the requirements that such knowledge representation should have to fit into the B2B domain well and complete its assigned tasks in the pattern defined above.

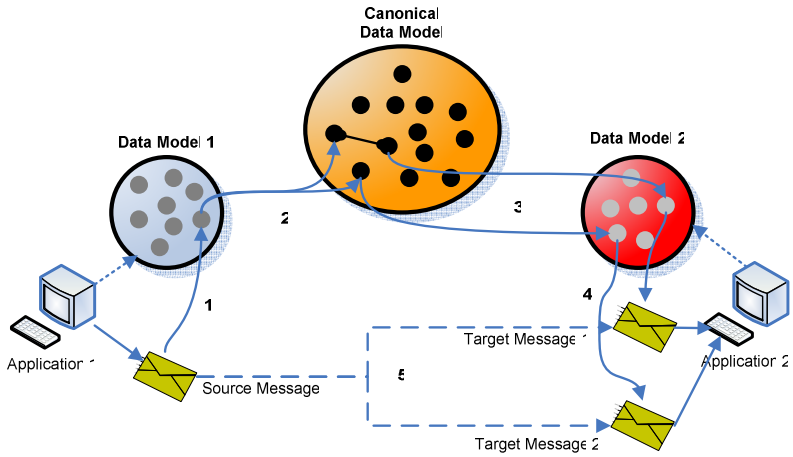

Figure 9 – Messages translation procedure

Ontology Requirements

There are some general features that have to be respected when building an ontology, independently of the application domain. For example Barry Smith in his paper (Smith, 2006) examines the ISO 15926 upper ontology (Batres, 2005) and furnishes a series of principles to follow when developing a reference ontology, of which we can mention: the principles of **intelligibility**; **openness**; **simplicity** and **re-use** of available resources; **coherence**; **compositional**, if two concepts are used to express a third concept, the formers must be included

into the ontology; **singular** nouns, the terms of an ontology should be formulated in the singular. In his analysis he concludes that ISO 15926 is not an ontology because it does not follow any of these principles and the result is just a coding scheme rather than an ontology.

In a general way we can summarize that ontologies glue together three important requirements to consider when developing one:

- Ontologies aim at consensual knowledge, their development requires a cooperative process and normally, for pragmatics reasons (e.g. limiting complexity and dimension) they are restrained to a specific domain or application.
- Ontologies formalize semantics for information, consequently allowing information processing by a computer.
- Ontologies implicitly use real-world semantics, which make it possible to link machine tractable content with meaning for humans.

We next detail some requirements that we have added specifically for the B2B use case, but they can fit other use cases as well.

Firstly the concept of **dynamicity** of an ontology for the e-business domain has been already introduced (Fensel, 2001b) which states that "Ontologies must have a network architecture and Ontologies must be dynamic". Also (Hepp, 2008b) sustains that otology must be able to grow dynamically without "bustling" existing applications. From the NeOn project we also find the concept of **networked ontologies** (Tran, 2007 and D'Aquin, 2008) where ontologies can be distributed in a dynamic environment, like a peer to peer network, and applied to an e-business integration use case. At the same time computational time for discovering the best matches between several ontologies is expensive, therefore the techniques applied to match elements should maintain previous discovered alignments and common uses in order to quickly recognize similarities between concepts and to compute only new information. We capture these characteristics in the *dynamism* attribute for a domain ontology. In reality an ontology is a static knowledge representation. In current literature the ontology dynamic is strictly associated to ontology evolution/versioning and has been investigated in several papers, like Noy *et al.* (Noy, 2004) which traces all possible changes that can take place in ontologies. However when dealing with dynamic ontologies we closely refer to the generation process of the ontology and with its capacity to introduce new knowledge interactively. To this end, the process should follow an iterative approach, i.e., conceptual knowledge may be integrated in turn. One condition that the ontology must respect in this case is the **completeness** criterion, which means that all matched concepts must be represented in the ontology, even after a merging operation, and in the simpler case where a concept has no conflict with other concepts it is simply added to the ontology. Consequently an ontology is a dynamic characteristic of the domain, thus evolution should not be equivalent to a classical versioning system, but more to a learning system, including a merge operation without loss of information and backward compatibility. We call this feature the *dynamism* of an ontology.

On top of these requirements, we want to be able to generate and enrich the domain ontology as automatically as possible. Indeed, even in a specific field, the concepts handled by the applications can be numerous and the quantity of information which we wish to maintain for each concept is vast. Solely relying on human management could quickly become impossible: recall that our example corpus size is thousands of XSD files and all the more concepts.

E-BUSINESS ONTOLOGIES

In this section we present some of the most representative works on e-business ontologies. We focus on development efforts to produce either upper or domain ontology. Where we recall that an upper ontology has the purpose to be a reference knowledge base for the whole domain and thus be useful to induce mappings among concepts of two or more application ontologies, as described by Guarino (Guarino, 1998). Moreover, as already mentioned above, we distinguish two kinds of ontologies for the e-business domain: the first one is more related to e-commerce applications and product description and categorization; while the second is closer to B2B applications, where messages and semantics are more difficult to categorize in a sole representation, as the multiple standards presented in Table 1.

Semantic Web for e-commerce

In the past years several research works have studied the integration of Semantic Web and e-commerce applications. The interest of this kind of semantic improvement for businesses is still under-estimated. Indeed the generation of semantically annotated documents can greatly increase the visibility of commercial products when searching on the Web. Traditional Search Engine Optimization (SEO) tries to put on top of all search results a Web page that matches a keyword best, but quite clearly, that can work only for one company. *Well semantically annotated* document put businesses on top of Web visibility for people who are looking for more precise products or services independently from the Web page itself. If data integration, thus applications capable of exchanging information automatically, still requires a lot of effort and new elements before achieving concrete adoption, the generation of linkable data on the Web requires a lower investment with a probable earlier return of benefits.

To this end, the Web Ontology for e-commerce produced by Hepp (Hepp, 2008) provides a complete framework to produce annotated Web pages in a simple manner. It is a good starting point for businesses that are seeking an early semantic adoption. The framework is based on the ontology derived from eClass and UNSPSC, namely eClassOWL (Hepp, 2008c) and the similar ontology unspscOWL, which is awaiting copyright clearance. The so called *GoodRelations* framework includes a language that can be used to describe business offers very precisely. It can be used to create a small data package that describes products and their features and prices, stores and opening hours, payment options and the like. The framework is also supported by: tools for creating directly GoodRelations annotated data; plug-ins/Extensions for e-commerce software; a tool that spots semantic inconsistencies in GoodRelations data beyond the axioms of the ontology. The result is easy to use: all it takes is to paste the data package into the Web page using W3C's RDFa format, as shown in Listing 1.

```
<!-- BEGIN: RDFa Meta-data for machines -->
<div xmlns="http://www.w3.org/1999/xhtml" xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:eco="http://www.ebusiness-unibw.org/ontologies/eclass/5.1.4/#"
xmlns:gr="http://purl.org/goodrelations/v1#"
xmlns:owl="http://www.w3.org/2002/07/owl#" class="rdf2rdfa">
   <div class="description" about="http://www.oettl.it/" typeof="owl:Ontology">
   <div rel="owl:imports" resource="http://www.ebusiness-
unibw.org/ontologies/eclass/5.1.4/"></div>
   <div rel="owl:imports" resource="http://purl.org/goodrelations/v1"></div>
   <div property="rdfs:label" content="RDF/XML data for Techn. Business, based on
http://purl.org/goodrelations/" xml:lang="en"></div>
  </div>
  <div class="description" about="http://www.oettl.it/#BusinessEntity"
typeof="gr:BusinessEntity">
     <div rel="gr:hasOpeningHoursSpecification">
        <div class="description"
about="http://www.oettl.it/#OpeningHoursSpecification_Sat_am"
typeof="gr:OpeningHoursSpecification">
         <div property="gr:closes" content="12:00:00" datatype="xsd:time"></div>
         <div rel="gr:hasOpeningHoursDayOfWeek"
resource="http://purl.org/goodrelations/v1#Saturday"></div>
         <div property="gr:opens" content="08:00:00" datatype="xsd:time"></div>
        </div>
     </div>
     <div rel="gr:hasOpeningHoursSpecification">
        <div class="description"
about="http://www.oettl.it/#OpeningHoursSpecification_Mon-Fr_pm"
typeof="gr:OpeningHoursSpecification">
         <div property="gr:closes" content="18:00:00" datatype="xsd:time"></div>
         <div rel="gr:hasOpeningHoursDayOfWeek"
resource="http://purl.org/goodrelations/v1#Thursday"></div>
         <div rel="gr:hasOpeningHoursDayOfWeek"
resource="http://purl.org/goodrelations/v1#Wednesday"></div>
         <div rel="gr:hasOpeningHoursDayOfWeek"
resource="http://purl.org/goodrelations/v1#Monday"></div>
         <div property="gr:opens" content="13:00:00" datatype="xsd:time"></div>
        </div>
     </div>
...
```

Listing 1 – Example of GoodRelations RDFa Web page annotation

B2B Ontologies

Conversely from e-commerce applications, in the B2B domain the higher complexity leaves Semantic Web adoption one step behind. In this specific context semantic systems still have difficulties to completely satisfy the requirements and the construction of an adequate domain ontology. In this section we present the most relevant works that have been developed to breach this gap. Among them, we can find some common points like: i) similarly to e-commerce ontologies, all of them are developed starting from existing standards; ii) except the Ontolog Community with the UBL Ontology Project, all others develop a direct transformation from the XSD format to an ontology language, mainly OWL; iii) B2B ontologies are used to improve matching and discovery of heterogeneous definition of similar concepts, but none of them continue to use ontologies as a message exchange formalism directly; iv) all these B2B ontologies are in a proof of concept phase or ongoing works, but as far as we know, no real business transactions are formalised with the help of ontology adoption yet; v) the generated ontologies are applicable to only a specific set of input sources, strictly related to the selected standard. Only the SET ontology tries to develop a more generic reference model, but still too close to the standards related to the CCTS model (UN/CEFACT, 2003). This last work confirms our idea expressed above that the ebXML standard is gathering the largest consensus and this is naturally reflected in

the produced ontologies. Below we present the ontologies derived from the UBL, XBRL, RosettaNet, ebXML, GS1 and OAGi standards

UBL Ontologies

The Ontolog Community UBL Ontology Project[ii] started the design of the UBL ontology in March 2003. The aim of the project was to develop a formal ontology of the UBL Business Information Entities as defined by the UBL OASIS technical committee. The ontology is mainly hand made following the Ontology 101 method (Noy, 2001) and conceived as extensions of the Suggested Upper Merged Ontology (SUMO) (Niles, 2001). They started formalizing UBL terms in SUO-KIF (SUO Working Group, 2003) extracting nouns and verbs from a UBL specification source text, then looked for classes in SUMO for the nouns and verbs extracted and finally mapped related terms as being either equal, subsuming or instance of. Figure 10 shows a view of the UBL ontology using Protégé editor.

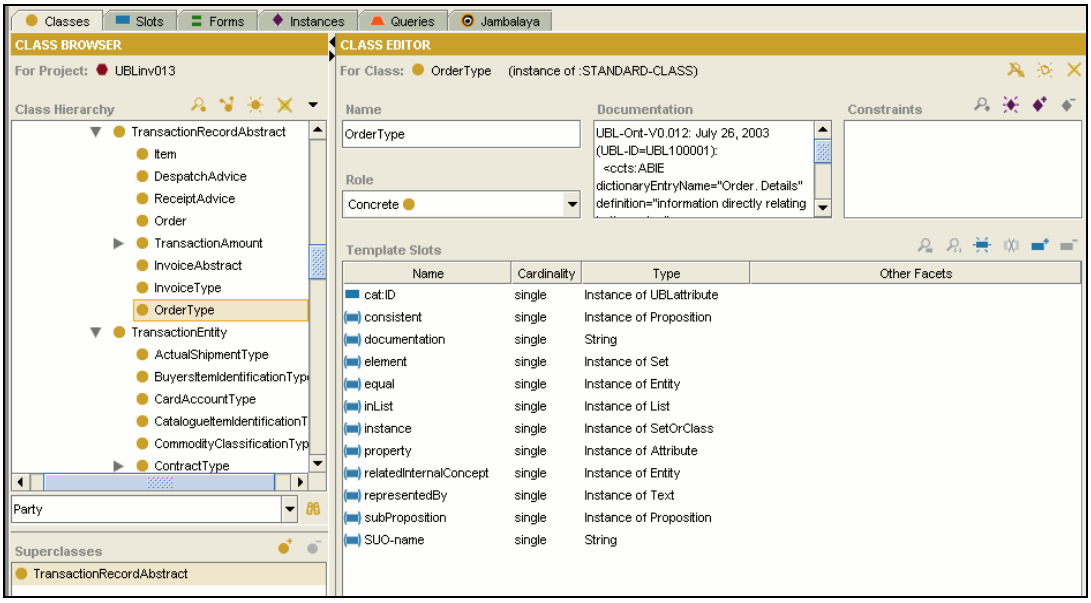

Figure 10 – Ontolog Community UBL Ontology view

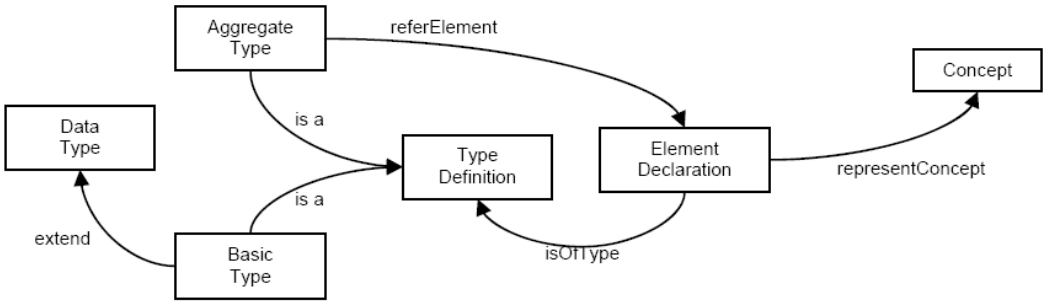

Figure 11 – Proposed UBL Component Ontology

Another experience targeting UBL Ontology has been developed by Yarimagan and Dogac (Yarimagan, 2008) from the Middle East Technical University. The so called UBL Component Ontology[iii] is generated automatically by a conversion tool that reads UBL schemas and creates corresponding class, object properties and existential restriction definitions in OWL.

The Component Ontology template, shown in Figure 11, represents relationships between entities, types and business concepts. Each *xsd:ComplexType* and *xsd:element* declaration is a corresponding subclass under *DataType*, *TypeDefinition*, *ElementDeclaration* and Concept root classes of the Component Ontology. Every UBL element represents a unique business concept or an entity. This allows the definition of multiple elements representing the same business concept/entity and their correspondence is expressed through their relation to the same Concept class.

Classes are related to each other through object properties where: Basic UBL types are defined through extending simple data types such as text, integer, date; the *referElement* object property represents the relationship between classes representing UBL aggregate types that refer to a similar set of elements; the *isOfType* object property represents the relationship between classes representing type definitions and element declarations; finally, the *representConcept* object property allows the definition of multiple elements that represent identical business concepts and relate element declaration classes to corresponding business concept classes. Listing 2 shows an example of the *ContactParty* concept expressed in OWL following the UBL Component Ontology representation.

XBRL Ontology Initiative

XBRL is a standard that formalizes financial reports. XBRL is used to define the so called XBRL taxonomies, which provide the elements that are used to describe information, instances, and give the real content of the elements defined. Ruben Lara *et al.* in (Lara, 2006) advocated the use of OWL as an alternative to XBRL and produced a set of OWL files able to describe DGI[iv], ES-BE-FS[v] and IPP[vi] taxonomies. For this they have developed a generic translation process of XBRL taxonomies into OWL ontologies[vii] so that existing and future taxonomies can be easily converted into OWL ontologies following the transformation rules defined in Table 2.

The conclusion was that extensions to OWL are required in order to fulfil all the requirements of financial information reporting, to incorporate mathematical relations and that while its semantics can be appropriate (e.g. for investment funds classification), they could sometimes be problematic (e.g. for validation purposes). Finally they validate the adoption of such an ontology to automate and improve the classification and discovery of funds but do not use them as a formal format for data exchange.

Parsed taxonomy element	Root OWL class	Direct OWL subclasses
XML complex types	DGI ComplexType	A subclass for each complex type
XBRL Tuples XBRL items	DGI Element	DGI Tuple DGI Item
XLink links	DGI Link	DGI LabelLink DGI PresentationLink DGI CalculationLink
XBRL Contexts	Context (range of properties is subclass of ContextElement)	Subclasses of ContextElement: ContextEntity ContextEntityElement (Identifier) ContextPeriod ContextScenario
XBRL units	Unit (range of properties is subclass of UnitElement)	Subclass of UnitElement: UnitMeasure

Table 2 – Summary of parsed taxonomy element translations

```
<owl:Class rdf:about=" urn:ubl:CAC-2#ContactParty">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource="#ContactPartyConcept"/>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#representConcept"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource=" urn:ubl:CAC-2#PartyType"/>
          <owl:onProperty>
            <owl:ObjectProperty rdf:ID="isOfType"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Class rdf:about="#ElementDeclaration"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:about="urn:ubl:CAC-2#PartyType">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:someValuesFrom>
            <owl:Class>
              <owl:intersectionOf rdf:parseType="Collection">
                <owl:Class rdf:about="urn:ubl:CBC-2#WebsiteURI"/>
                <owl:Class rdf:about="urn:ubl:CBC-2#EndpointID"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#PartyIdentification"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#PartyName"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#Language"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#PostalAddress"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#PhysicalLocation"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#Contact"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#Person"/>
                <owl:Class rdf:about="urn:ubl:CAC-2#AgentParty"/>
              </owl:intersectionOf>
            </owl:Class>
          </owl:someValuesFrom>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#referElement"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Class rdf:about="#TypeDefinition"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Listing 2 – Excerpt of the UBL Component Ontology

RosettaNet Ontology

Armin Haller *et al.* (Haller, 2008) developed a Web Service Modeling Ontology (WSMO) (Lausen, 2005) core ontology expressed in the WSML (De Bruijn, 2005) formal language for the Supply Chain Management based on the RosettaNet standard. The process of developing a complete Supply Chain ontology from RosettaNet schemas is carried out in two steps: i) the core ontology is obtained by a direct translation from XSD to WSML including a reconciliation phase to hierarchically structure the ontology and to add a proper subsumption hierarchy; ii) RosettaNet specifications are analysed to identify remaining sources of

heterogeneity in order to model and reference richly axiomatised ontologies, forming the outer layer in our ontological framework. As the previous experience they defined a set of rules from the XML representation to the selected ontology language, Listing 3 shows an example of such mapping from the XML extension element to its corresponding WSML formalism.

```
<xs:complexContent>
 <xs:extension base="uat:IdentifierType">
  <xs:sequence>
   <xs:element name="ProductName" type="xs:string" minOccurs="0">
   <xs:element name="Revision" type="xs:string" minOccurs="0">
  </xs:sequence>
 </xs:extension>
</xs:complexContent>

hasIdentifierType ofType extIdentifierType

concept extIdentifierType subConceptOf uat#IdentifierType
  ProductName ofType (0 1) _string
  Revision ofType (0 1) _string
```

Listing 3 – Example of Complex extension type mapping to WSML

Authors argued that their ontology is able to resolve most of the heterogeneity problems between different RosettaNet implementations that are not structurally and semantically covered by the RosettaNet specification.

The SET Harmonized Ontology

The SET Harmonized Ontology is an initiative of the OASIS Semantic Support for Electronic Business Document Interoperability (SET) Technical Committee[viii]. The purpose of this SET TC deliverable (Dogac, 2009) is to provide standard semantic representations of electronic document artefacts based on UN/CEFACT Core Component Technical Specification (CCTS) (UN/CEFACT, 2003) and hence to facilitate the development of tools to support semantic interoperability. The basic idea is to explicit the semantic information that is already given both in the CCTS and the CCTS based document standards in a standard way to make this information available for automated document interoperability tool support.

The resulting ontology[ix] provided by Asuman and Kabak is currently the most valuable effort in describing an upper ontology for the real B2B domain. The SET Harmonized Ontology contains about 4758 Named OWL Classes and 16122 Restriction Definitions. Their approach is a semi-automatic derivation of an ontology from the business data components defined by OAGIS, GS1, UBL and UN/CEFACT CCL, which are all B2B standards based on the CCTS specification. Another point of interest is that it is one of the rare experiences applying a strong adoption of Semantic technologies, like DL reasoners, SPARQL, OWL and OWL queries to derive a harmonized ontology. This can be viewed as similar to a merging operation.

Without delving into details Figure 12 shows an overview of the SET upper ontology. The overall process to get the harmonized ontology is as follows: i) first specify an upper ontology, which is an OWL description of the CCTS specification; ii) transform input source documents into schema ontologies, which are afterwards mapped manually to the defined upper ontology format and thus automatically transformed to OWL compliant files; iii) define four normative upper ontologies, one for each of the UBL, GS1 and OAGIS® 9.1 standards separately, while the UN/CEFACT CCL is considered as upper ontology of reference. While creating these ontologies,

the relations with the CCTS upper ontology classes are also established. Finally, with the help of additional heuristics, using a Description Logics (DL) reasoner, a Harmonized Ontology is computed.

The resulting ontology and heuristics enable the discovery of equivalences and subsumptions of structurally similar document artefacts between two document schemas. When translating such document artefacts, automatically generated XSLT rules are used, that produce query templates (SPARQL and Reasoner based queries) to facilitate the discovery and reuse of document components.

The advantage of this approach is twofold. Firstly it shows the powerful benefits of semantic technologies. Even with a more complex syntax description, a reasoner is able to autonomously discover several useful subsumptions and equivalences. It also shows that it is possible to provide a first real normative upper ontology formalization that could lead to a new era of B2B standard ontologies development.

However a strong and somewhat limitative hypothesis is that input sources must be compliant with the CCTS specification. This is not applicable to the whole domain and thus prevents a larger adoption of this solution. It is also unclear how the different semantics of input elements are matched. For example, as presented in Figure 13, it is not clear how the *NameAndAddress* class has been associated to the owl *Address* class. For instance an automatic matcher should have to choose between the classes *Name* and *Address*, which is not the case in the resulting ontology. Another example is the detection of the semantic equivalence between *Postal_zone* and *Postcode*, which is not explained.

To conclude, this approach also lacks the definition of a semantic matcher and we argue that the integration of such a module could improve resulting correspondences and help with possible ambiguities.

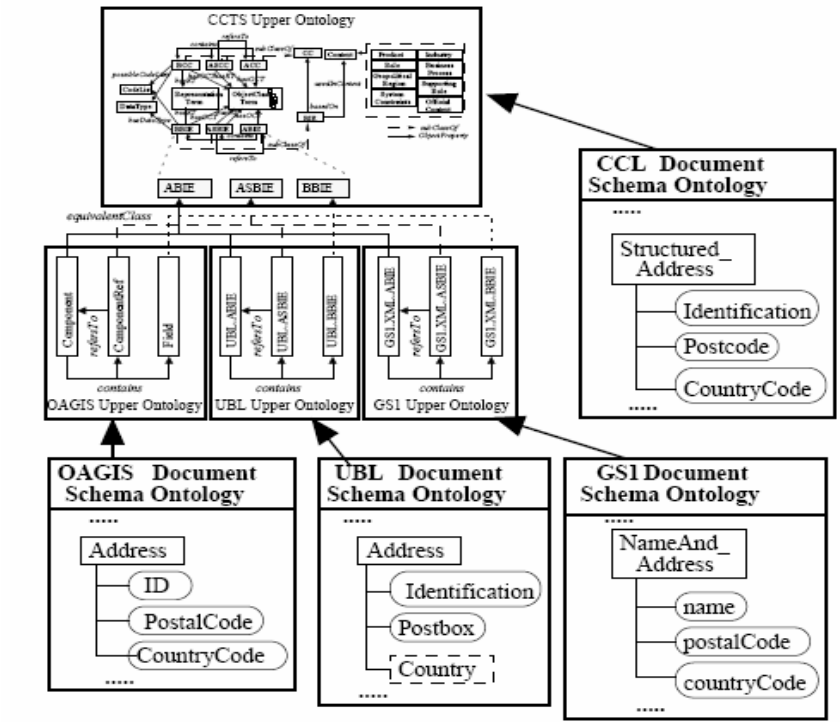

Figure 12 – An Overview of SET Upper Ontologies and Document Schema Ontologies

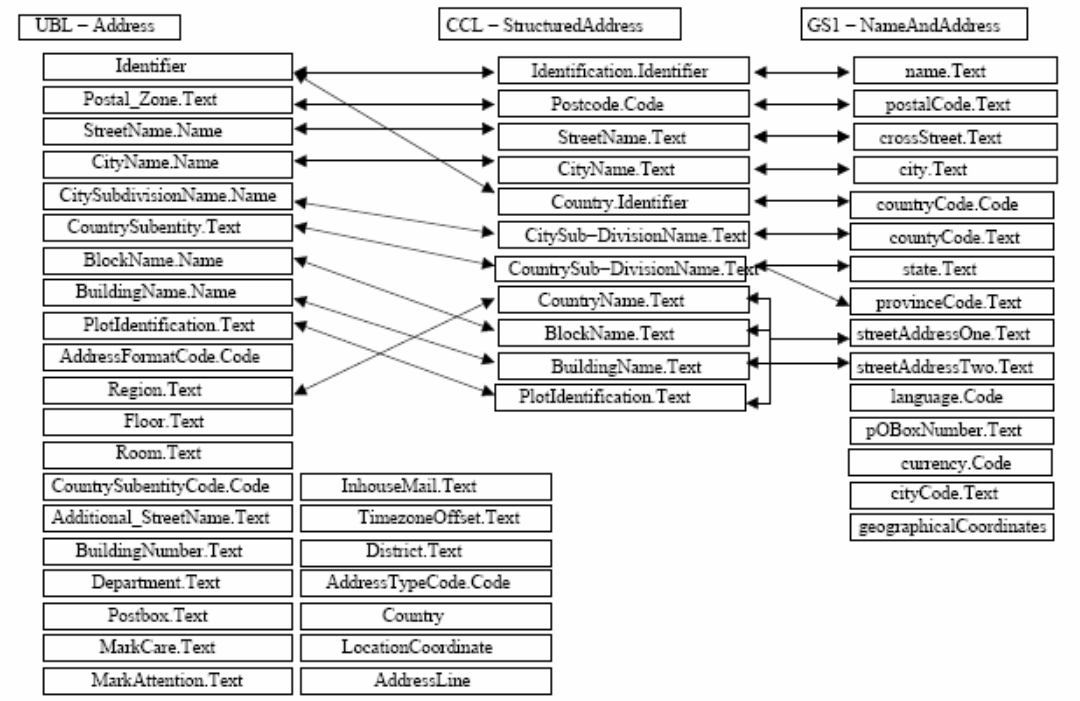

Figure 13 – The Semantic Equivalences among the BBIEs of UBL-Address, CCL-Structured Address and GS1-NameAndAddress Discovered through the Harmonized Ontology

JANUS: AUTOMATIC ONTOLOGY BUILDING SYSTEM FROM XML SCHEMAS

Over the past ten years, the Semantic Web wave has shown a new vision of ontology use for application integration systems. Researchers have produced several software tools for building ontologies (like Protégé or OntoEdit) and merging them two by two (like FCA Merge or Prompt) or producing alignments (like S-Match, OLA, Mafra, H-MATCH, COMA). Nevertheless these solutions, as well as adopted ontology building methodologies, are mainly human driven or sometimes assisted by semi-automatic software tools. Furthermore, all of them make reference to either an upper or domain ontology to improve the run-time automatic matching that often is inadequate, if it exists at all.

Limitations to their adoption for integration of enterprise applications, among others reasons, are: (i) the lack of tools capable of extracting and acquiring information from a large collection of XML files (the "de-facto" format for applications information exchange definition); (ii) the complexity of aligning and merging more than two sources, a complex task excessively consuming of computational time; (iii) the difficulty of validation based on background knowledge hard to produce and maintain.

The aim of this section is to introduce Janus, the software that we have developed. This system is an implementation of our approach to ontology generation integrating SDMO, a Semantic Data Model for Ontology, extracting information from XML Schemas and capable of providing a solution to the limitations described above. Indeed as we show with our experimental results, it is able to automatically generate and maintain a collective memory resource that facilitates the discovery of alignments when matching concepts in a given domain with satisfactory results.

The section is outlined as follows. Firstly we analyse the matching problem as it is seen by systems aiming the integration of data. As consequence of the shortcomings of the studied architectures we propose a semantic data model as solution to solve the multiple inputs integration problem. We finish with the overall presentation of our prototype.

The Matching Problem

Even when input sources are either well formed ontologies or XML Schemas, definitions can be similar but also heterogeneous, semantics different, and thus the discovery of correspondences is probably the most basic, and at the same time the most challenging task that must be conducted.

The matching problem is often related to ontology learning and matching and it has been largely investigated in literature. Among them, we can cite the paper by Mehrnoush and Abdollahzadeh (Mehrnoush, 2003) which proposes a complete framework for classifying and comparing ontology learning systems. The authors propose six main categories (called *dimensions*) as follows: **elements learned** (concepts, relations, axioms, rules, instances, syntactic categories and thematic roles); **starting point** (prior knowledge and the type and language of input), **pre-processing** (linguistic processing such as deep understanding or shallow text processing); **learning methods** including also an evaluation about the degree of automation (manual, semi-automatic, cooperative, full automatic); the **result** (ontology vs. intermediate structures and in the first case the features of the built ontology such as coverage degree, usage or purpose, content type, structure and topology and representation language); and finally **evaluation methods** (evaluating the learning methods or evaluating the resulted ontology).

We share the most part of the conclusion of their analysis, especially regarding the importance of input sources, which of course are essential to the automation process and highly influence the result of the final learned ontology. In fact ontology learning systems extract their knowledge of interest from inputs, which can differ by type and language (e.g., English, German or French). Types can be **structured data** like already existing ontologies, some schemata or lexical semantic nets such as WordNet. Other sources for ontology learning systems are **semi-structured data** such as dictionaries, HTML and XML schemas and DTDs (document type definitions), which probably constitutes in the Web environment the most hot topic today. Finally, the most difficult type of input from which to extract ontological knowledge are the **unstructured** ones (e.g., free text). Tools that learn ontologies from natural language exploit the interacting constraints on the various language levels (from morphology to pragmatics and background knowledge) in order to discover new concepts and stipulate relationships between concepts (Aussenac, 2002). Finally the authors of (Mehrnoush, 2003) assert that the first two kinds of input data are more appropriate to build ontologies for the Semantic Web, thus with DL implications, while the latter is more adapted to build more general lexicons such as taxonomies or dictionaries.

They also identify some open problems to be considered to improve the field, in particular: (i) the way to **evaluate** ontology learning systems, currently evaluated only on the basis of their final results; no measure is defined for specific parts of the learning process proving the accuracy, efficiency, and completeness of the built ontology. (ii) **Full automation** of ontology learning process is not described yet and integrating successful **modules** to build complete autonomous systems may eliminate their weaknesses and intensify their strengths. (iii) At last, moving toward **flexible** neutral ontology learning method may eliminate the need for reconstruction of the learning system for new environments.

Moving forward the automation process to enter in more technical surveys, in (Buitelar, 2005) authors provide a comprehensive tutorial and an overview on learning ontology from text. Rahm

et al. (Rahm, 2001) present an overview on techniques used for the schema matching automation. Euzenat *et al.* in (Euzenat, 2004) provide a detailed overview and classifications of techniques used for ontology alignment and a *state of the art* on existing systems for ontology matching/alignment, probably the best known software at present. From the book *Ontology Matching* by Euzenat and Shvaiko (Euzenat, 2007), which probably represents the most complete work in the current literature around the matching theme, not only techniques by also theoretical aspects and definitions involved into the matching process as well as their evaluation measures are presented. As last, let us cite the survey presented by Castano *et al.* (Castano, 2007), which provides a comprehensive and easily understandable classification of techniques and different views of existing tools for ontology matching and coordination.

Moreover into the area of data and knowledge management we can find interesting surveys in (Do, 2002; Doan, 2002; Ehrig, 2004) and still more focused on semantic integration in (Noy, 2004b; Shvaiko, 2005).

All these works provide a real detailed overview of the matching problem, ontology generation tools and aspects of possible automation, at least for some specific tasks. As such, it is not the scope of this chapter to provide an overview of them. Indeed, even if the frontier between matching and generation tools is not always clearly definable, we can say that except the first one, all referenced papers mainly focus on the matching step but do not cover the whole ontology automation process, that is finally what we target with the system we have implemented. We can also add that the matching problem is probably the most challenging part and this is the reason why we analyse it more deeply below.

Known Matching Features

Classical matching approaches lack efficiency. This can be explained by three main reasons: (i) the algorithm computational complexity order; (ii) the fact that algorithms compute measures between every couple of items of ontologies to map, even when they do not have anything in common; (iii) the lack of memorization: a comparison is done every time two items are met, regardless of what has already been calculated.

As we can see from existing works, many researchers in the Semantic Web and Knowledge Engineering communities agree that discovering correspondences between terms in different sets of elements is a crucial problem. Sometimes two ontologies refer to similar or related topics but do not have a common vocabulary, although many terms they contain are related. So this complex task requires the application of several algorithms (each algorithm realizes at least a matching operation) and once again we lose efficiency. Consider looking for correspondences between sets of elements more complex than that presented in the example above: Figure 14 illustrates a non exhaustive list of possible mismatches that can be established between the definitions of a same high level concept expressed in XML Schema format. For instance the example shows two different vision of the concept address as defined by two B2B standards, OAGIS and Papinet. It is clear that although both of these standards are based on the "upper" standard UN/CEFACT CCTS, there are considerable differences in the resulting document fragments. This illustrates why we need more than one algorithm to discover possible similarities between two sets of elements. To this end we provide a first classification of the nature of these algorithms categories: syntactic, semantic, and structural. A good process for matching discovery should cover at least these three categories and also implement a combination of them in order to improve results.

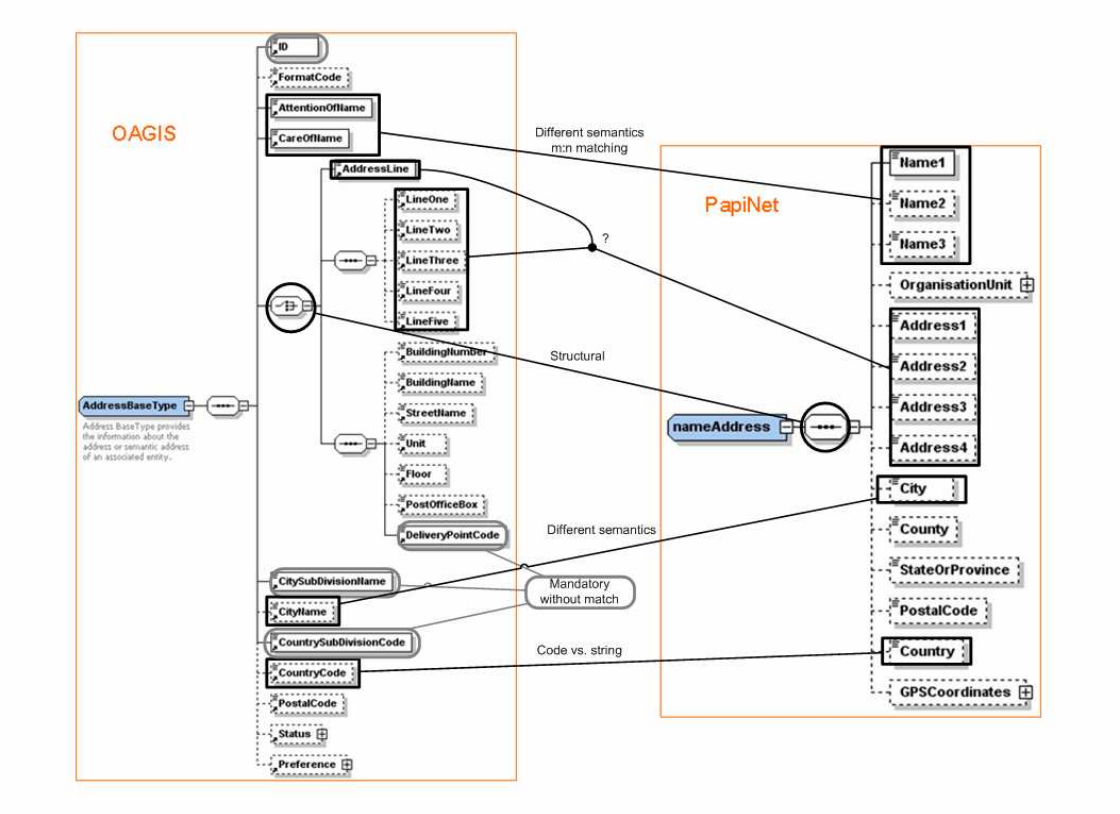

Figure 14 – Example of possible mismatchings between two XML Schemas definitions

The Matching Process

As already mentioned above matching problems can be approached from various standpoints and this fact is reflected by the variety of the definitions that have been proposed in the literature. We observe that there are some recurring terms often leading to confusion and thus producing overlaps on the process definition. *Learning*, *matching*, *anchoring*, *alignment*, *transformation*, *mapping* and *merging* are almost used to this purpose. Figure 15 proposes a view about the role and sequence that each of these common terms play in the ontology "life-cycle" process.

The *Learning* phase aims to extract knowledge information from sources handling their different representations. As output it provides a formal representation, sometimes an ontological view of inputs. From here we assume that we have two or more input ontologies. This term often refers to a larger operation that comprises the final ontology generation, but we prefer to use this term just to highlight the fact that ontological knowledge is mainly retrieved, thus learnt, at this stage of the process. The *Matching* phase realises similarity detections between input entities executing one or more algorithms. As described previously, the "matcher" (the application realising this phase) computes the algorithms for each couple of input entities and provides as output a list of the best matches found, selected on the basis of parameters. The following *Alignment* phase tries to select the best set of correspondences between all those provided by the matcher. It permits to combine the different similarity algorithms executed previously and to provide a uniform view of correspondences, normally without inconsistencies. At this stage the match can be also contextualized, choosing a match rather than another because of heuristic practices or if an existing upper ontology for the concerned domain suggests so.

Finally, depending on the purpose, alignments can be used to merge input ontologies (*Merging* phase) or to transform instances of an ontology into another (*Mapping* phase).

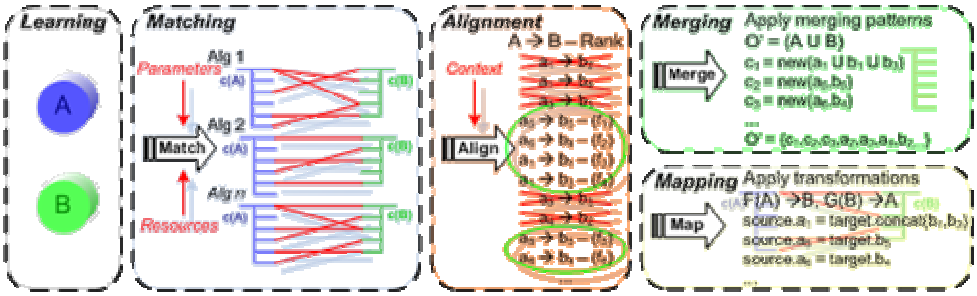

Figure 15 – *Ontology learning, matching, alignment, mapping and merging phases*

This disambiguation enables us to situate the problem that we want to address well.

The *Matching process* considers only the matching phase described above. In our analysis we argue that this is a core part that: i) mainly contributes to the computation time and; ii) is the most generic and thus reusable part. These are the main reasons that conducted us to look for a scalable solution to improve the whole ontology generation process in this phase.

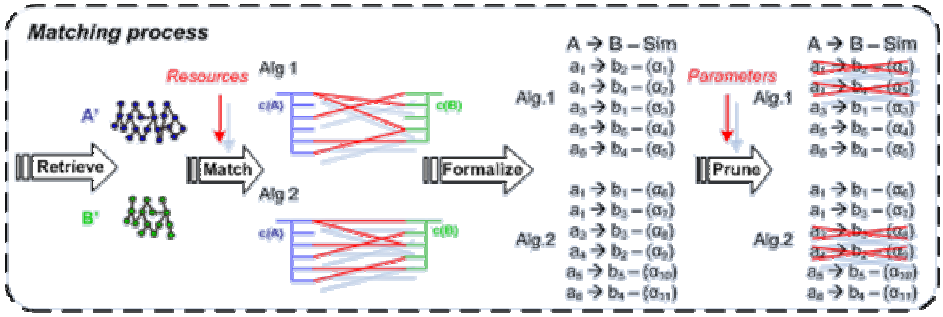

Figure 16 – *Matching process details*

As shown in Figure 16 the matching phase can be split in different steps. The *Retrieve* step takes as input information extracted from sources, and transforms this knowledge in an internal ontology matching format, sometimes called reference model. In its simpler form it is a list of terms representing semantics of input entities, and in other cases it can be a more complex Galois lattice representation like in (Stumme, 2001). Subsequently the *Match* step is able to execute similarity algorithms and *Formalizes* results with a correspondent confidence value for each match found. Some algorithms, like synonymy detection, can also require external resources (e.g.: WordNet or electronic dictionaries). Thresholds and some heuristic are used in the *Prune* step to filter sets of matches. Techniques for matching sources are really numerous and the survey published in (Euzenat, 2007) is a good reference for discover and compare them.

The Semantic Data Model for Ontology

In this section, we describe the Semantic Data Model for Ontologies (SDMO) defined to provide an organized model to record as much knowledge as possible for matching systems. The goal is improving the concept correspondences similarity detection. The improvement that we target with this model is the machine capability to recognise similar concepts faster, on the basis of their

relationships and consequently the ability to adopt more efficient algorithms to refine mappings, thus overcoming the matching problem seen above.

The basic representation of SDMO is data about concepts and relationships. Such **object-based** modelling allows a high level of data definition independent from the different representations. A second basic precept of our model is that many relationships are **functional** like they are in nature. These functional relationships are often called *has attribute* in models like the Relational Model and Entity-Relationships, or *functional property* in OWL. In our model these relations are part of the set of what we call **structural** relationships which also provide hierarchical mechanisms for building object types out of other object types. For example, *address* and *postal address* that might be the aggregation of *street*, *city*, and *country*.

A third basic precept is the **semantic** relationship, which specifies the fact that some concepts share a common meaning, like synonyms.

A fourth basic element of the model is the set of **syntax** or **linguistic** relationships. The aim of this kind of concept relations is to maintain the link among concepts sharing a similar name, like *postcode* and *postal code* attributes, or names sharing the same stem. This kind of relation brings us more inside the characteristics that we want to give to the model. These are not natural human precepts that we find in other models for the real-world representation, but rather a natural feature for matchers, needed to compute an operation.

The fifth and final basic element is a link to the original input. A matcher usually normalizes initial labels and during this operation some little details can be lost; yet it is important to maintain the link with the source in order to be able to regain the original context or to produce a mapping. In our model these relations are part of the set called **source** relationships.

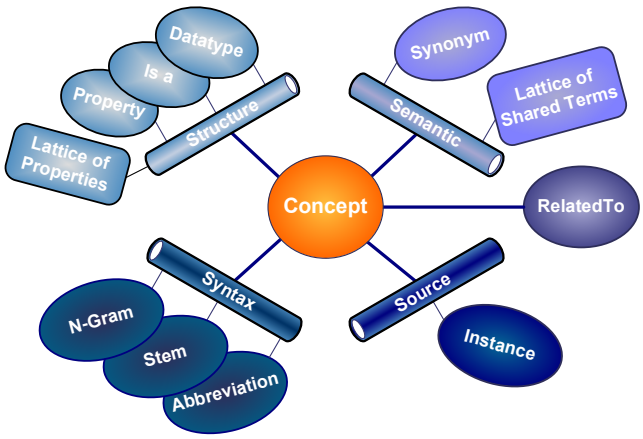

Figure 17 – SDMO Concept relationships overview

Figure 17 shows the overall view of SDMO concept relationships. A SDMO concept is the constituent entity of the model and is defined as a quadruple:

$$c = < l, R, S, f >$$

Where:

- **l** is a set of words, simple or compounds, that best represents the name of the concept. Among them we also define a *preferred label* as the best representative label as concept

name (e.g.: having equivalent concepts named *geographical_coordinate* and *coordinate*, they can be merged to form the same concept and the final name can be one of them)

- **R** is the set of relations between concepts (all seen above)
- **S** for Source, is the set of originating instances of a concept (not to be confused with instances as individuals in OWL representations)
- **f** is a frequency and/or rank measure

Moreover, similarly to UML and many other models, in SDMO we defined three basic kinds of concepts, also called nature of the concepts, but a concept can be of more kinds at the same time or change all over its "life in the model". No mandatory relationships are required beforehand for a concept, but depending on them, we can determine dynamically its nature. These three types are: *class*, *property* (or *attribute*) and *printable-type*.

The main concept type is called **class** and corresponds intuitively to non atomic concepts, thus to concepts characterised by a finite set of attributes. The second basic nature of a concept is the **property** (or attribute). It represents either a specific and atomic characteristic of a class or also a role that semantically redefines another concept class, like an UML association (e.g. *address* that becomes a *residence for a* person or a *delivery address* in another context). The former typically corresponds to concepts in the world (of data exchange) that have no underlying structure. Simple examples are *first name* and *last name* of a *person*, or *city name*, *etc.* The latter and most basic concept type in the SDMO structure is the **printable type**. This kind of concept can be also considered as the type that serves as the basis for application inputs and outputs. It can be a conventional basic type, such as *string* or *integer* or a more complex representation of a printable data type like *measure*, *amount*, or *text* that in turn are directly linked to basic types.

We stress out the fact that a concept can be of different types at the same time, they are not strictly closed to be of only one nature at once, but depending on their behaviours they can be seen for example as a class or a property. For instance a **class property** SDMO concept is allowed and is a non atomic concept, thus a class, which is also property for another concept class.

We have also defined a SDMO graphical representation that provides a global view of concepts organization with their relationships. Figure 18 illustrates the graphical syntax we use to describe a SDMO schema.

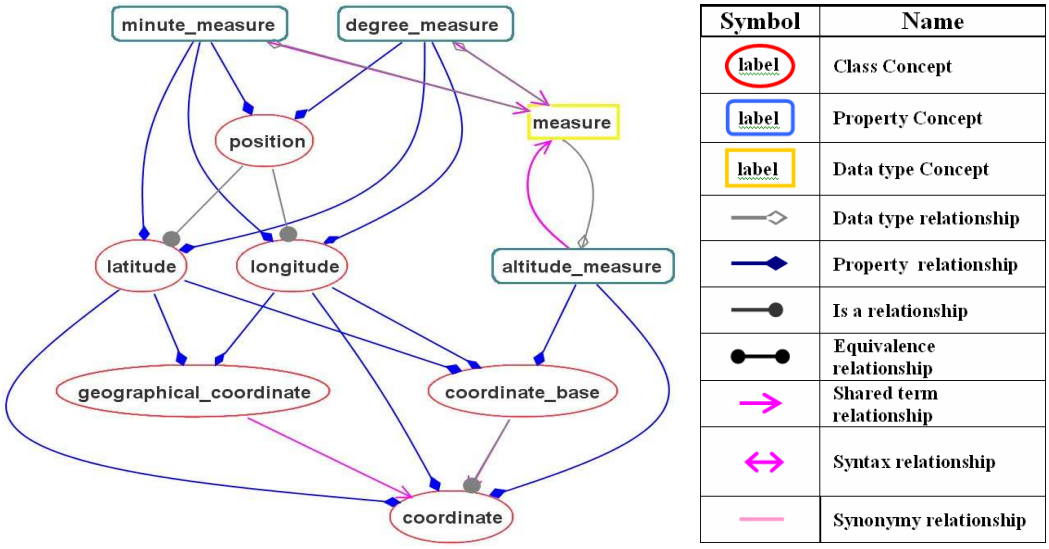

Figure 18 – SDM Graphical Representation

Implementation

Janus is a system that enables the automatic generation of dynamic ontologies from XML Schemas. It is an implementation of the system described throughout previous Sections. Figure 19 shows the overall architecture of Janus.

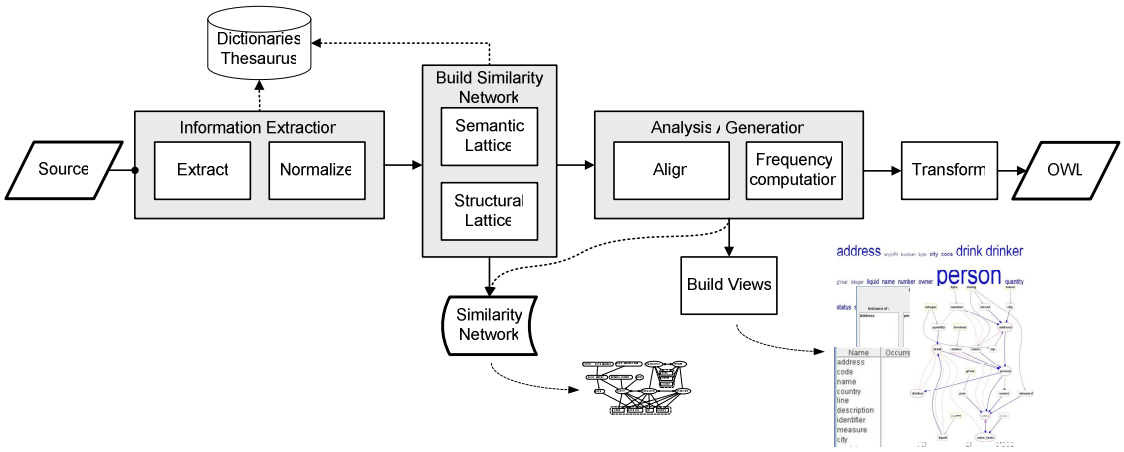

Figure 19 – Janus overall architecture

The extraction task represented by the **Extract** arrow and **Normalize** rectangle supplies the knowledge needed to generate the ontology. This knowledge is merely composed by candidate concepts, properties, printable types, relationships of different nature and at the same time it contains counters and ranks for each element. Implemented techniques for knowledge acquisition are a combination of different types, such as: NLP (Natural Language Process) for morphological and lexical analysis, association mining for calculating term frequencies and association rules, semantics for finding synonymy, and clustering for grouping semantic and structural similar concepts. We call **XML Mining** the adaptation of these techniques applied to XML schemas.

XML Mining is used to parse sources to extract XML constructs and to process XML tags declarations. In addition it also includes a pre-matching treatment that aims to mutualise element's processing that are clustered in a Galois Lattice and Formal Concept Analysis based form. This treatment provides as output a pre filled model ready for automatic analysis.

The following step is **build semantic network** represented by the corresponding block. This step finalizes the model integrating information coming from external sources, like other existing ontologies or thesaurus. Moreover at this stage we do not look at similar concepts to be merged, but only execute matching algorithms to collect as much correspondences as possible among them. All these connections are stored and maintained in the model in order to be quickly detected and not recalculated in future integrations.

The **Analysis** step aligns correspondences and looks for equivalent concepts to be integrated. This step establishes the best similarities and analyses the model to unveil new possible relations and correspondences not directly detected by matching algorithms and computes frequency and rank measures.

The **Generation** step finalizes the meta-model used by the tool into a final semantic network. The final model can be serialized in OWL built by the **Transform** module. Finally the **Build Views** module derives useful views from the network provided to users.

The implementation phase of the prototype has been more complex than expected in the beginning and this for a lot of more or less little problems we met. Problems generally were not directly linked to the system approach but more of a technical nature. Like the lack of matching API adequate to our scope, the lack of software capable of extracting information from XML schemas rather than text corpus or OWL and last but not least the lack of reference ontologies for tests and developments. Despite these numerous problems we have been capable of validating the initial hypothesis that the model we designed to maintain a sort of memory of concepts correspondences is realisable and its implementation is scalable. It can manage large input sources and new sources can be added incrementally. Current problems are more linked to implementation issues and a good compromise between storage and real time requirements can resolve the most part of them. In the first case if we target a system with low physical space requirement we can store only information extracted. Conversely if we target run time applications we can store the whole generated model that provides very fast similarity detection with acceptable precision. Thus, the system coupled with advanced matching systems can provide a very useful support to run time data integration.

More detail on the implementation and results can be found in (Bedini, 2008; Bedini, 2008b; Bedini, 2008c; Bedini, 2010).

Perspectives

The system we have developed is only a part of the whole architecture to achieve run-time data integration with the adoption of semantics technologies. Semantic data must be produced at the source and conceived as such, their direct transformation is still to hard to be completely and safely automated. Nevertheless our system provides an essential part of the architecture that right now has been misled, the lack of domain ontologies. Although it has been designed for a more general use-case, its behaviours have been profiled over the e-business domain. Its early adoption can be seen as a facilitator to the fast transformation of existing e-business XML documents into a skeleton of an ontology to quickly build and test a semantic matcher for the domain. Indeed it is quite fast and is only costly in computing resources during the generation of the model calculations. The graphical representation is very powerful and with a lot of visualizations options

and visual measures (like importance of an edge or a concept with respect to others) are available and of simple understanding for both human and software implementations. These are the reasons why we believe that our system achieved the initial requirement to be able to extract very useful knowledge from a large set of XML Schemas belonging to a common domain that can be simply translated into an ontology.

Beyond what we have implemented another general trend of earlier semantic adoptions in the domain are related to the SOA (Service Oriented Architecture) paradigm. Indeed the growing number of services available on the Web and the tendency to split legacy software in a choreography of services require a more advanced description of both data and services. Again the adoption of Semantic technologies (i.e. OWL with SA-WSDL (Farrel, 2007) formalisms) is the best alternative to follow for the next few years.

CONCLUSION

In this chapter we presented the e-business domain, with a more specific focus on the B2B domain, the requirements that it currently imposes on companies and their information systems in order to support business messages exchanges. Through this analysis we pointed out the current architecture limitations and explained why ontologies are the best approach to follow to gain in flexibility and dynamicity.

Nevertheless facts show that it is still not the case and e-business standards, which are the most adopted solutions for e-business, do not define standards as ontologies but only as XML Schemas. Although it is already a respectable improvement with respect to older systems like EDIFACT, they still require relevant human effort to be operational.

In this sense we have provided an analysis of e-business ontology requirements and summarized them into the need of a dynamic knowledge that can be built incrementally. Afterwards we have presented some well-known ontologies for e-business. Despite the interest of these works, real businesses still seem hesitant to use them in their implementations. We have identified two main topics to develop, one is the definition of an enterprise semantic repository, and the other one is a way to facilitate the automation of business document mapping. Finally we have presented a system that facilitates, by automation, the transformation from the current model to the "next one", from XML to OWL, believing that the existing gap can be breached by improving this direction.

After a large overview of e-business standards and their derivate ontologies, we have seen that existing systems aiming at data integration are strictly related to ontology and matching systems. Research in this area is active and some frameworks dedicated to the e-business domain are already appearing. The current lacking we have identified is the need for domain ontologies in order to provide the necessary reference knowledge to improve existing matching systems. Moreover, the adoption of Semantic Web technologies to business messages exchanges has an essential requirement, which is that messages must be semantically well defined using ontologies. To this end we have detailed a first prototype that provides a general viable solution.

REFERENCES

Anicic, N., Ivezic, N., & Jones A. (2005). Semantic Web Technologies for Enterprise Application Integration. In Proceedings of the *International Journal ComSIS Vol.2, No.1*.

Aussenac-Gilles, N., & Maedche, A. (2002). *Workshop on Machine Learning and Natural Language Processing for Ontology Engineering*. In conjunction with the ECAI'02 conference, Lyon, France, July 22-23, 2002.

Batres, R., West, M., Leal, D., Price, D., & Naka, Y. (2005). An Upper Ontology based on ISO 15926. In proceedings of *European Symposium on Computer Aided Process Engineering (ESCAPE 15)*. Barcelona, Spain. June 2005.

Bedini, I., Nguyen, B., & Gardarin, G. (2008). *Janus: Automatic Ontology Builder from XSD files*. Developer track at 17th International World Wide Web Conference (WWW2008). Beijing, China, April 21 - 25, 2008

Bedini, I., Nguyen, B., & Gardarin, G. (2008b). *B2B Automatic Taxonomy Construction*. In Proceedings 10th International Conference on Enterprise Information Systems. 12 - 16, June 2008 Barcelona, Spain.

Bedini, I., Gardarin, G, & Nguyen, B. (2008c). *Deriving Ontologies from XML Schema*. In Proceedings of the Entrepôts de Données et Analyse en Ligne (EDA), France, June 2008. RNTI, Vol. B-4, 3-17 (Invited Paper).

Bedini, I. (2010). *Deriving ontologies automatically from XML Schemas applied to the B2B domain*. Doctoral dissertation, University of Versailles, France. (Available from: http://bivan.pagespro-orange.fr/Janus/index.html)

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), pp 34-43.

Buitelaar, P., Cimiano, P., Grobelnik, M., & Sintek, M. (2005). *Ontology Learning from Text Tutorial*. ECML/PKDD 2005 Porto, Portugal; 3rd October - 7th October, 2005. In conjunction with the Workshop on Knowledge Discovery and Ontologies.

Castano, S., (Ed.). (2007). *State of the Art on Ontology Coorination and Matching*. BOEMIE Project. Deliverable 4.4 Version 1.0 Final, March 2007.

Coates, A.B. (2007). Semantic data models and business context modeling. Invited speaker at *XML2007*. Boston, Massachusetts, USA. 3-5 December 2007.

Corcho, O., & Gomez-Perez, A. (2001). Solving integration problems of e-commerce standards and initiatives through ontological mappings. In Proceedings of the *Workshop on e-business and Intelligent Web*.

D'Aquin, M., Haase, P., & Gómez-Pérez, J.M. (2008). NeOn - Lifecycle Support for Networked Ontologies: Case studies in the pharmaceutical industry. In proceedings of *European Semantic Technology Conference.* September 2008, Vienna, Austria.

De Bruijn, J., & Lausen, H. (2005). *Web Service Modeling Language (WSML)*. W3C Member Submission 3 June 2005. Available from: http://www.w3.org/Submission/WSML/

Do, H., & Rahm, E. (2002). COMA - A System for Flexible Combination of Schema Matching Approaches. In Proceedings of *28th International Conference on Very Large Databases* (VLDB 2002), Hong Kong, China.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002). Learning to Map between Ontologies on the Semantic Web. In Proceedings of the *11th International World Wide Web Conference* (WWW 2002), Honolulu, Hawaii, USA, pp. 662–673

Dogac, A., & Kabak, Y. (2009). *Semantic Representations of the UN/CEFACT CCTS-based Electronic Business Document Artifacts*. Draft OASIS Profile. Retrieved November 15, 2009.

E-Business W@tch observatory. (2007). *The European e-Business Report, 2006/07 edition*. 5th Synthesis Report of the e-Business W@tch, on behalf of the European Commission's Directorate General for Enterprise and Industry. (http://www.ebusiness-watch.org)

Ehrig, M., & Sure, Y. (2004). Ontology Mapping - An Integrated Approach. In Proceedings of the *1st European Semantic Web Symposium*, Heraklion, Greece, Springer Verlag, pp. 76–91

Euzenat, J., (Ed.). (2004). *State of the Art on Ontology Alignment*. Knowledge Web Deliverable D2.2.3. 2004.

Euzenat, J., & Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag, Heidelberg (DE).

Farrel, J., & Lausen, H. (2007). *Semantic Annotations for WSDL and XML Schema*. W3C Recommendation 28 August 2007.

Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., & Flett, A. (2001). Product Data Integration in B2B E-Commerce. *IEEE Intelligent Systems*, vol. 16, pp. 54-59.

Fensel, D. (2001b). *Ontologies: Silver bullet for knowledge management and electronic commerce*. Springer-Verlag, Berlin (DE).

Gruber, T. (2008). *Encyclopedia of Database Systems.* Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag.

Guarino, N. (1998). Formal Ontology and Information Systems. In Proceedings of *International Conference on Formal Ontology in Information Systems (FOIS).* Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

Haller, A., Gontarczyk, J., & Kotinurmi, P. (2008). Towards a complete SCM ontology: the case of ontologising RosettaNet. In Proceedings of *23rd Annual ACM Symposium on Applied Computing*, pp. 1467-1473.

Hepp, M. (2006). Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. *International Journal on Semantic Web & Information Systems*, Vol. 2, No. 1, pp. 72-99.

Hepp, M. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In Proceedings of the *16th International Conference on Knowledge Engineering and Knowledge Management*, Italy. Springer LNCS, Vol 5268, pp. 332-347.

Hepp, M. (2008b). E-Business Vocabularies as a Moving Target: Quantifying the Conceptual Dynamics in Domains. In Proceedings of the *16th International Conference on Knowledge Engineering and Knowledge Management*, Italy. Springer LNCS, Vol 5268, pp. 388–403.

Hepp, M. (2008c). *eClassOWL. The Products and Services Ontology*. Retrieved May 20, 2008, http://www.heppnetz.de/eclassowl/

Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, October 2003. ISBN13:9780321200686 ISBN10: 0-321-20068-3.

Kabak Y., & Dogac A. (2008). A Survey and Analysis of Electronic Business Document Standards. Under revision in *ACM Computing Surveys*.

Kajan E., & Stoimenov L. (2009). An Approach for Semantic-Based EC Middleware. In Proceedings of *e-Commerce 2009*. pp. 69-76.

IEEE SUO Working Group. (2003). *Standard Upper Ontology Knowledge Interchange Format*. IEEE P1600.1 Standard Draft. Available from: http://suo.ieee.org/SUO/KIF/index.html

Lara, R., Cantador, I., & Castells, P. (2006). XBRL taxonomies and OWL ontologies for investment funds. *1st International Workshop on Ontologizing Industrial Standards at the 25th International Conference on Conceptual Modeling.* Tucson, Arizona.

Lausen, H., Polleres, A., & Roman, D. (2005). *Web Service Modeling Ontology (WSMO)*. Member submission, W3C. Available from: http://www.w3.org/Submission/WSMO/.

Léger, A. (Ed.). (2002) *OntoWeb: ontology-based information exchange for knowledge management and electronic commerce*. OntoWeb D2.2 final. 2002.

Mehrnoush, S., & Abdollahzadeh, B. (2003). *The State of the Art in Ontology Learning: A Framework for Comparison*. The Knowledge Engineering Review, Volume 18, Issue 4.

Missikoff, M., & Taglino, F. (2003). Symontox: a web-ontology tool for ebusiness domains. In Web Information Systems Engineering. In Proceedings of the *Fourth International Conference on Web Information Systems Engineering*, pp 343-346.

Motta, E., & Sabou, M. (2006). Next Generation Semantic Web Applications. In Proceedings of the *1st Asian Semantic Web Conference*, China.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In Proceedings of the *International Conference on Formal Ontology in Information Systems* (FOIS), pages 2–9.

Noy, N.F., & McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

Noy, N. F., & Klein, M. (2004). Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems* 6(4), 428–440.

Noy, N. F. (2004b). Semantic Integration: a Survey of Ontology-based Approaches. *SIGMOD Record Special Issue on Semantic Integration.*

Rahm, E., & Bernstein, P.A. (2001). A survey of approaches to automatic schema matching. The *VLDB Journal* 10: 334–350. November 2001.

Shvaiko, P., & Euzenat, J. (2005). A Survey of Schema-based Matching Approaches. *Journal on Data Semantics* (JoDS).

Smith, B. (2006). Against Idiosyncrasy in Ontology Development. In Proceedings of *International Conference on Formal Ontology in Information Systems (FOIS)*. Baltimore, Maryland (USA), November 9-11, 2006.

Stumme, G., & Maedche, A. (2001). FCA-MERGE: Bottom-Up Merging of Ontologies. In Proceedings of the *17th International Joint Conference on Artificial Intelligence* (IJCAI), Seattle, WA, 2001.

Tran, D.C., Haase, P., Lewen, H., Munoz-Garcia, O., Gómez-Pérez, A., & Studer R. (2007). Lifecycle-Support in Architectures for Ontology-Based Information Systems. In Proceedings of the *International Semantic Web Conference.*

UN/CEFACT Techniques and Methodologies Group. (2003) *UN/CEFACT Core Components Technical Specification (CCTS)*. Part 8 of the ebXML Framework, ISO\TS 15000-5. Version 2.01, 15 November 2003.

Yarimagan, Y., & Dogac, A. (2009). A Semantic based Solution for the Interoperability of UBL Schemas. To appear in *IEEE Internet Computing Magazine*.

Zhao, Y., & Sandahl, K. (2003). Potential Advantages of Semantic Web for Internet Commerce. Proceedings of *International Conference on Enterprise Information Systems* (ICEIS), Vol 4, pp151-158, Angers, France, April 23-26, 2003.

Zhao, Y., & Lövdahl, J. (2003b). A Reuse-Based Method of Developing the Ontology for E-Procurement. In Proceedings of *Second Nordic Conference on Web Services* (NCWS'2003), ISBN 91-7636-392-9, Växjö, Sweden, Nov 20-21, 2003.

ADDITIONAL READING SECTION

Euzenat, J., (Ed.). (2004). *State of the Art on Ontology Alignment*. Knowledge Web Deliverable D2.2.3. 2004.

Euzenat, J., & Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag, Heidelberg (DE).

Fensel, D. (2001b). *Ontologies: Silver bullet for knowledge management and electronic commerce*. Springer-Verlag, Berlin (DE).

Hepp, M. (2007). Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. *IEEE Internet Computing 11(1)*: pp. 90-96.

Hepp, M. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In Proceedings of the *16th International Conference on Knowledge Engineering and Knowledge Management*, Italy. Springer LNCS, Vol 5268, pp. 332-347.

Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, October 2003. ISBN13:9780321200686 ISBN10: 0-321-20068-3.

Kent, W. *Data and Reality.* 1stBooks Library, rev. 3/28/2000. ISBN-13: 978-1585009701

Madhavan, J., Bernstein, P.A., Domingos, P., & Halevy, A. (2002). Representing and reasoning about mappings between domain models. In Proceedings of the *18th National Conference on Artificial Intelligence* (AAAI'02), Edmonton, Alberta, Canada, August 2002.

Motta, E., & Sabou, M. (2006). Next Generation Semantic Web Applications. In Proceedings of the *1st Asian Semantic Web Conference*, China.

Noy, N. F. (2004b). Semantic Integration: a Survey of Ontology-based Approaches. *SIGMOD Record Special Issue on Semantic Integration.*

Rahm, E., & Bernstein, P.A. (2001). A survey of approaches to automatic schema matching. The *VLDB Journal* 10: 334–350. November 2001.

UN/CEFACT Techniques and Methodologies Group. (2003) *UN/CEFACT Core Components Technical Specification (CCTS)*. Part 8 of the ebXML Framework, ISO\TS 15000-5. Version 2.01, 15 November 2003.

KEY TERMS & DEFINITIONS

Design-time: Design time covers all the necessary tasks for modeling and for setting up the execution of B2B collaborations. This phase involves the business process specification, the partner profile definition, the trading partner contract establishment, the business document conception and the message exchanges integration (or mapping) to the existing information system. Design time also includes the discovery and retrieval of existing business data.

Run-time: Run time covers the real execution of business exchanges from beginning to their termination. (i.e., business processes execution, messages exchange and dynamic services discovery).

B2B: Even though in this document we tend to use B2B as term to describe the environment of our research, electronic message exchanges are not limited to businesses. Administrations are increasingly confronted with similar problems in their relationships with companies or other administration departments: they need to provide high quality services to a wide audience, targeting both private and public sectors, while improving their efficiency and reducing their costs. Even internally, companies need dynamic message exchange solutions.

Ontology: An ontology is an explicit specification of a conceptualization (Gruber, 2008)

Ontology evolution: with evolution of an ontology for the e-business data integration we specifically mean an ontology as a dynamic characteristic of the domain. Thus evolution should not be equivalent to a classical versioning system, but more to a learning system, including a merge operation without loss of information and backward compatibility

[i] http://www.cxml.org
[ii] http://ontolog.cim3.net/cgi-bin/wiki.pl?UblOntology
[iii] http://www.srdc.metu.edu.tr/ubl/UBL_Component_Ontology.owl

[iv] DGI stands for General Data Identification of economic agents Spanish taxonomy de agentes económicos (DGI as Spanish acronym)
[v] DGI is the Financial information report taxonomy for the Estados Públicos Individuales y Consolidados
[vi] ES-BE-FS is the Taxonomy of the Stock Quote Exchange National Commission
[vii] The resultant OWL ontologies can be found here:
http://www.tifbrewery.com/tifBrewery/resources/XBRLTaxonomies.zip
[viii] http://www.oasis-open.org/committees/set/
[ix] The SET Harmonized Ontology is publicly available from http://www.srdc.metu.edu.tr/iSURF/OASIS-SET-TC/ontology/HarmonizedOntology.owl