B2B AUTOMATIC TAXONOMY CONSTRUCTION

Keywords: Automatic Taxonomy Generation, Semantic Web, B2B, XML Mining.

Abstract: The B2B domain has already been subject to several research experiences, but we believe that the real advantage of introducing semantic technologies within enterprise application integration has not yet been investigated fully. In this paper we provide a new use case for the next generation Semantic Web applications with regards to enterprise application integration. We also present the results of our experience in automatically generating a taxonomy from numerous B2B standards, constructed using *Janus*, a software tool we have developed in order to extract semantic information from XML Schema corpora. The main contribution of this paper is the presentation of the results of our tool.

1. INTRODUCTION

One of the most frequently asked questions during exchanges with other colleagues is surely: "Why introduce ontologies in the area of enterprise applications integration and interoperability? What is their contribution and what are the new benefits compared to existing technologies?"

While current solutions work, and enterprises are able to exchange electronic information between each other, as testified by the several B2B standard bodies available, several experiences nevertheless show it is practically impossible to connect two or more enterprise applications that implement two different standards without any additional developments, even if both standards claim conformance to the same base and same type of message! An example of this is shown by (Anicic, 2005), where authors argue that the integration of two applications, one based on the Standards in Automotive Retail (STAR) and the second on the Automotive Industry Action Group (AIAG), where both of their native interfaces are based on the Open Application Group (OAG) standard, requires the construction of a supplementary external module to connect them. Many other similar examples exist, and form the motivation of this work.

Advantages of Semantic Web (SW) adoption for enterprise applications integration has been widely recognised (Perez 1999), (Fensel, 2001a), (Leger, 2002), (Zhao, 2003a). However the predominant view of application integration is that it will be completely performed at *design time*, i.e. when deciding on integration rules between applications, rather than being performed at *run time*, i.e. during the business exchange execution.

Obviously, this new scenario brings novel challenges for application integration that can not be entirely resolved by SW, but surely it plays a fundamental role that can not be mistreated and unemployed by enterprise application solutions.

As always shown by (Anicic, 2005), the problem is that even under the hypothesis that enterprises and B2B standard bodies will produce ontology for defining business documents, the lack of a background reference knowledge for producing mappings prevents us from the possibility of implementing this new approach. This problem is clearly presented by (Sabou, 2006), (Motta 2006) and (Lopez, 2006), where authors argue the advantages of the adoption of such a knowledge base to improve the ontology mapping, that in this context we consider equivalent to enterprise applications integration. They go further and also claim that it is currently possible to obtain information from existing sources thanks to the fact that there is a reasonable amount of on line semantic data. Supporting this idea, we have investigated the B2B domain to study its particularities. We have also enquired the feasibility of gathering most on-line resources available and organizing them in a reference ontology.

The aim of this paper is to provide the analysis of the B2B use case for the Semantic Web; to present Janus, the tool that we have developed in order to retrieve semantic information from existing “ontologies”; and the results obtained by the application of Janus on a collection of 23 B2B XML based standards freely available on the Web.

We will start, in Section 2, with the presentation of our B2B use-case, showing current approaches to business exchanges. Section 3 presents a first look at Janus, and some of its results. In Section 4, we discuss related works. Section 5 is a conclusion.

2. THE B2B USE CASE

In this section, we present the B2B use-case, and advocate the use of ontologies to solve integration problems.

2.1 Why we need semantics

The book by Gregor Hohpe (Hohpe, 2003) clearly shows that there are many problems with application integration. He provides an exhaustive list composed of 65 patterns to be considered when building a system able to manage the whole process of application integration, basing his approach on a messaging system. In this paper we do not address the whole process of integration, but we focus on the content of messages exchanged between enterprise applications.

B2B 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… yet no existing approach implements techniques based on semantics! Currently, applications exchange information on the basis of passing parameters or data, formatted according to strict, pre-defined syntaxes. We define this approach as the *exactness method*. This method has the advantage of allowing total error management, except application bugs of course, but leaves no space for data interpretation. In consequence, reasoning on data of this type is virtually impossible because of the limits of its definition.

As asserted below, most interactions between B2B applications are implemented by interfaces based on standard messages defined by several consortiums and it appears that standardization organizations are often organized by business area. Thus to create electronic connections with different industry partners, as real life requires, means that we need a new application layer for each partner and a new design every time a new partner joins the collaboration on the fly, with the objective of integrating information describing the same set of concepts, but with different uses.

2.2 Business Exchange Approaches

As far as we know, current approaches to message content definition for electronic business exchanges are based on three types of solutions, which are:

Ad-hoc solution - The format is defined multilaterally during the design time phase of the application. This system shows some kind of "flexibility", in the sense that every time a new design is carried out, it does not present specific constraints. This flexibility on the other hand clearly shows a low degree of reusability and integration with new partners;

Proprietary solution - The format is decided unilaterally (e.g. by a main contractor in cooperation with small businesses, such as a big retail group and its suppliers). The 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;

Adoption of standards - The format is defined by a consortium. It has the advantage of guaranteeing a certain level of compatibility, durability and reuse of past experiences. The negative point is that it is a standard, so it requires a tremendous standardization effort and moreover, quite often several standards coexist in the same sector, which implies the need to implement multiple standards which in most cases are not compatible.

As shown in the European e-business report (E-Business W@tch, 2007) at least three enterprises out of four that realize business exchanges with partners, declare implementing applications based on B2B standards solutions (at least for Europe). Moreover, the authors of this report also state that the broad adoption of XML based standards in combination with web services, could become the key to shape electronic business transactions between enterprises in the future.

Our experience shows we can at least confirm that XML Schema is the most widely supported solution by consortiums and it is becoming the *de-facto* standard document format. It has overtaken other formats like the "old" EDI and the "new" RDF/OWL. In fact, in our research we have investigated more than 30 B2B standards, that are all XML Schema based. Only cXML (http://www.cxml.org/) provides a DTD based standard, and no RDF/OWL format is officially provided by any consortium.

2.3 The Canonical Data Model

Gregor Hohpe (Hohpe, 2003) suggests building a Canonical data model in order to minimize dependencies from different data formats, but he does not explain *how* to build it. We suggest adopting an ontology system when building the canonical data model, specifically for messages and using semantic web technologies to improve application integration. This approach is quite different from other experiences in the e-business domain, such as (Corcho, 2001), because it targets message definition rather than a thesaurus: a message is not a well defined hierarchical set of products. This means that a message meets a specific request, which is not always the same for different standards. This practice complexifies the matching of two messages, and therefore application integration, because standards can develop them with different pieces of information.

In other words, 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. Correlating these messages with common concepts is still a missing part. For this reason we suggest a procedure to construct a mapping between messages with the help of ontology based semantic web technologies. Figure 1 depicts the procedure of such a mapping, which is composed by the following steps: 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 best fit the requirement and finally; 5) translate.

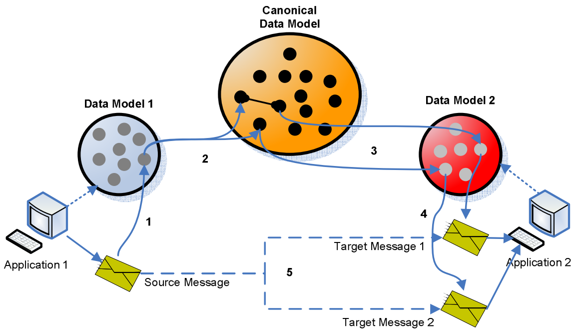

Figure 1 - Messages translation procedure

As we can see, the main problem is building the canonical model. The difficulty is that the classical development of a domain ontology, typically entirely based on strong human participation, does not adequately fit this use case, because it needs a more dynamic and automatic ontology building system, in order to be able to integrate new business partners on the fly. Also such a knowledge base must be able to serve as background knowledge for messages or services mappings.

3. AUTOMATIC CONSTRUCTION OF THE TAXONOMY

In this section we present Janus, a tool we have developed that manages information extraction from XML schema files. We also present the firsts results obtained from the automatic construction of a B2B taxonomy.

3.1 B2B Corpus Source

For this experience we have investigated more than 30 B2B standards, but not all are freely available and require membership fees (these have not been studied during the tests presented here).

As explained in Section 2.2, of all the freely available standards, only one of them is not in the form of XML Schema files describing business messages and none produce an OWL ontology. For this reason we decided, at least for the prototype, to consider only those standards offering XML Schema files and to focus our efforts on information retrieval specifically in this format. In fact XML Schema provides the great advantage, in respect to textual corpora, to define a structure for elements (candidate concepts for the ontology) notably limiting the difficulties of natural language interpretation. However as we show below, these documents introduce some noise at semantic level that needs special attention in order to provide good quality results.

Almost all organizations provide a package containing several XSD files, one for each specific message, one for grouping common data, others for grouping common data type definitions and code lists. At the end we get a corpus source composed of a collection of 23 standards (listed in Table 1), with more than 2000 XSD files that has been considerer enough in order to have significant information about B2B business message definition practices and semantics. Others standards can be added in future in an incremental way.

3.2 Janus: Taxonomy Builder Tool

Our tool implements an adaptation of several techniques originating from the text mining and information retrieval/extraction fields, applied to XML files (that we call **XML Mining**), in order to pre-process simple and compound terms from XML tags, such as XSD elements and XSD complex types. In reality our tool goes further in trying to build a reference ontology, making the hypothesis that each standard's set of files provides enough information to be considered an ontology itself.

Figure 2 shows the overall architecture of Janus. Currently the firsts steps of corpus discovery and clustering is hand made by taking advantage of the natural subdivision of B2B standards in business areas. Also this approach permits us to better understand the feasibility of translations between different standards measuring the "distance" between them. In the future we aim at crawling the net and implementing a TF-IDF measure for clustering documents.

Let us now detail the algorithm for term extraction and automatic taxonomy construction from XML tags :

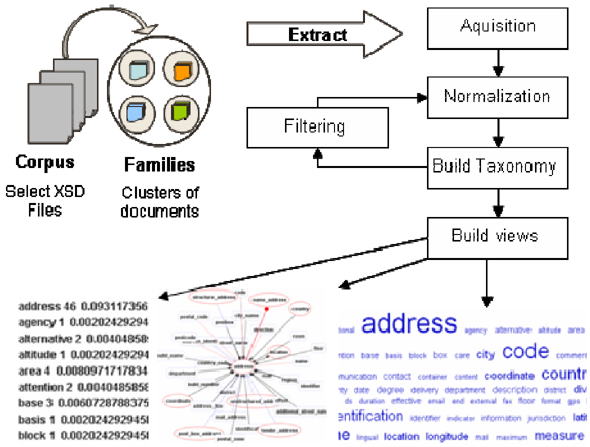

Figure 2 - Janus overall architecture

Acquisition Step

The aim of this step is to organize the corpus source and to select useful terms for the taxonomy.

The extraction tasks are:

1. XSD parsing and extraction of XML tag values for complex types, elements and simple types.
2. Checking for composite words (e.g.: on-line)
3. Checking for previously identified "useless" words, like systematic addition of unrelated semantic sense to the tag (e.g.: *CommonData* for *UnitOfMeasureCommonData*).
4. Splitting compound terms forming the tag, using the UCC convention, or '_' or '-' as separators, taking careful of special cases (e.g.: *PersonIDCode* = person + id + code).
5. Checking for known abbreviations (e.g.: Addr = Address, PO = Purchase Order)

As output to this step we produce a set of extracted tags for each family in the form: Term$_1$_Term$_2$_..._Term$_X$ (ex.: *ABIEPostalAddressType* that becomes *ABIE_Postal_Address*)

Normalisation Step

At this step the machine is not able to say if a term composing a tag is a real term or something else (abbreviation for example). Thus in order to compute semantic similarities between tags and to cluster them better, we add the use of a dictionary as external resource in order to be able to say if a term is a real human word or not. In our case we have integrated WordNet version 3.0 (Miller, 1995). Tasks for this step are:

1. *Case normalisation*, all terms are converted to lower case;
2. *Stop-word normalisation*, removes words like "of", "a", "for",…;

3. *Bad words detection*, terms unknown by the dictionary are cast aside;
4. *Morphological and semantic normalisation*, which consists in finding the stem and lemma form.

Build Taxonomy Step

The aim of this step is to create a first level of semantic relationships and hierarchy between words of the taxonomy.

1. *Calculate Terms Frequencies*
2. *Synonyms Check*, applied to words belonging to the taxonomy itself.
3. *Recompose tags.* All tags are recomposed using their lemma in order to be able to detect similarities between terms (thus between tags, thus between concepts of the ontology that we are building).
4. *Build Tags Lattice.* Tags are usually composed by more than one word, thus: we build a graph, based on Galois lattice, to relate those tags having the same words (ex. *address* and *postal_address*); we calculate the frequency of graph nodes and; we remove the nodes that are insignificant (values below a threshold)

Filtering Step

In this step we analyse the words that rejected by a first pass and we try to detect false semantics present within a tag.

1. *Bad words "reconciliation"*. At this time we try to detect as many abbreviations as possible applying a modified version of the N-Gram algorithm and Levenstain distance, to terms that already exist within the taxonomy. We restrict ourselves to terms within the taxonomy, because if we used the complete dictionary, we would detect too many similar terms, most of them out of context.
2. *Useless words detection*. Using the lattice we try to detect automatically those words that present disproportionate relationships between graph nodes (like *Type* or *CommonData*), and therefore do not convey any semantics in reality.
3. *Finalize.* Integrate new terms.

Build Views Step

We have implemented some visualization methods to view our taxonomy, at this point. Right now we have implemented the following views: as list, as tags lattice (with synonyms relationships) and as tag cloud. Others, like "Social Network of Word", are under development.

3.3 Results

Table 1 resumes the collection of B2B standards and some information about their declared relationships with other organizations. This table also resumes for each standard body the following information: number of XML Schema files that they provide (or in some cases, just those files that we have considered), the total number of complex type and element tags, the resulting number of "semantically" different words and; since XML tags can be composed of real dictionary words, mere abbreviations, or simply any sequence of characters, the last column provides the number of words unrecognised by the system.

More detailed Tables are provided in appendix. These tables show several aspects regarding current B2B business standards. On one hand they highlight some XML schema definition practices by standardization bodies, such as the use of anonymous types for elements, rather than declared types (elements without types); the adoption of Upper Camel Case or hyphen for tags to separate compound words (which is what we implement); the trend that financial and related bodies often use abbreviations rather than real terms for tags whereas standardization bodies mainly use common words for tags. Therefore it is possible to define a common taxonomy for the B2B domain. In fact, as shown in Table 2 and Figure 4, by adding one standard at a time, even in a random order, we have observed that after half a dozen of additions, less than 20% of the words are really new, to obtain about 9% new words in the last standard to be added. We have noted that these words usually represent terms characterizing the standard, but that the other, more general terms are already present in the global dictionary. Also we have observed that 60% of words are shared between standards, 11% of words are used by more than 10 of them and that this trend increases if measured over tags. So it shows that a dynamic taxonomy like this evolves easily and that a shared vocabulary emerges naturally.

We obtain 70976 tags, of which after normalization about 20000 are distinct. The total number of different words composing them is only 2887. On average, standards share three words over four. For example, *PostalAddress* is a tag, composed of 2 words. *PostalAddressTown* is a tag composed of 3 words. A standard composed of these two tags (normalized elements) would have 5 words, of which 3 are different (Postal, Address and Town). A tag called *PostAddrTwn* would be the same normalized element as *PostalAddressTown*.

Table 1 - Presentation of involved B2B standard and of the correspondent extraction of XML semantics

Standard Body	Business Area	Alliances	Files	Tags	Dictionary words	Unknown words
ACORD	Insurance, reinsurance and related financial service	X12, XBRL, HR-XML	8	5263	1162	657
AdsML	graphics communication		14	737	301	10
AgXML	Agriculture supply chain	ebXML, CIDX, RAPID	11	808	216	4
ARTS	Retail		44	5853	734	44
CIDX	Chemical	ebXML, RAPID	61	1881	437	20
ebXML	Cross industry		74	1401	408	10
ebInterface	Invoice		1	105	66	6
ETSO	Specific electric transaction	ebXML	1	27	32	0
FIX	Mainly banks, broker-dealers, exchanges and institutional investors	SWIFT (ISO 20022), FpML	18	552	117	93
FpML	Financial	FIX, FIXML	21	2124	544	34
GS1	Supply chain for Healthcare, Defence, Transport & Logistics	ebXML	289	2360	216	8
HR-XML	Human Resource	ACORD	166	12717	949	71
IFX	Financial		310	4256	446	249
ISO20022	Financial	IFX, OAGIS, TWIST	74	11082	256	384
MISMO	Residential, commercial, eMortgage	IFX, ACORD, ASC X12	14	1432	252	26
OAGIS	Cross industry	ebXML	515	4584	704	170
OTA	Tourist		233	3649	552	67
PapiNet	Paper		42	1394	530	18
PIDX	Petroleum	ebXML, CIDX	26	745	341	9
STAR	Automotive retail	OAGIS, ebXML	181	5518	1130	88
TWIST	Supply chain, payment	FpML, FIX, SWIFT	18	2489	457	184
UBL	Invoicing, ordering	ebXML	11	650	274	10
X12	Cross industry		9	1349	271	23
		Sum*:	2141	70976	10395	2185

* This sum value does not consider eventual correspondence of common tags or words between different bodies, for this take a look at table 2 below

3.4 Special Concern for "Bad Words"

As Table 2 shows, a considerable number of unrecognised words still remain, at least at first sight.

The analysis shows that these bad words are of the following type: mostly abbreviations (about 50%); about 30% are compound words not split by the system (for example compound words not written in UCC form like *worktime* or *preowned*); about 10% are words not included in the dictionary; and another 10% are acronyms.

Several techniques can be implemented in order to improve the detection of hidden words. Our implementation of abbreviation discovery is able to detect more than 70% of them automatically, which in reality corresponds to 80% of total occurrences (for example *amt* => *amount* has 958 occurrences thus more important than *liquidityfeature* with just one occurrence). Improving these results means (a) adopting a more complex management of abbreviations in order to detect different words having the same abbreviation, (b) implementing NLP techniques in order to mine text documents that often come with XML files and; (c) improving the external dictionary's capabilities.

Therefore we can say that solutions that provide good precision and recall exist, but in order to fully exploit the potential of semantic technologies, source document should be somehow *semantically well formed*. No semantic application will be able to understand the sense behind tags such as *AmortMktValDiffPct* or *setr.100.101*.

The adoption of XML based standards has already notably improved this opportunity, made this issue more apparent and has accelerated the drive towards convergence, as testified by the numerous alliances between standard bodies (see Table 1). Another improvement in this direction should be to exploit the structural content of XML files. Rather than using tag name with abbreviations for indicating structural relations like *PostAddrTwn* (11

Tableau 2 - Results from the families terms merging

Standard Body	Words	Dictionary Words	Addition %
X12	271	271	100,00
UBL	274	436	60,22
OAGIS	704	851	58,95
ACORD	1162	1539	59,21
GS1	216	1564	11,57
FIX	117	1606	35,90
ARTS	734	1828	30,25
FpML	544	1982	28,31
ETSO	32	1983	3,13
CIDX	437	2042	13,50
OTA	552	2157	20,83
IFX	446	2251	21,08
ISO 20022	256	2279	10,94
TWIST	457	2305	5,69
HR-XML	949	2468	17,18
ebInterface	66	2472	6,06
AdsML	301	2497	8,31
ebXML	408	2529	7,84
PapiNet	530	2678	28,11
PIDX	341	2701	6,74
STAR	1130	2844	12,65
AgXML	216	2864	9,26
MISMO	252	2887	9,13

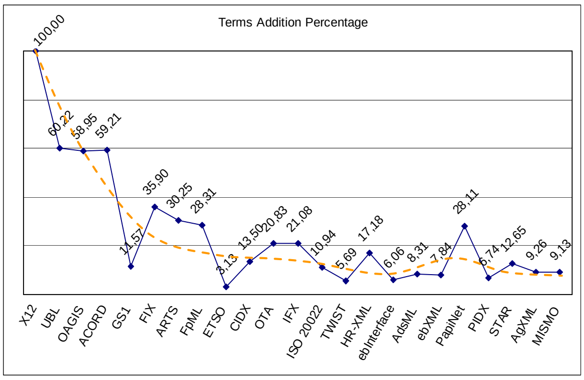

Figure 3 - Graph of sequential of terms addition (measures are in percentage)

chars) using simply *Town* (4 chars) as sub-element of *PostalAddress* should be enough for a machine to understand that town is a propriety of the address concept. A positive side effect is the economy of physical space.

4. RELATED WORK

Our work is related to several research domains. For work closer to B2B we can cite an interesting experience by Zaho and Lövdahl (Zaho 2003b), that provides an approach to develop ontology for Internet commerce by reusing XML-based standards. They also define layers and relationships of the common vocabulary as shared in the following parts: Core, General, Reusable and Special. But they do not go any further and do not provide concretely a taxonomy. Gloria Giraldo and Chantal Reynaud (Giraudo, 2002) have developed a semi-automatic ontology generation software for the tourism industry domain extracting information contained in DTD files. This experiment is really close to our use case but is limited to the sole domain of tourism, which is defined in advance with great precision, and therefore the detection of relevant concepts does not produce conflicts between different representations.

Other experiences that try to mix semantic integration and B2B taxonomies were developed by (Fensel, 2001b) and (Corcho, 2001), but their work was limited to catalogues of products like UNSPSC and eCl@ss, which have hierarchy and semantics well defined. In practice, the goal is the mapping of two taxonomies rather than the construction of a new one.

For more related semantic integration the document by Noy (Noy, 2004) provides an exhaustive list of experiences where our tool should be effective in terms of construction techniques, but they mainly target the merging of two input sources at a time sources.

Concerning the automation process of taxonomy and ontology generation in (Bedini, 2007) is shown that solutions implementing an automatic method for such a task are rare. We do not have the room to detail this here.

Finally, the construction of reference ontologies, the experience of D'Aquin et al. (D'Aquin, 2007) is significant for our work, but they does not consider XML Schema sources

5. CONCLUSION AND FUTURE WORK

In this paper we have presented our starting point for building B2B applications in agreement with the "Next Generation Semantic Web Applications" as described in (Motta 2006).

Despite the great amount of XML files available, current tools and software are only able to extract semantics from text corpora, or ontologies: tools providing the analysis of a consistent group of XML files are rare, and none really exist in the B2B domain.

We have thus developed Janus, a tool capable of extracting valuable semantic information from such corpora and have demonstrated its results with the automatic construction of a B2B taxonomy.

Although these results are encouraging, it is clear that our system does not yet offer enough to build a canonical data model for the B2B use case, nor does it reduce application integration to an automatic task. We plan on continuing this work with the development of a more complete tool, capable to associate semantic concepts to discovered taxonomy's terms in order to build as automatically as possible a reference ontology for the B2B domain.

REFERENCES

N. Anicic, N. Ivezic, A. Jones, 2005. *An Architecture for Semantic Enterprise Application Integration Standards*. In proceedings of INTEROP-ESA 05, Geneva, Switzerland

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

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

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

Y. Zhao, K. Sandahl, 2003a. *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

Gregor Hohpe, Bobby Woolf, 2003. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, October 2003. ISBN13:9780321200686 ISBN10: 0-321-20068-3

C. Welty. *Ontology Research*. AI Magazine, 24(3), 2003

Jean Charlet, Bruno Bachimont and Raphaël Troncy, 2004. *Ontologies pour le Web sémantique*. In Revue I3, numéro Hors Série «Web sémantique», 2004.

Asuncion Gomez Perez and V. Richard Benjamins, 1999. *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*. IJCAI-1999, Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends

Marta Sabou, Mathieu d'Aquin, and Enrico Motta (2006) *Using the Semantic Web as Background Knowledge for Ontology Mapping.* In Proc. of the International Workshop on Ontology Matching, collocated with ISWC'06

Vanessa Lopez, Marta Sabou and Enrico Motta (2006) *PowerMap: Mapping the Real Semantic Web on the Fly*. In Proc. of the 5th International Semantic Web Conference (ISWC'06), Athens, GA, USA.

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. January 2007. (http://www.ebusiness-watch.org)

N. Noy, 2004. *Semantic integration: a survey of ontology-based approaches.* SIGMOD Record, Vol. 33, No. 4, December 2004.

Miller, G.A. (1995). WORDNET: A lexical database for English. Communications of ACM (11), 39-41.

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

Gloria Giraldo, Chantal Reynaud, 2002. *Construction semi-automatique d'ontologies à partir de DTDs relatives à un même domaine.* 13èmes journées francophones d'Ingénierie des Connaissances, Rouen

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

Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, Marta Sabou, and Enrico Motta, 2007. *Watson: A Gateway for Next Generation Semantic Web Applications*. Poster session of the International Semantic Web Conference, ISWC 2007.

Enrico Motta and Marta Sabou, 2006. *Next Generation Semantic Web Applications*. In Proc. of the 1st Asian Semantic Web Conference (ASWC), Beijing, China, 3-7 September, 2006

Ivan Bedini and Benjamin Nguyen, 2007. *Automatic Ontology Generation: State of the Art*. Technical report, University of Versailles (http://194.199.139.28/RepXMLWeb/servlet/DownloadServlet?fileName=Automatic_Ontology_Generation_State_of_Art.pdf)

APPENDIX

The first Table below shows a detailed view of global extracted information from B2B standards XML Schema documents.

Std Body	SW	W	Add%	SBW	BW	Add%
X12	271	271	100,00	23	23	100,00
UBL	274	436	60,22	10	33	100,00
OAGIS	704	851	58,95	170	200	98,24
ACORD	1162	1539	59,21	657	845	98,17
GS1	216	1564	11,57	8	852	87,50
FIX	117	1606	35,90	93	921	74,19
ARTS	734	1828	30,25	44	960	88,64
FpML	544	1982	28,31	34	990	88,24
ETSO	32	1983	3,13	0	990	0,00
CIDX	437	2042	13,50	20	1008	90,00
OTA	552	2157	20,83	67	1052	65,67
IFX	446	2251	21,08	249	1134	32,93
ISO20022	256	2279	10,94	384	1372	61,98
TWIST	457	2305	5,69	184	1396	13,04
HR-XML	949	2468	17,18	71	1435	54,93
ebInterface	66	2472	6,06	6	1438	50,00
AdsML	301	2497	8,31	10	1444	60,00
ebXML	408	2529	7,84	10	1448	40,00
PapiNet	530	2678	28,11	18	1463	83,33
PIDX	341	2701	6,74	9	1469	66,67
STAR	1130	2844	12,65	88	1505	40,91
AgXML	216	2864	9,26	4	1508	75,00
MISMO	252	2887	9,13	26	1522	53,85

Legend:

SW – Standard body Words. Indicate the number of dictionary words for each standard body.

W – Words. Indicate the number of real different normalised words that constitutes the terms for the B2B taxonomy.

SBW – Standard body Bad Words. Indicate the number of unrecognised words (or sequence of terms) for each standard body.

BW – Bad Words. Indicate the number of real different unrecognised words for the global extraction

Add% – Addition Percentage. Indicate the percentage of words/bad words really added to the dictionary.

Table below shows a detailed view of extracted information from B2B standards XML Schema documents for each standard body without considering overlapping common words and tags.

Std. Body	Files	Elt	Norm	WType	Words	BadW	StopW	CT	Norm	Words	BadW	StopW	Tags	Norm	Words	BadW	StopW
Acord	8	4164	2741	10	1154	211	61	1099	531	401	498	24	5263	2827	1162	657	62
AdsML	14	593	484	125	289	8	27	144	124	118	3	9	737	559	301	10	28
AgXML	11	540	367	2	216	4	15	268	183	129	0	8	808	368	216	4	15
Arts	44	4562	1318	1069	727	41	39	1291	423	316	18	18	5853	1445	734	44	43
CIDX	61	1078	932	1	437	20	29	803	678	324	8	23	1881	932	437	20	29
ebXML	74	1088	553	0	404	10	15	313	190	170	4	7	1401	566	408	10	15
ebInterface	1	75	65	0	65	6	5	30	26	32	2	3	105	67	66	6	6
Etso	1	0	0	0	0	0	0	27	26	32	0	2	27	26	32	0	2
FIX	18	333	37	3	48	83	9	219	74	109	50	9	552	109	117	93	12
FpML	21	1450	1042	0	509	32	37	674	573	328	18	20	2124	1242	544	34	44
GS1	289	1586	315	114	212	6	16	774	146	106	4	5	2360	358	216	8	16
HR-XML	166	10103	2089	1567	927	60	69	2614	620	401	32	24	12717	2302	949	71	70
IFX 170	310	2925	650	0	420	248	36	1331	217	175	73	15	4256	688	446	249	36
ISO 20022	74	8244	86	0	73	383	26	2838	313	205	7	16	11082	391	256	384	29
Mismo	14	719	266	30	251	22	18	713	423	252	25	17	1432	617	252	26	18
OAGIS	515	2919	1551	0	677	166	38	1665	836	328	16	16	4584	1734	704	170	40
OTA	233	3153	1042	1610	541	67	22	496	376	277	22	14	3649	1159	552	67	24
PapiNet	42	1316	1193	786	528	18	41	78	75	89	1	11	1394	1246	530	18	44
PIDX	26	705	644	256	341	9	21	40	38	47	0	6	745	652	341	9	21
STAR	181	4214	3191	0	1113	88	67	1304	893	420	17	22	5518	3308	1130	88	67
Twist	18	1929	911	159	431	175	46	560	319	211	23	16	2489	1039	457	184	48
UBL	11	441	370	0	267	9	5	209	180	160	5	2	650	382	274	10	5
X12	9	727	329	18	252	22	14	622	118	96	4	5	1349	438	271	23	15
Sum* :	2141	52864	20176	5750	9882	1688	656	18112	7382	4726	830	292	70976	22455	10395	2185	689

* This sum value does not consider eventual correspondence of same tags or words between different bodies, for this look at table 2.

Legend:

Files - Files. Indicate the number of files from which tags has been extracted for each body.

Elt - Element. Indicate the number of defined XML tags of XSD elements for each body (ex.: <xsd:element name="Location" type="LocationType"/>)

CT - Complex Type. Indicate the number of defined XML tags for XSD Complex Types for each body (ex.: <xsd:complexType name="LocationType"/>)

Tags - Tags. Indicate the sum of the XML Elements and Complex Types for each body.

Norm - Normalised. Indicate the number of real different tags after the normalisation task of each tag (ex.: “AttentionOfName” = (norm) => “attention_name”) .

WType - Without ComplexType. Specifically for XML Elements to indicate the number of without type declaration (also known as “orphans elements”). (ex.: <xsd:element name="Location"/>)

Words - Words. Indicate the number of real different terms used for defining XML tags after the normalisation step.

BadW - Bad Words. Indicate the number of terms that are not recognised as real existing dictionary terms (e.g. abbreviations and acronyms). (ex.: endrsmnt => endorsement)

StopW - Stop Words. Indicate the number of terms that are recognised as terms without relevant semantics sense for the tag (e.g. the, of, a, with…)