Improving B2B Transactions by Exploiting Business Registries

Manjith Gunatilaka, Gihan Wikramanayake, Damitha Karunaratna
University of Colombo School of Computing
manjith@dmsswt.com, gnw@ucsc.cmb.ac.lk, ddk@ucsc.cmb.ac.lk

ABSTRACT

One of the essential requirements in electronic business is to exchange business information between agreed business partners. However discovering suitable partner(s) and establishing collaboration with these partners is a problem faced by many current e-commerce infrastructure models. Therefore effective implementation of a strategic B2B communication between business partners by proposing a feasible solution to overcome existing concerns/problems is one of the main objectives that we need to address in order to develop the existing Info-Structure concept.
In this research paper we are proposing how we can improve the storage of business domain knowledge and effective searching of business services stored in registries utilizing the implementation of Ontologies. We have selected "IT Outsource" as our example Ontology which helps IT companies engaged in outsourcing to setup their business repositories in an effective manner. Therefore within a short period of time customers can effectively search business services advertised in these registries.

1. INTRODUCTION

The Electronic Business XML (ebXML)[1] and Universal Description, Discovery and Integration (UDDI)[2] are the commonest business infrastructure frameworks available in the market. All of these frameworks implement business registries to advertise business services. These registries are domain specific and it publishes business information relevant to particular business domain. In real world there are a number of business registries such as Adobe – eForm registry [3], SDMX - statistical registry [4], Apelon – Medical guideline registry [5], General Motors and Vokswagon B2B artifacts registry, IBM UDDI business test registry [6], etc.

Ontology is a description of a content and relationship expressed as a formal vocabulary. This can be used to classify domain specific information in different areas such as in education, engineering, natural sciences etc. Most of the business information stored in repositories in Business to Business (B2B) infrastructure frameworks is difficult to extract due to unstructured domain classification in business registries. Therefore how we can improve these business structures (Business frameworks) is an area most researchers are trying to address.

Use of Ontology as an information source to formalize domain knowledge and finding a way of mapping this information to registry content may be a practical solution which will enhance B2B communication and hence Info-Structure facilities. But today the use of Ontological classifications inside the registry is not used by any of these registries. Therefore registries implement this knowledge using coding schemes which is specific to its own implementation. During our research effort, we have selected OASIS [7] open ebXML registry implementation (freeebXML) [8] as an example registry. Web Semantic Content in the repository is represented using OWL [9] Ontology language.

2. IMPORTANCE OF REGISTRY CENTRIC OPERATION IN B2B TRANSACTIONS

Let's consider B2B transaction between two parties in point-to-point interface. This type of transactions needs sender and receiver to know detail knowledge about each others application for mapping values. Though information sharing is direct it is cumbersome due to non availability of standard format or automation. The other problem it has is the number of mapping between each party can be n(n-1)/2 if it has n number of end points. This is practically not a feasible solution.

The second type of sharing of information in B2B is based on "hub n' spoke" pattern [10] creating integration broker which passes messages and control information in a token. This integration broker is not an ideal solution because it is even worse when compared to the existing problems in the point-to-point solution. Alternative to these methods is the hybrid solution based on two of these approaches which provides optimal results and eliminates many problems existing in the point-to-point and hub n' spoke approaches.

Therefore this new solution or the third type of B2B transaction approach should be based on centralized and decentralized concepts where it maintain decentralized Web Services, queries etc. and make meta data (context) as centralized elements. This approach is based on Centralized Registry/Repository which keeps meta data of your organization in the specific domain. The Return on Investment (ROI) of an organization is based on Enterprise Agility (Business Process), domain knowledge and information services it offers. Therefore implementation of Business Registry/Repository concept will encourage any organization to move fast in the right direction and better knowledge of domain rather than sluggish movement or wasted motion .

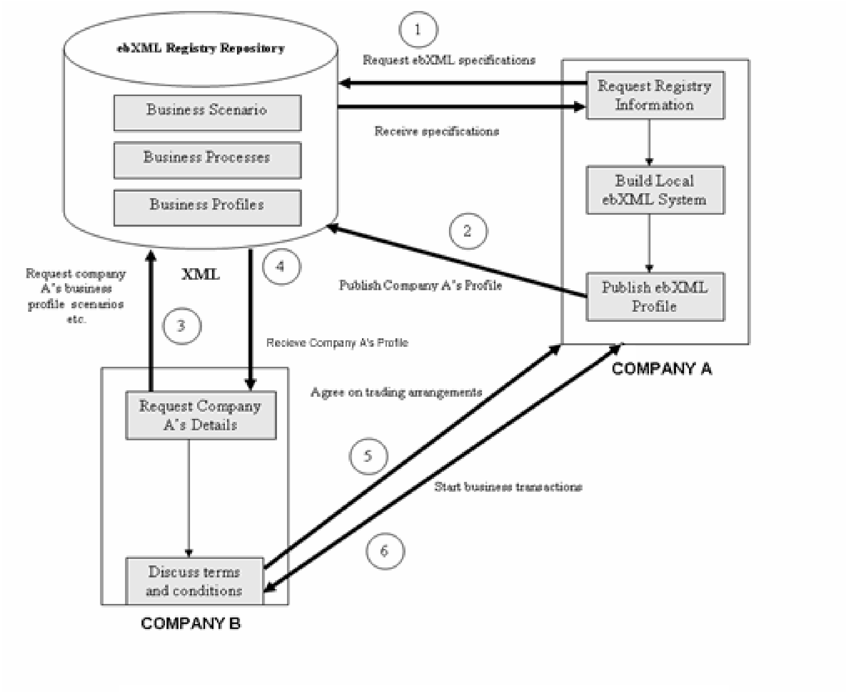
ebXML Registry Repository
Business Scenario
Business Processes
Business Profiles
1
Request ebXML specifications
Receive specifications
Request Registry Information
Build Local ebXML System
Publish ebXML Profile
2
Publish Company A's Profile
COMPANY A
Request company A's business profile scenarios etc.
XML
3
4
Recieve Company A's Profile
Request Company A's Details
Discuss terms and conditions
COMPANY B
Agree on trading arrangements
5
Start business transactions
6

Figure 1: ebXML interaction process

3. ebXML B2B TRANSACTION PROCESS

According to the diagram shown in Figure 1. Company A establishes its own (step 1 and 2) Business Processes, Business Scenarios, Business Specifications and Business Profiles in the ebXML repository. Similarly repository contains other companies' business profiles, business scenarios and specifications. Step (1) and (2) illustrates implementation phase of these Business Collaboration Protocol Profile (CPP) for any given company. Once Company A registers its business profile in the registry, it is available for any company to refer A's business scenarios, processes and specifications to start business with company A. Once A's profile is available in the repository, Company B will search for a suitable business partners (Discovery of partner information and negotiation phase) in the registry (step 3). Company B's ebXML compliant system is capable of extracting suitable partner(s) from repository according to business scenarios, business processes etc. specified in the repository. Company B extracts A's CPP (step 4) and analyze its compatibility with B's CPP. At this stage company B directly negotiate with Company A's (step 5) terms and conditions and generate Collaboration Protocol Agreement (CPA) to initiate business transactions. Once CPA is established, the two parties begin actual transactions (step 6).

4. ebXML BUSINESS REGISTRY

The comparison of registry/repository scenario can be a good example for a library system where borrower needs to find a book using a catalogue. Once relevant book is found it can be retrieved physically. Therefore books catalogue acts as a registry and ***discovery*** is a process to locate services advertised in the registry. Once service is found, it can be physically ***retrieved*** from the repository. All registered information stored in the repository is managed by registry services. These services are stored as metadata in the registry as Registry Objects which can be relate to any given instance.

The ebXML registry contents are classified according to Registry Information Model (RIM) [11]. The composition of registry structure is shown in class diagram given in Figure 2. According to the diagram all objects are represented by a root called ***RegistryObject*** which is an abstract class. It represents minimal meta data information in the registry. All domain hierarchies are grouped into ***ClassificationScheme*** (i.e. grouping of registry objects). It group RegistryObject into categories of tree structure. In our example business domain, "IT Outsource", "Industry Category" are several Classification Schemes. We have classified according to our own way of classification not based on standard classification schemes such as North American Industry Classification System (NAICS) [12] or ISO 3166 [13] etc. The reason not to select standard coding scheme was mainly due to lack of proper code sets currently available on IT domain and designing such a scheme is a cumber exercise. Therefore we have decided not to design and incorporate standard codes based on NAICS/ISO which is out of the scope in this research project. The ***ClassificationNodes*** represents tree nodes in ClassificationScheme. For example "Transportation", "Consumer electronics" are ClassificationNodes in "Industry" ClassificationScheme. The ***RegistryEntry*** provide metadata to the repository item. The ***Association*** class represents relationship between two registry objects in the Registry Information Model. These two registry objects are called source RegistryObject and target RegistryObject.

The ***Service*** instance represents published registry services. The interface to these services are implemented according to ebXML Registry Service Specification (ebRSS) [14]. These services are published as Web Services, service discovery and specification details using Web Service Description Language (WSDL).

5. SEMANTIC WEB CONTENT MANAGEMENT (WCM) AND ONTOLOGY

5.1 Relationship between Semantic Web and Ontology

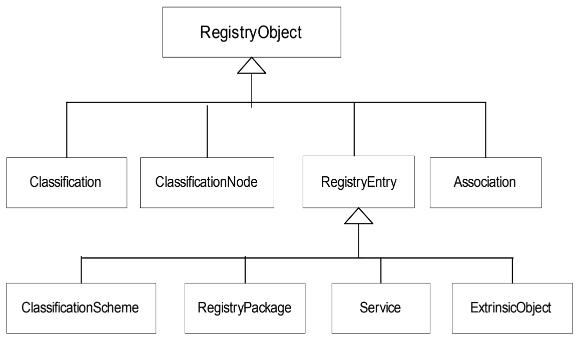

Figure 2: RIM Class hierarchy diagram

Semantic Web is an effective way of representing information in a link way that can be easily processed by machines on a global scale. The representation of this information can be achieved using various Web Semantic languages such as DAML+OIL [15], RDF [16], OWL etc. The Ontology is a way representing the link information in a structured way so that it can be shared/reuse among people or software agents. In other words it is an explicit specification of conceptualization [17]. This will help to separate domain knowledge in particular area from operational knowledge. In our research, we have selected OWL as Semantic Web language to represent "IT Outsource" Ontology. The creation of Ontology was carried out using Ontology development editor called Protégé [18] which is a popular Ontology development tool.

5.2 What is Semantic Web Content Management?

WCM is a mechanism to manage web site or repository contents remotely and linked, categorized and arrange the contents in a way we required. Currently in ebXML WCM is performed according to RIM specification. But representation of Ontology based classification mechanism is not supported in ebXML RIM model. Therefore it is important to transform Ontology based

OWL model into RIM model and map similarities shown in OWL and ebXML RIM. Once transformation is available information is stored according to converted RIM model which actually represents common Ontology of a selected domain such as "IT Outsourcing".

5.3 Defining "IT Outsource" Ontology using OWL

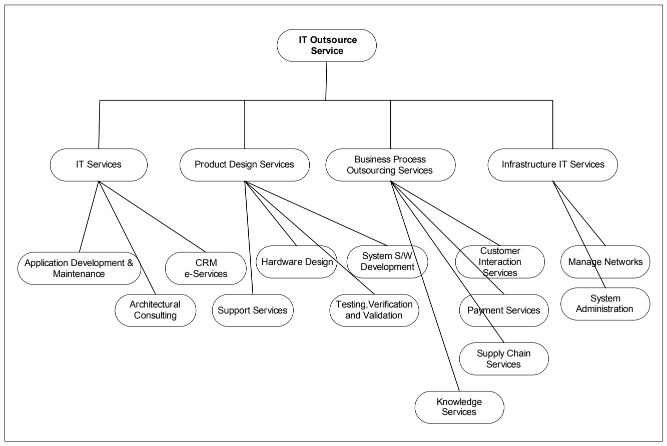

Figure 3: An Example "IT Outsource" Ontology Services Class diagram

Diagram shown in Figure 3 represents "IT Outsource" domain and its relationship with the Ontology. It represents main services in IT Outsource domain and this class relationship was converted to OWL using protégé tool and its code snippet is shown below as Figure 4.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#
"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns="http://a.com/ontology#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://a.com/ontology">

  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
    <rdfs:comment
rdf:datatype=http://www.w3.org/2001/XMLSchema#string>
         IT  Outsourcing Ontology
     </rdfs:comment>
  </owl:Ontology>

<owl:Class rdf:ID="PaymentServices">
   <rdfs:subClassOf>
      <owl:Class rdf:about="#BusinessProcess"/>
   </rdfs:subClassOf>
  <owl:Class rdf:ID="InfrastructureIT">
   <rdfs:subClassOf>
     <owl:Class rdf:about="#OutsourceServices"/>
   </rdfs:subClassOf>
   <rdfs:label>Infrastructure IT</rdfs:label>
 </owl:Class>

 <owl:Class rdf:ID="KnowledgeServices">
   <rdfs:subClassOf>
    <owl:Class rdf:about="#BusinessProcess"/>
   </rdfs:subClassOf>
 </owl:Class>
…………………..
…………………..
……………………
<owl:ObjectProperty rdf:ID="ITServiceIndustry">
   <rdfs:domain rdf:resource="#ITServices"/>
   <rdfs:range>
    <owl:Class>                    …………………. (4.1)
      <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Finance"/>
      <owl:Class rdf:about="#Government"/>
     </owl:unionOf>
    </owl:Class>
   </rdfs:range>
 </owl:ObjectProperty>

 <owl:DatatypeProperty rdf:ID="costSolution">
   <rdfs:domain rdf:resource="#Solutions"/>
   <rdfs:range                        ……………… (4.2)
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
……………………………..
…………………………….
<ERPSol rdf:ID="OracleFinancials">
    <costSolution
rdf:datatype=http://www.w3.org/2001/XMLSchema#string>
         US$ 5000 per user               ……………….. (4.3)
   </costSolution>

  <providerSolution>
    <Provider rdf:ID="DMS">
     <nameOrganization
rdf:datatype=http://www.w3.org/2001/XMLSchema#string>
              DMS Software Technologies
      </nameOrganization>
    </Provider>
   </providerSolution>

 <providerSolution>
    <Provider rdf:ID="MIT">
      <nameOrganization
rdf:datatype=http://www.w3.org/2001/XMLSchema#string>
             MIT
     </nameOrganization>
    </Provider>
  </providerSolution>
</ERPSol>
```

Figure 4: OWL Code for Sample Ontology

In OWL, there are three basic elements called *Classes, Individuals* (Instance of a Class) and *Properties*. The Classes describe concepts in the domain. For example, “ITInfrastructure”, “PaymentServices” and “Solution” are some of the example Classes in our selected domain. It is possible to define *Sub Classes* within a class to further classify the base Class. The second important element *Individual* describes is the specific instance of a Class. For example, in the “ERPSolution” Class, “*OracleFinancial*” is an instance. The third element *Properties* defines values that each *Individual* can take. For example, in “*OracleFinancial*” instance (individual) there are three types of Properties namely ObjectProperty, DatatypeProperty and *Individual Property*. These properties represent binary relation. The *ObjectProperty* (Figure 4, Section 4.1) represents relationship between two instances. The *rdfs:domain* and *rdfs:range* restrict the relationship by defining its domain and range Classes. The *DatatypeProperty* (Figure 4, Section 4.2) represents relation between instance Classes and RDF literals and XML Schema datatypes. The final type of Property, *Individual,* (Figure 4, Section 4.3) defines Instance specific XML Schema datatype based element.

6. MAPPING OWL INTO ebXML RIM

Figure 5 represents example “IT Outsource” Ontology in ebXML RIM class hierarchy format. During the analysis of OWL and ebXML RIM syntax what we gathered is, it has its own concepts in representing information. Therefore we need to map each representation into corresponding counterpart. In OWL *owl:Class* can be mapped as *ClassificationNodes* in the ebXML RIM.

```
<ClassificationScheme
    id="ITOutsource" isInternal="true"
    nodeType="UniqueCode"
    xmlns="urn:oasis:names:tc:ebxml-egrep:rim:xsd:2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
nstance" xsi:schemaLocation="urn:oasis:names:tc:ebxml-
regrep:rim:xsd:2.0 rim.xsd">

<ClassificationNode   id="ERP-id"
      parent="ITServices-id" code="ERP">
</ClassificationNode>
</ClassificationScheme>
```

All other RDF properties such as ObjectProperty, DatatypeProperty and Individual Property can be mapped as an *Association* in the ebXML RIM. The rdf:domain and rdf:range can be mapped as sourceObject and targetObject attributes in the <Association> tag. If the targetObject is a collection of registry objects which is represented as *rdf:parseType* then it should be mapped with the *RegistryPackage* instance in RIM.

```
<RegistryPackage id="IndustryPkg-id" >
  <Name> ……. </Name>
  <Description> …. </Description>
</RegistryPackage>

<Association id="ITServiceIndustryAsso-id"
    associationType="ITServiceIndustry"
    sourceObject="ITServices"
    targetObject="IndustryPkg-id"
</Association>
```

Similarly we transformed all OWL specifications into ebXML registry RIM schema specification. Once this transformation is completed new RIM schema can be stored in the ebXML registry. The process of transformation was automated using a Jena [19] Ontology parser and predefined OWL and ebXML RIM mapping table. These mappings are used as a reference in the conversion process. The process of transformation and the components used are shown in Figure 6. It also shows how registry can be accessed via Registry browser which is based on JAXR (Java API for XML Registry) [20] API.

The analysis we carried out on Ontology based language OWL and its representation inside the open business registry (free ebXML) can be automated using ***OWL-ebXML Onto*** transformation engine. This kind of plug-in can be developed for various business registries (ebXML. UDDI etc.) Therefore this can be considered as an effective software component used as a semantic web plug-in for Business Registries. This will improve content arrangement and the efficient retrieval of information by different business partners. Incorporation of Ontology based web semantic will make search process in the business registry automated and it can be delegated to Software Agents which act on behalf of human agent in the real world.

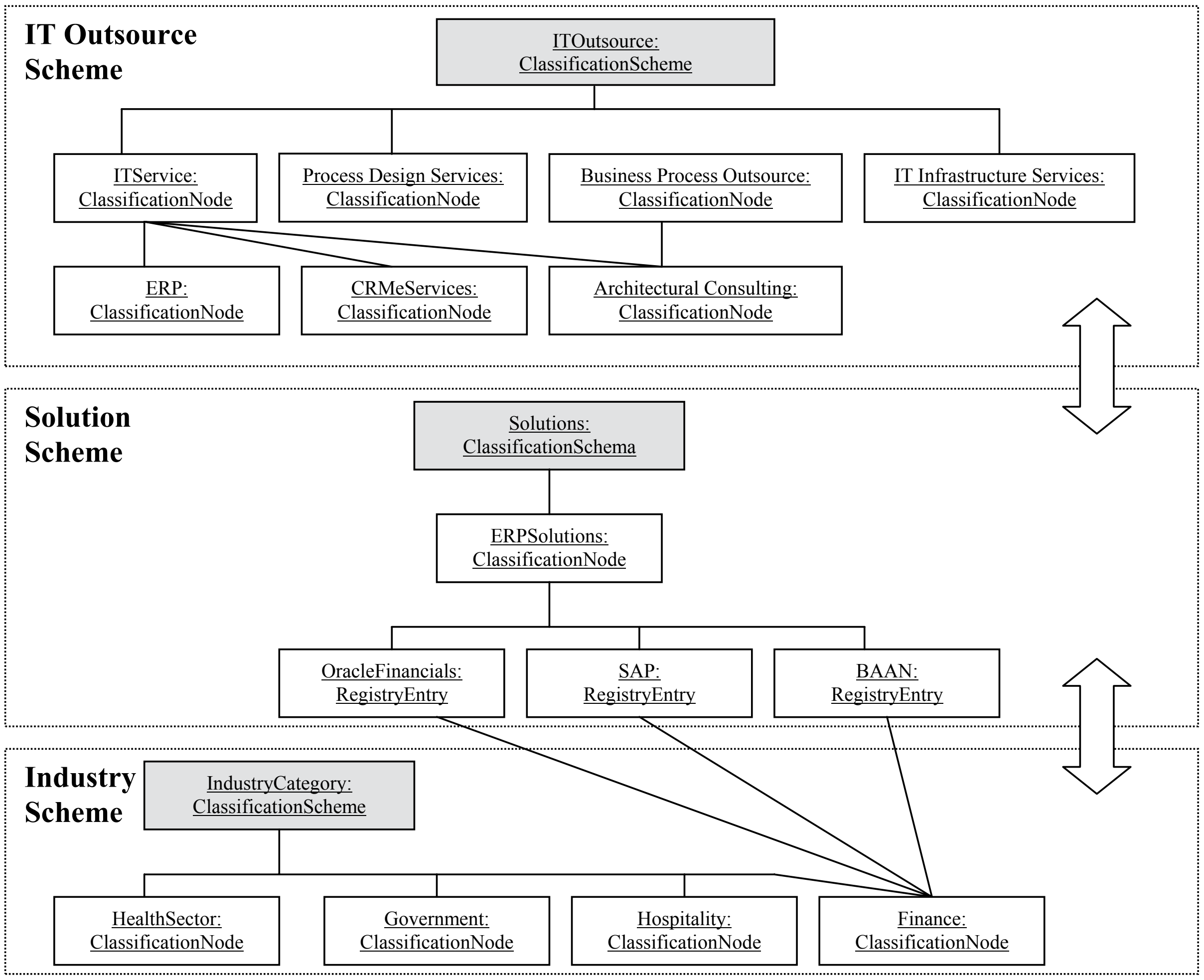
IT Outsource Scheme
ITOutsource: ClassificationScheme
ITService: ClassificationNode
Process Design Services: ClassificationNode
Business Process Outsource: ClassificationNode
IT Infrastructure Services: ClassificationNode
ERP: ClassificationNode
CRMeServices: ClassificationNode
Architectural Consulting: ClassificationNode
Solution Scheme
Solutions: ClassificationSchema
ERPSolutions: ClassificationNode
OracleFinancials: RegistryEntry
SAP: RegistryEntry
BAAN: RegistryEntry
Industry Scheme
IndustryCategory: ClassificationScheme
HealthSector: ClassificationNode
Government: ClassificationNode
Hospitality: ClassificationNode
Finance: ClassificationNode

Figure 5: Representation of IT Outsource Ontology mapping in ebXML RIM

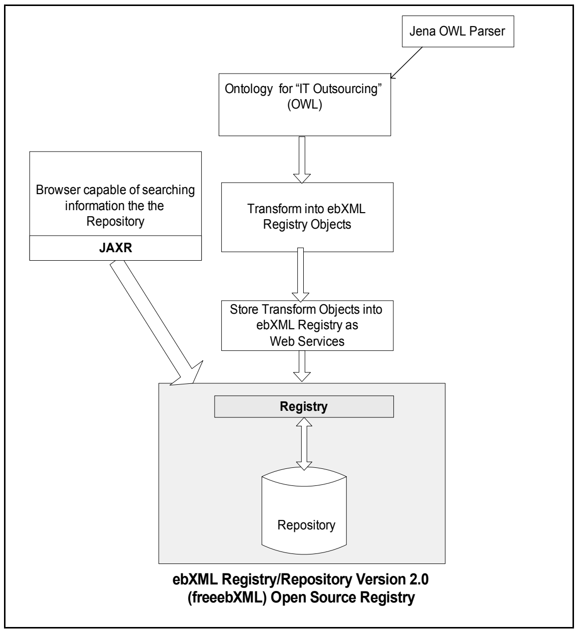
Jena OWL Parser
Ontology for "IT Outsourcing" (OWL)
Browser capable of searching information the the Repository
JAXR
Transform into ebXML Registry Objects
Store Transform Objects into ebXML Registry as Web Services
Registry
Repository
ebXML Registry/Repository Version 2.0 (freeebXML) Open Source Registry

Figure 6: "OWL-ebXML Onto" transformation engine and its implementation architecture

7. CONCLUSION

An effective business collaboration between two parties in B2B environment is an issue in info-structure models due to inadequacy and inability to represent business domain knowledge. Though there are few commercially available business repositories, storing business domain knowledge and formalizing business jargon was a problem. Once this knowledge was formalized (using Ontology) how to incorporate it to existing business registries is another problem.

During our research effort we addressed this concern by exploiting similarities in Ontology language OWL and example registry ebXML to propose a transformation method. Based on this method we discussed and proposed an implementation of a transformation engine called OWL-*ebXML Onto* which is an Ontology based semantic web content plug-in. The future work in Ontology and Business registry would be direct incorporation of Ontology based repositories, which will remove third party plug-in and make business repositories more organized and efficient. The concept of Ontology based registries will emerge in the near future which is similar to web servers. This will enhance Small, Medium and Enterprise (SME) business partners to setup there registry centric business operation more effective and efficient by improving existing bottlenecks in the B2B transaction process, which will strategically enhance the existing Info-Structure model.

8. REFERENCES

[1] ebXML,http://www.ebxml.org,[visited on 12/05/2004]
[2] UDDI, http://www.uddi.org, [visited on 12/05/2004]
[3] Adobe eForm, http://xml.gov/presentations/adobe/PDFA-XMP-Registry.pdf, [viited on 12/05/2004]
[4] SDMX , http://www.sdmx.org,[visited on 12/05/2004]
[5] Apelon – http://www.apelon.com, [visited on 12/05/2004]
[6] IBM UDDI registry – https://uddi.ibm.com/testregistry/find, [visited on 12/05/2004]
[7] OASIS - http://www.oasis-open.org/home/index.php, [visited on 12/05/2004]
[8] freebXML - http://ebxmlrr.sourceforge.net/, [visited on 12/05/2004]
[9] OWL language overview, http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3, [visited on 12/05/2004]
[10] David A. Chappell, Vivek Chopra, Bruce Peat, Colleen Evans, "Professional ebXML", 2001
[11] RIM - ebXML Registry Information Model Version 2.0, April 2002, http://www.ebxml.org/specs/ebRIM2.pdf, [visited on 12/05/2004]
[12] North American Classification System, http://www.census.gov/epcd/www/naics.html, [visited on 12/05/2004]
[13] ISO 3166 - http://www.iso.ch/iso/en/prods-services/iso3166ma/, [visited on 12/05/2004]
[14] ebRSS, http://www.ebxml.org/specs/ebRS2.pdf [visited on 12/05/2004]
[15] DAML+OIL, http://www.w3.org/TR/daml+oil-reference
[16] RDF, http://www.w3.org/RDF/, [visited on 12/05/2004]
[17] Thomas R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", August 1993, pp1
[18] Protégé, http://protege.stanford.edu/, [visited on 12/05/2004]
[19] Jena , http://jena.sourceforge.net/documentation.html, [visited on 12/05/2004]
[20] JAXR, http://java.sun.com/xml/jaxr/index.jsp, [visited on 12/05/2004]