行政院國家科學委員會專題研究計畫 成果報告

本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(2/2)
研究成果報告(完整版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 95-2416-H-004-006-
執 行 期 間 ：95 年 08 月 01 日至 96 年 07 月 31 日
執 行 單 位 ：國立政治大學資訊管理研究所

計 畫 主 持 人 ：諶家蘭

計畫參與人員： 此計畫無參與人員：無

報 告 附 件 ：國外研究心得報告
出席國際會議研究心得報告及發表論文

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 98 年 05 月 12 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告
□期中進度報告

本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(2/2)

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 95 – 2416 – H – 004 – 006 -
執行期間：　95 年　8　月　1 日至　96　年　7　月　31 日
計畫主持人：謝家蘭
共同主持人：
計畫參與人員：
成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立政治大學會計學系所

中　華　民　國　96　年　5　月　31　日

摘 要

隨著網際網路和企業內部網路的盛行，異質資訊整合成為電子化企業中一個重要的議題。在網路上進行異質資訊整合涉及許多不同新的資訊技術，目前已經有些研究試圖利用延伸標記語言以及本體論當作中介技術來整合異質資訊。為了有效管理企業內的資訊，我們需要一個績效評估模型來衡量異質資訊整合的效能。在本研究中，我們將提出一個在異質資訊整合中運用延伸標記語言及本體論的績效評估工作量模型，並且將建立系統雛形。本研究目的是希望發展出一個結合延伸標記語言及本體論的泛用型工作量模型，以測試在電子化企業中的異質資訊整合是否能整合不同的資訊模型，並且從這些資訊模型中衍生出語意。此工作量模型包含了延伸標記語言與本體論的資料模型與查詢模型，它們是依照延伸標記語言與本體論學名式的資料模式與查詢功能；此外，控制模型將定義績效評估執行環境中所需設定的變數，讓此工作量模型具有可攜性和延展性，以便應用在不同的領域情境中。最後，本研究採取學名結構式且使用者定義、領域獨立的方法、和雛形實驗來驗證本研究所提出的研究結果。

關鍵字：**延伸標記語言，本體論，異質資訊整合，績效評估，工作量模型**

Abstract

With the popularity of Internet/Intranet, heterogeneous information integration becomes a hot IT topic in electronic business (EB) field. Heterogeneous information integration on the Web involves a number of new techniques. There have been research projects applying XML and ontology as mediated techniques to consolidate heterogeneous information. In order to manage and use information more effectively within the enterprise, a benchmark used to evaluate the mechanism of heterogeneous information integration is needed. In this research, we develop a XML and ontology benchmark workload model in heterogeneous information integration, and build a workload generation prototype. The objective of this research is to develop a workload model combines XML and ontology to test whether the heterogeneous information integration system under EB environment can overcome the diverse formats of content and derive meaning from this content. The workload model consists of XML and ontology data model and query model according to the generic data structure and query functionality. Also, a control model is created to set up the benchmark environment. In order to apply the workload model to different scenarios easier, this workload model is designed to be domain independent and generic-construct-based. Finally, we validate the research model through the prototype implementation.

Keywords: XML, Ontology, Heterogeneous Information Integration, Benchmark, Workload Model, Performance Evaluation

Table of Contents

1. Introduction

1.1 Research Motivation

In the past decades, World Wide Web (WWW or Web) has changed everything, especially business computing. More and more enterprises adopt Internet/Intranet business model, and large amount of business information are exchanged over the Internet everyday. But data sources on the Web are often distributed and heterogeneous. Enterprises have to spend more time and efforts searching the data they need. Recently, electronic business (EB), enterprise information integration (EII), and enterprise application integration (EAI) become popular within the enterprise gradually. Enterprise application must interact with disparate data sources, including databases, file systems, Web pages, and other applications. Heterogeneity and interoperability become one of the key issues in enterprise information extraction and integration. A solution to this heterogeneous information integration problem is that providing a uniform access to data obtainable from different sources in EB environment.

Extensible Markup Language (XML) and XQuery have become the standard data exchange format and query language respectively. Therefore, most researches have adopted them as standard input and output to integrate heterogeneous data sources. Using XML can resolve the problem that different data sources store their data in different structures. But XML shows some limitations on the semantic heterogeneity resolution. Because XML tags are human-readable, not machine-readable, the meaning of the information interchanged cannot be understood across different systems.

Ontology provides much richer modeling means with classes and properties organized into is-a hierarchy and enriched with axioms and relations processable with inference. Using ontology for the explication of implicit and hidden knowledge is a possible approach to overcome the problem of semantic heterogeneity. However, in method, there is still no adequately systematic benchmark method for “quantitative” performance measurement and evaluation on heterogeneous information integration.

1.2 Research Problem

Information integration issue has arisen in 1980s. Today, with the popularity of Internet/Intranet, it becomes a hot information technology (IT) topic in EB field. There have been research projects applying XML and ontology as mediated techniques to consolidate heterogeneous information. In order to measure and evaluate the mechanism for heterogeneous information integration, a benchmark approach to these new techniques is needed.

There are separate benchmarks on XML, relational database, object-oriented database, and Web server. But they are independent and use a predetermined set of test database and test query. When the domain or application changes, they are not reproducible.

Domain dependency is a core issue in current benchmarks. Heterogeneous information integration needs to incorporate XML and ontology. However, there is still no adequate benchmark developed for such integration in EB environment. Information integration methods can be classified into global-as-view (GAV) defining the global schema as a view over the local schemas and local-as-view (LAV) defining the local sources as views over the global schema (Manolescu, Florescu, & Kossmann, 2001). GAV approach is chosen over LAV in this research, because query on the global schema transformed into queries on the local sources is better fit in EB environment.

1.3 Research Objective

Developing a benchmark requires the definition of a workload model. A workload is the core of a benchmark. In this research, we propose a workload model that incorporates XML and ontology in heterogeneous information integration under EB environment. It evaluates the XML processing performance of heterogeneous information integration systems. In order to capture the semantic aspect of them, the workload model is also designed to evaluate whether the ontology can represent the real meaning of heterogeneous information.

This workload model is designed to be domain independent and generic-construct-based. It is hard to apply a domain-specific benchmark to different application domains. The generic model describes the data structure and usage of the system that do not tie with a predetermined scenario. The workload model is developed with intent to meet the desirable characteristics of a good benchmark. First, the workload model can scale with the complexity of data and operation. Second, the workload model adopts open standards, such as generic constructs in relational model, object model, Web page, XML, and ontology. This makes the workload model to be portable. Third, the workload model is simple to understand and implement because of generation process is automated.

2 Literature Review

Benchmark is an important method to evaluate the performance of different computer systems. It is wildly used in the database system performance measurement and evaluation for a long time. While XML emerging as a leading data format, several benchmarks used to evaluate the XML management system have been proposed. However, there is still no guideline for evaluation of ontology.

To develop a benchmark, it is necessary to define a workload first. The workload model consists of three parts: the data model, the operation model, and the control model. In XML benchmark, the operation model is a comprehensive set of queries, and it is the

most important part in the benchmark. We introduce the XML query functionality first. It would help us to review the operation model of each existing XML benchmark. Consequently, three XML benchmark projects are introduced by their data model, operation model, and control model. Then we compare them with this research. Finally, the ontology related benchmark works are reviewed and discussed.

2.1 XML Query Capability

Benchmarking the XML data management systems should consider many factors. Designing a set of comprehensive queries to test the XML databases' performance is an important point. XML query languages should capture the whole characteristics of a XML document, and the functionalities they provide would influence the query performance. The W3C XML Query Language working group (Chamberlin, Fankhauser, Marchiori, & Robie, 2003) list 20 XML query language "must have" functionalities. Some of the expected functionalities may affect the efficiency of the system significantly.

XQuery has met all of the requirements except F12 and F16, and it becomes a standard query language to test the performance of XML data management systems. Generally speaking, queries to benchmark XML databases would fall into several categories: Match, Join, Navigation, Casting, Reconstruction, and Update. Queries for Match are mainly used to test the database ability to handle simple string lookups with a fully specified path.

Join queries can be divided into two parts: Join on References, and Join on Values. References are an important part of XML, because they allow richer relationships than just hierarchical structure. Queries Join on References would test if query optimizer can take advantage of references to be joined. Queries Join on Values, on the other hand, would test the database's ability to handle large intermediate results. Differing from the former, their joins are on the basis of values. Navigation Queries investigate how well the query processor can optimize path expressions, and avoid traversing irrelevant parts of the tree. Strings are the basic data type in XML documents. Casting strings to another data type that carries more semantics is necessary. Queries for Casting challenge the ability of the database to cast different data types. Reconstruction Queries attempt to reconstruct the original document from its fragmentations stored in the databases. Update Queries try to add, delete, and modify elements in the XML document. These queries test the databases' ability to manage XML document. Furthermore, other XML query functionalities such as sort, ordered access, text search, and aggregation also should be captured in the benchmark query set.

2.2 XML Benchmarks

As XML becomes a dominant technology on the Web, it is beginning to be

extensively used in various application domains. In order to manage large amounts of XML documents, XML storage and management systems are being offered by most data management vendors. A benchmark to identify the important performance parameters for these various systems under varying levels of load and differing environments has thus become a necessity. XMark, XMach-1 and XOO7 are three benchmarks available today that can be used to evaluate certain aspects of XML database systems. We will first briefly describe these benchmarks and their queries before comparing them with this research.

2.2.1 XMark

XMark (Schmidt, Waas, Kersten, Carey, Manolescu, & Busse, 2002) is a single-user benchmark.

- Data Model

The data model of XMark is an Internet auction site. Therefore, its database contains one big XML document with text and non-text data. XMark enriches the references in the data, like the item IDREF in an auction element and the item's ID in an item element. The text data used are the 17000 most frequently occurring words of Shakespeare's plays. The standard data size is 100MB with a scaling factor 1.0 and users can change the data size by 10 times from the standard data (the initial data) each time. However, it has no support for XML Schema.

- Operation Model

In operation model of XMark, 20 XQuery challenges are designed to cover the essentials of XML query processing. No update operations are specified in XMark. We find that XMark includes almost complete query functionality. Each query only tests a single aspect of XML. This would help people to explain the performance result easily. But some queries are functionally similar in testing certain features of the query optimizer. Another notable feature is that XMark does not specify any update operation.

- Control Model

In XMark, the control model contains a repetition factor, which indicates how often a query was executed in the same environment.

2.2.2 XMach-1

XMach-1 (Böhme & Rahm, 2001) is a scalable multi-user benchmark. The main objective of the benchmark is to stress-test XML systems under a multi-user workload.

- Data Model

The data model of XMach-1 is designed for B2B applications and considers text documents and catalog data. It assumes that size of the data files exchanged will be small. It provides support for DTD only and does not consider XML Schema for optimization.

- Operation Model

The operation model of XMach-1 consists of eight queries and three update operations.

Queries specified in XMach-1 cover typical database functionality (join, aggregation, sort) as well as information retrieval and XML-specific features (document assembly, navigation, element access). Update operations cover inserting and deleting of documents as well as changing attribute values. We find that some queries contain several query functionalities. For example, Q8 needs count, sort, join and existential operations and accesses metadata. It is hard to analyze the experiment result and ascertain which feature leads to the given performance result. Specially, XMach-1 has defined three update operations that are unique across other XML benchmarks.

- Control Model

To achieve a true multi-user environment with a realistic number of concurrent clients, XMach-1 requires that each browser and loader runs at most one operation at a time. Furthermore, after completing an operation there is a think time between 1 and 10 seconds before the next operation is started.

2.2.3 XOO7

XOO7 (Li, Bressan, Dobbie, Lacroix, Lee, Nambiar, & Wadhwa, 2001) is an XML version of the OO7 benchmark, which was designed to test the efficiency of object-oriented DBMS. XOO7 is a single-user based benchmark for XMLMS that focuses on the query processing aspect of XML.

- Data Model

The data model of XOO7 comes from the OO7 benchmark by mapping the OO7 schema and data set to XML. No specific application domain is modeled by the data of XOO7. It is based on a generic description of complex objects using component-of relationships. XOO7 also proposes three different databases of varying size: small, medium, and large. It supports DTD only.

- Operation Model

In operation model, XOO7 provides relational, document and navigational queries that are specific and critical for XML database applications. These queries test the primitive features and each query covers only a few features.

XOO7 contains large amount of queries, each query covers only a few features. Comparing to the other two benchmarks, XOO7 has certainly the highest ratio which stresses its data-centric focus. However, we can find that some queries are focus on the same functionality. Similar to XMark, no update operation is specified in XOO7.

- Control Model

The control model in XOO7 is the number of repetitions.

2.3 XML Benchmarks Comparison

The key features include application focus, evaluation scope, database and workload characteristics, etc. Compared to other XML benchmarks, XMark provides a concise and comprehensive set of queries. However, it does not provide update operations to manipulate XML documents. XMach-1 only defines a small number of XML queries that cover multiple functions and update operations for which system performance is determined. XOO7 maps the original queries of OO7 into XML, and adds some XML specific queries. In general, XMach-1, XMark and XOO7 cover only a subset of the XML query requirements. In this research, we attempt to propose a generic workload model. In order to cover the whole functionalities of XML query processing, we combine queries of these three XML benchmarks and integrate them into 10 types of queries. In particular, the information integration system is generally used for query data, not provide data manipulation functions. Therefore, the query model in this research does not support update operations.

2.4 Ontology

Currently the information integration issue attracts researchers from all around the world. Numerous information integration systems are already available and the number is growing fast. Ontologies play an important role for integration as a way of formally defined terms for communication. They aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused, shared, and operationalized across applications and groups.

A good ontology should represent the domain specific knowledge explicitly. The question is how do we know an ontology is good? The answer is the ontology benchmark. There are plenty of benchmark studies in other fields like database or compilers. However, there are no specific benchmarks studies or tools for evaluating ontology-based applications. In fact, there is still no guideline to evaluate ontologies and related technologies.

In this section, we introduce the role of ontologies in information integration first. And then we discuss a major inference task, which is the main operation of an ontology benchmark. Finally, the ontology related benchmark works are reviewed and discussed.

2.4.1 Ontology and Information Integration

Traditional integration approaches use inexpressive models of database schemas or XML trees to integrate heterogeneous data sources. This would cause many semantic heterogeneity problems. Ontologies provide much richer modeling means with classes and properties organized into is-a hierarchy and enriched with axioms and relations processable with inference. Main benefits for an ontology-based approach are illustrated

as follows (Maier, Aguado, Bernaras, Laresgoiti, Pedinaci, Pena, & Smithers, 2003):

- The ability to picture all occurring data structures, for ontologies can be seen as nowadays most advanced knowledge representation model.
- The combination of deduction and relational database systems, which extends the mapping and business logic capabilities.
- A higher degree of abstraction, as the model is separated from the data storage.
- Its extendibility and reusability.

Almost all ontology-based integration approaches ontologies are used for the explicit description of the information source semantics. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts. Some approaches use ontologies not only for content explication, but also either as a global query model or for the verification of the (user-defined or system-generated) integration description (Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann, & Hübner, 2001). Ontologies are usually expressed in a logic-based language, so that fine, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations. Therefore, ontologies not only have the expressiveness needed in order to model the data in the sources, but their reasoning ability can help in the selection of the sources that are relevant for a query of interest, as well as to specify the extraction process. Ontologies let domain experts, system developers, and applications perform reasoning about information content in an application domain.

2.4.2 Ontology and Reasoning

Ontologies intend to provide a machine-understanding syntax for information integration. Understanding is closely related to reasoning. Reasoning is important to ensure the quality of an ontology. During ontology design, it can be used to test whether concepts are non-contradictory and to derive implied relations. It may also be used when the ontology is deployed, one can determine the consistency of facts stated in the annotation with the ontology or infer instance relationships (Baader, Horrocks, & Sattler, 2003). Therefore, reasoning is the major operation in the ontology-based application. The workload model of the ontology benchmark should identify key reasoning tasks in the operation model.

Tempich and Volz (2003) mention that a reasoner supporting ontology languages usually offers several different query services with respect to an ontology. These query services primarily target queries about classes. They fall into four categories, class-instance membership queries, class subsumption queries, class hierarchy queries, and class satisfiability queries. There are similar queries about properties, i.e. property-instance membership, property subsumption, property hierarchy, and property

satisfiability, and also the possibility to check the consistency of the whole ontology.

Simov and Jordanov (2002) cite that ontologies within their ontology-based project have two types of reasoning tasks, terminological reasoning and instance reasoning. Terminological reasoning checks the classes are defined and the relations between them are explicitly represented. Instance reasoning involves first an already developed ontology (after some terminological reasoning) and next large amounts of instances.
We find that terminological reasoning is similar to class subsumption queries, class hierarchy queries, and class satisfiability queries. Instance reasoning is similar to class-instance membership queries. This would provide this research with the basis of major reasoning tasks in the operation model of the ontology benchmark workload model.

2.4.3 Ontology and Benchmark

To the best of our knowledge, the benchmark presented here is the first one for ontology-based information integration. The ontology benchmark model in this research differs from database benchmarks, such as Wisconsin benchmark, OO7 benchmark, and BUCKY benchmark. They are all DBMS-oriented and storage benchmarks, and there is no inference ability included. In this research, the ontology workload model is applied to an information integration system, and we focus on the inference ability of the ontology.

Ontology and XML are often found together and are often confused. XML is a standard for marking up - adding additional information, called metadata - to documents. The purpose of XML is to tag textual information with additional structure that enables it to be “understood” and exchanged by programs. However, XML tags still require humans to interpret their meanings. Therefore, XML benchmarks only focus on structural and syntactic evaluation of systems, and they have no semantics. On the other hand, ontology benchmark is devoted to capture the semantic expressions in the system. Thus, ontology and XML are complementary technologies: ontology provides the meaning for XML standards; XML provides a valuable medium for information exchange between programs that share the same ontology (Andersen, 2001).

As mentioned above, there is still no guideline for evaluation of ontology-based application. Horrocks and Patel-Schneider (1998) benchmark description logic systems, or so-called knowledge bases. Description logics (DLs) are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way. Description logic systems provide their users with various inference capabilities that deduce implicit knowledge from the explicitly represented knowledge. Horrocks and Patel-Schneider try to evaluate the reasoning algorithms in description logics.

The knowledge base is composed of a Tbox and an Abox. Terminological part (Tbox)

is a set of axioms describing the structure of domain. Assertional part (Abox) is a set of axioms describing concrete situation (Horrocks, 2002). They are related to this research. In an information integration system, the ontology can be viewed as the Tbox, and the heterogeneous data can be viewed as Abox. However, the logic described is only a subset of the ontology languages, such as DAML+OIL and OWL. DAML+OIL and OWL can be seen to be equivalent to a very expressive description logic. They provide more constructors and allow more axioms than description logic. Therefore, the inference services of ontology are more complex than traditional description logic systems.

3 Research Method

We describe the research method and research model of this work. After related literature review in the previous section, we begin to develop the research model. The research structure is described in the next section.

3.1 Research Structure

In this research, the benchmark workload model is used to evaluate the performance of the heterogeneous information integration systems. The benchmark is run on several different systems, and the performance and price of each system is measured and recorded. The literatures would help us to identify the important performance factors for XML and ontology processing. We analyze the XML-specific and ontology-specific requirements in more details to justify the design of the benchmark.

The benchmark study consists of two benchmark workload models, the XML benchmark workload model and the ontology benchmark workload model. Both of them consist of the data model and query model according to the generic constructs and constraints requirements. Next, the control model is created before the generic workload model to be generated and executed so as to measure and evaluate the systems.

3.2 Research Model

In this research, we focus on heterogeneous information sources integrated in XML and ontology. The benchmark model we propose would capture most features of the released XML-based and ontology-based specifications. Designed as a generic benchmark model, it is easy to implement the benchmark on many different systems and architectures. To support scalability, this benchmark also provides different workloads.

Developing a benchmark requires the definition of the test workload model first. In this research, we provide a benchmark workload model that combines XML and ontology in heterogeneous information integration. In XML workload model, the data model describes a generic XML data model and the operation model defines a comprehensive

set of test queries that covers the major aspects of XML query processing. In ontology workload model, the data model describes the major ontology component, and the operation model defines some important criteria to query the ontology. The control model defines the variables that used to set up the benchmark environment. The data model, operation model, and control model define the experimental factors of the benchmark. In addition, we should define the performance metrics to measure the benchmark results.

3.3 XML Data Model

XML is a hierarchical data format for information exchange on the Web. An XML document consists of nested elements that contain data or other elements. The boundaries of these elements are either delimited by start-tags and end-tags, or, for empty elements, by empty-element tags. The text between start-tags and end-tags is the content of the element. Each element has a type, identified by name, sometimes called its "generic identifier" (GI), and may have a set of attribute specifications. Each attribute specification has a name and a value (Bray, Paoli, Sperberg-McQueen, & Maler, 2000). XML documents may comply with a Document Type Definition (DTD) or a XML Schema. DTD has traditionally been the most common method for describing the structure of XML document. But DTD lacks enough expressive power to properly describe highly structured data. XML Schemas are an XML language for describing and constraining the content of XML documents. It provides a richer and more powerful means for defining the data. Therefore, XML schema becomes the most common method for defining and validating highly structured XML documents rapidly.

We employ the XQuery 1.0 and XPath 2.0 Data Model published by the W3C to represent XML documents (Fernández, Malhotra, Marsh, Nagy, & Walsh, 2003). In the XQuery 1.0 and XPath 2.0 Data Model, XML documents are modeled as an ordered tree. The tree contains seven distinct kinds of nodes: document, element, attribute, text, namespace, processing instruction, and comment. In this research, for simplicity, we only consider document, element, attribute, and text nodes. The data model is a node-labeled, directed graph, in which each node has a unique identity. Document order is defined for all the nodes in the document and corresponds to the order in which the first character of each node occurs in the XML document. We briefly introduce the four nodes as follows:

- **Document nodes:** The document node is a virtual node pointing to the root element of an XML document. The document element in a XML document is a child of the document node.
- **Element nodes:** Every element in the document is an element node. Element nodes have zero or more children that can be element nodes or text nodes.
- **Attribute nodes:** Each element node has an associated set of attribute nodes. Note that the element node that owns this attribute is called its "parent" even though an

attribute node is not a "child" of its parent element. An attribute node has an attribute name and an attribute value. Attribute nodes have no child nodes. If more than one attribute of an element node exists, the document order among the attributes is not distinguished. This is because there is no order among XML attributes.

- **Text nodes:** A text node must have only one parent and have no child nodes. A text node cannot contain an empty string as its content.

Document order in this representation can be found by following the traditional in-order, left-to-right, depth-first traversal. The value D1 represents a document node; the values E1, E2, etc. represent element nodes; the values A1, A2, etc. represent attribute nodes; the values T1, T2, etc. represent text nodes. The IDREF attribute nodes are used for intra-document references (Fernández et al., 2003; YoshiKawa & Amagasa, 2001; Jiang, Lu, Wang, & Yu, 2002).

3.4 XML Query Model

In this research, we attempt to propose a generic XML query model applicable to any scenario. We do not describe the queries based on a specific application domain such as an auction site or a library. The queries are specified in generic terms. It is easy for user to apply them in different scenarios. Besides, we further identify key factors that influence the complexity of each query. This would help users to evaluate performance of the system with increasing complex queries.

After analysis previous three XML benchmarks, we identify a comprehensive set of queries. The query model we defined can be classified into 10 categories, including 14 different queries. Each of them challenges different aspects of XML processing. Besides, users can specify queries according to their requirements, called "user-driven query". The following will describe each category briefly, and express each query in generic terms. In each query, the generic term is written in italics. Then we illustrate them in XQuery. We use E1, E2 etc. to denote a certain element, and A1, A2 etc. to denote a certain attribute. The number of them does not indicate their order in a XML document, just for representing convenience. Finally, the complexity factors will be discussed.

3.4.1 Exact Match

This type of queries specifies a full path expression. One main concept of XQuery is the use of path expressions for selecting nodes. The length of the path expression depends on the levels of predicates being queried in XML documents. This is the simplest query type. We can use this type of queries to establish a simple "metric" comparing performance of the following queries. It tests the database ability to handle simple string lookups with a fully specified path.

Generic terms:

Given a full path expression, find elements E1 that have an attribute A1 in a value X.

XQuery expression:

```
FOR     $a IN input()/SUBPATH/E1[@A1 = "X"]
RETURN  $a
```

The complexity of the query is influenced by the length of the path expression. Queries with different level of predicate would have different performance.

3.4.2 Joins

References are an integral part of XML identifying the relationship between related data. With using of reference, richer relationships can be represented than just hierarchical element structures. The system must be able to combine separate information together using joins. Horizontal traversals are defined in this type of queries. Joins can be on the basis of references and values. References are specified in the DTD and may be optimized with logical OIDs for example. The system should make use of the cardinalities of the sets to be joined. Joins based on values test the database's ability to handle large (intermediate) results.

- **Join on Reference**

Generic terms:

Find element E1 by the reference attribute A1 of E2. The reference attribute A1 of E2 refer to E1.

XQuery expression:

```
FOR     $a IN input()//E1
        $b IN input()//E2
WHERE   $a/@A2 = $b/@A1
RETURN  $a
```

- **Join on Value**

Generic terms:

This time reference is based on join of the data values. Find element E1 whose attribute A1 is equal to the attribute A2 of E2.

XQuery expression:

```
FOR     $a IN input()//E1
        $b IN input()//E2
WHERE   $a/@A1 = $b/@A2
RETURN  $a
```

The queries specified above are 2-way join. It is the simplest form. 3-way join, 4-way join, and N-way join would be generated with increasing complexity. In addition, the result size would affect the query efficiency too.

3.4.3 Regular Path Expressions

Regular path expressions are a basic building block of almost every XML language including XPath, XQuery, and XSLT. The system should be capable of optimizing path expressions and reducing traversals of irrelevant parts of the tree. We often use wildcards in regular path expressions and the system should realize that it is not necessary to traverse the complete document tree to execute such expressions. This type of queries tries to quantify the costs of long path traversals that do not include wildcards, and the costs of path traversals that include wildcards.

- **Full Sub-path**

Generic terms:

Find element E1 with a long path expression.

XQuery expression:

```
FOR     $a IN input()/SUBPATH/E1
RETURN  $a
```

- **Unknown Sub-path**

Generic terms:

Find element E1 with a regular path expression include wildcards.

XQuery expression:

```
FOR     $a IN input()//E1
RETURN  $a
```

The length of path expression would influence the complexity. In a path expression, each step can apply one or more predicates to eliminate nodes that fail to satisfy a given condition. Therefore, numbers of element unknown in the sub-path would also affect the query complication.

3.4.4 Document Construction

Structure is very important to XML documents. But XML documents storing in relational DBMSs often need to be broken down. Reconstructing the original document is a big challenge to systems. We might retrieve fragments of original documents with original structures. But sometimes we may want to construct document fragments with

new structures. These queries tests for the ability of the system to reconstruct portions of the original XML document.

- **Structure Preserving**

Generic terms:

Return a XML document constructed by element E1 and its sub-element E2. Retrieve E2 of E1 that has an attribute A1 equal to a certain value X.

XQuery expression:

```
FOR     $a IN input()//E1[@A1 = X]
RETURN  <$a> $a/E2 </$a>
```

- **Structure Transforming**

Generic terms:

Construct a new XML document. Find element E1 with an attribute A1 equal to a certain value X, and select several sub-element of E1 to construct a new XML document.

XQuery expression:

```
FOR     $a IN input()//E1[@A1 = X]
RETURN  <output>
            {$a/E2/E3}
            {$a/E2/E4}
            {$a/E2/E3/E5}
            {$a/E6}
</output>
```

The complication of the XML document structure would increase the difficulty of reconstruction. On the other hand, the structure of output document would also influence the query complexity.

3.4.5 Ordered Access

Order of elements is important in XML documents. Because documents will sometimes be fragmented when they are stored on disk, it is important that the order of these fragments in the original document is preserved. The system should be able to preserve these intrinsic orders. This type of queries attempts to test how efficient the system handle queries with order constraints.

Generic terms:

Find element E1 with attribute A1 in certain value X, and return the first sub-element E2 of E1.

XQuery expression:

```
FOR      $a IN input()//E1[@A1 = X]
RETURN   $a/E2[1]
```

The complexity depends on order constraints specified in the query. If there is an index build on the attribute, the query can take advantage of set-valued aggregates on the index attribute to accelerate the execution.

3.4.6 Sorting

The order by clause is the only facility provided by XQuery for specifying an order other than document order. In XML documents, the generic data type of element content is string, but users may cast the string type to other types. Therefore, the system should be able to sort values both in string and in non-string data types. This type of queries tests whether the system can do sorting efficiently.

- **By String**

Generic terms:

List sub-element E3 of element E1 sorted by sub-element E2.

XQuery expression:

```
FOR        $a IN input()//E1
ORDER BY   $a//E2
RETURN     $a/E3
```

- **By Non-string**

Generic terms:

List sub-element E3 of element E1 sorted by sub-element E2. This time E2 is a non-string value.

XQuery expression:

```
FOR        $a IN input()//E1
ORDER BY   $a//E2
RETURN     $a/E3
```

The number of tuples that are generated by the FOR and LET-clauses and satisfies the condition in the WHERE-clause would influence the complexity of the query. Also, if the ORDER BY-clause uses several options, the complexity would increase.

3.4.7 Missing Elements

In XML, schemas are more flexible and may have a number of irregularities. Queries in this type are to test how well the system knows to deal with the semi-structured aspect of XML data, especially elements that are declared optional in the schemas.

Generic terms:

Find element E1 whose sub-element E2 has NULL value.

XQuery expression:

```
FOR     $a IN input()//E1
WHERE   EMPTY($a/E2/text())
RETURN  $a
```

The complexity depends on the FOR and LET-clauses that generate the test tuples.

3.4.8 Text Search

Text search plays a very important part in XML document systems. This type of queries conducts a full-text search in the form of keyword search. They will challenge the textual nature of XML documents.

Generic terms:

Find element E1 whose sub-element E2 contains a specific text Y.

XQuery expression:

```
FOR     $a IN input()//E1
WHERE CONTAINS ($a/E2, "Y")
RETURN  $a
```

This query has to scan large part of the document. Therefore, the number of tuples that are generated by the FOR and LET-clauses would influence the query complexity. Also, if the query contains multiple texts, the difficulty would increase.

3.4.9 Data-type Cast

Strings are the generic data type in XML documents. But we often need to cast strings to another data type that carries more semantics. These queries challenge the system's ability to transform between data types.

Generic terms:

Find element E1 with a constraint that contain operations need to transform data value

of sub-element E2 to other data-type. Retrieve element E1 whose sub-element E2 is bigger than a certain number X.

XQuery expression:

```
FOR     $a IN input()//E1
WHERE   $a/E2 > X
RETURN  $a
```

The number of tuples needs to be transformed affects the execution efficiency. If there are several casting conditions in a query, the complexity would increase.

3.4.10 Function Application

The following query challenges the system with aggregate functions such as count, avg, max, min and sum.

Generic terms:

Group element E1 by sub-element E2, and calculate the total number of elements for each group.

XQuery expression:

```
FOR     $a IN DISTINT-VALUES (input()//E1/E2)
LET     $b := input()//E1[E2 = $a]
RETURN  count($b)
```

The complexity of this query is influenced by the number of tuples that are generated by the FOR and LET-clauses.

3.5 Ontology Data Model

The term Ontology has been used in several disciplines. Recently, ontology becomes even common in computer science area. It can be used for many purposes, including enterprise integration, database design, information retrieval, and information interchange on the World Wide Web to overcome many traditional problems.

Gruber (1993) defines an ontology as "a formal, explicit specification of a shared conceptualization". An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and

constrain the possible interpretations of terms (Uschold, King, Moralee, & Zorgios, 1998).

Generally speaking, an ontology consists of the following main constructs (Stevens, Goble, & Bechhofer, 2000; Weißenberg & Gartmann, 2003).

- **Facts** represent explicit knowledge, consisting of:

1. **Classes or concepts** are generalizations of instances. Concepts are the focus of most ontologies. A concept is a representation for a conceptual grouping of similar terms. A concept can have subconcepts that represent concepts that are more specific than the superconcept. Concepts fall into two kinds:

(a) Primitive concepts are those which only have necessary conditions (in terms of their properties) for membership of the class.

(b) Defined concepts are those whose description is both necessary and sufficient for a thing to be a member of the class.

2. **Properties** can be subdivided into scalar attributes and non-scalar relations. The property can be defined to be a specialization (subproperty) of an existing property. An attribute is a property of a concept that refers to a datatype (integer, string, float, boolean etc.). An example of an attribute is "has-name" related to a string. A relation is a property of a concept that refers to another concept. Specialization / Generalization are one of the standard relations. For instance, "is a kind of" defines a relation that may be applied to the concepts "Enzyme" and "Protein".

3. **Instances** represent individual entities and are connected by type-of relation to at least one class; some authors only consider facts about instances as real facts. Strictly speaking, an ontology should not contain any instances, because it is supposed to be a conceptualization of the domain. The combination of an ontology with associated instances is what is known as a knowledge base. However, deciding whether something is a concept of an instance is difficult, and often depends on the application.

- **Axioms** are rules used to add semantics and to infer knowledge from facts. In contrast to facts, they represent implicit knowledge about concepts and relations, e.g., whether a relation is transitive or symmetric.

3.6 Ontology Query Model

Initially, ontologies are introduced as an "explicit specification of a conceptualization". In an information integration system, ontologies can be used to establish common vocabularies and semantic interpretations of terms from information sources. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts. People can share and exchange information in a semantically consistent way.

Using ontology basic components described in the previous section, user can define their

own ontology in any application domain. Then we conducted a series of tests to see how the system handles such ontologies. The operation model in the ontology benchmark workload model is a set of queries, and the answers are generated by inferring from the ontology. The queries we present here are representative for different application domains. We conclude the reasoning tasks and construct six basic reasoning queries for the ontology benchmark. We introduce these queries briefly and described them in generic terms as follows.

1. **Concept Subsumption Queries:** checks if one concept is a subconcept of another.
Generic terms:
Given concepts C and D, determine if C is a subcconcept of D with respect to ontology O.

2. **Concept Hierarchy Queries:** determines the concepts that immediate subsume or are subsumed by a given concept.
Generic terms:
Given a concept C return all/most-specific superconcepts of C and/or all/most-general subconcepts of C.

3. **Concept Consistency Queries:** checks for (in)consistency of concept definitions.
Generic terms:
Given a concept C, determine if the definition of C is generally satisfiable (consistent).

4. **Instance Checking Queries:** given a partial description of an individual (instance) and a concept description, finds whether the concept describes the instance.
Generic terms:
Given a concept C, determine whether a given individual A is an instance of C.

5. **Instance Retrieval Queries:** finds all instances that are described by a given concept.
Generic terms:
Given a concept C, determine all the individuals in ontology O that are instances of C.

6. **Instance Realization Queries:** given a partial description of an instance, finds the most specific concepts that describe it.

Generic terms:
Given an individual A, determine all the concepts in ontology O that A is an instance of.
The queries mentioned above are simple and basic. When querying the heterogeneous

information integration system, the reasoning service may not be so straightforward. We need to evaluate the ontology with increasing complexity. When formulating the complex benchmark queries, several factors should be taken into account (Guo, Heflin, & Pan, 2003):

- **Input size:** This is measured as the proportion of the class instances involved in the query to the total class instances in the benchmark data.
- **Selectivity:** This is measured as the estimated proportion of the class instances involved in the query that satisfy the query criteria.
- **Complexity:** We use the number of classes and properties that are involved in the query as an indication of complexity.
- **Hierarchy information assumed:** This considers whether information of class hierarchy or property hierarchy is required to achieve the complete answer. Besides, the depth and width of class hierarchies should also been considered.

More complex queries may be formulated according to these factors mentioned above. It lets the system be evaluated under different level of workloads.

3.7 Test Database Generation

In order to evaluate the performance of the heterogeneous information integration system, we must define the workload. The workload consists of a test operation and a test database. The test database identifies what data must be loaded into the data sources, as well as the volume of the test data. Information integration system data sources are disparate and heterogeneous. Information comes from various sources (including structured, semi-structured and unstructured sources) and formats (such as database tables, XML files, PDF files, streaming media, internal documents, and Web pages). For this research, the data sources can be divided into three kinds: relational databases, object-oriented databases, and Web pages. For each data source, we must analyze the actual data and extract statistical data. Data analysis characterizes data in terms of the size of the database, the number of records, the length of records, the types of fields, and the value distributions.

- **Determine data values**

A number of data types are supported in this research, including long integer number, double precision floating point number, decimal number, money, datetime, fixed-length and variable-length character strings. We must conduct extensive studies to characterize each data source with several distribution parameters. Frequency distributions are computed and standard probability distributions are fit to the data in order to generate the value of test data. Several standard benchmarks including the TPC-C, TPC-D, TPC-E, and AS^3AP all support uniform and non-uniform data distributions (e.g. exponential,

normal, discrete, rotating, zipfian[2] or constant). Data values are created with these common data distributions.

- **Determine scaling factors**

After determining the value of the test data, we must define how much data should be generated, i.e. defining the database scaling factor. Generally speaking, the logical size for the test database used for the benchmark is at least equal to the logical size of physical memory on the host(s). For this research, we refer to the AS^{3}AP benchmark standard. In the AS^{3}AP benchmark, the test database consists of four generic relations. Each has the same number of fields and the same number of records. The database scales up by increasing tenfold number of records for each relation. The tuple length is 100 bytes on the average. The logical size of the AS^{3}AP database is defined as follows:
Logical size = (# tuples per relation) * (100 bytes/tuple) * (4 relations in database)
For this research, the tuple length is fixed at 100 bytes on the average as well. The size of the logical database can be scaled from 1 megabyte to 100 gigabytes by varying the number of tuples from 10,000 to one billion.

- **Open data source**

We must determine the test data of the open data source, i.e. World Wild Web, but this is problematic. There is in excess of 9 billion pages on the Web, which include HTML files, text documents, PDF files, Microsoft Office documents and other similar data files. We cannot possibly download every page from the Web much less adequate sample size. Even the most comprehensive search engine currently indexes just a small fraction of the entire Web.

As such, it is important to carefully select the so-called "important" pages, so that the fraction of the Web that is visited becomes more meaningful. In order to select these important pages, we can use several metrics for prioritizing them. For any given Web page, we must define its importance using the following methods (Arasu, Cho, Garcia-Molina, Paepcke, & Raghavan, 2001):

1. **Interest-driven.** The goal is to obtain pages of interest to a particular user or set of users. Important pages are those that match user interest. One particular way to define this notion is through what we call a *driving* query. For any given query, the importance of a page is defined by the "textual similarity" between the page and the driving query. Assuming that query represents the user's interest, this metric shows how relevant the page is. Another interest-driven approach is based on a hierarchy of topics. Interest is defined by a topic, and we attempt to guess the page topics that will be visited by analyzing the link structure that leads to the candidate pages.

2. **Popularity-driven.** Page importance depends on how popular a page is. For instance, one way to define popularity is to use a page's backlink count. (We use the term backlink for links that point to a given page.) Intuitively, a page that is linked to by many pages is more important than one that is seldom referenced.

3. **Location-driven.** The importance of a page is a function of its location, not its contents. For example, URLs ending with ".com" may be deemed more useful than URLs with other endings, or URLs containing the string "home" may be of more interest than other URLs. Another location metric that is sometimes used considers URLs with fewer slashes more useful than those with more slashes.

For this research, the data unit for the Web is a page. The size of a Web page is 1 kilobyte on average. We can use rules mentioned above to prioritize Web pages, and select the important pages to load into the test database.

3.8 Control Model

The control model defines environment variables to execute the benchmark. These variables are used to set up the execution environment.

- **Steady State**

The benchmark test must be executed in a steady state, in order to return true performance of the system.

- **Test Mode**

There are three kinds of test mode, cold mode, warm mode, and hot mode. In cold mode, there is no data in the cache. The system cannot retrieve data from the cache directly. Therefore, the performance in cold mode is usually slower than other two modes. In warm mode, the data is left in the cache from prior query. Because of that, the test response time decreases. In hot mode, a query is executed in cold mode first, and then be executed with cache data for several times. The average response time is computed.

- **Test Duration**

Test duration means time intervals of the benchmark. Each interval must begin after the system has reached steady state and be long enough to generate reproducible throughput results. Each interval must extend uninterrupted for a period of time.

- **Test Sequence**

Test sequence indicates the order of the queries execute.

- **Number of Repetitions**

Number of repetitions means execution repeated times of an operation in a test.

3.9 Performance Metrics

Performance metrics are used to measure the execution result. Response time and throughput are two performance metrics often used in evaluation of computer systems.

- **Response time** means time interval between when a request is made and when the response is received by the requester.
- **Throughput** means the number of operations completed by the system per unit time.

In the ontology benchmark, system would generate answers that are entailed by the ontology. Notably it is not sufficient to consider response time and throughput as metrics. Two fundamental measures of the quality of information retrieval, recall and precision, can be used to evaluate the performance. Besides, error probability should be taken into account together.

- **Recall** is the percentage of relevant data which has been retrieved. In this research, it can be used to measure whether the system can generate all answers that are entailed by the ontology. Therefore, it also can be called **completeness**.

Recall = A / (A + B)

- **Precision** is the percentage of retrieved data which is relevant. In this research, it defines the level of "noise" in the information presented to the user.

Precision = A / (A + C)

- **Error probability.** If the answer is irrelevant, or the relevant answer is not retrieved, there is an error occur. The error probability should be calculated.

Error Probability = (B + C) / (A + B + C + D)

Relevance is an abstract measure of how well the data satisfies the user's information need, i.e. what the user really wants to know. Ideally, your system should retrieve all of the relevant documents for you. Unfortunately, this is a subjective notion and difficult to quantify (Weiss, 1997). In this research, the ontology benchmark workload model is implemented on a small, human observable ontology. It is easy for users to identify the relevant information of each query.

4 Conclusions and Future Research Directions

4.1 Summary

Heterogeneous information integration on the Web involves a number of new techniques. This includes mechanisms for information encoding and manipulation (e.g. XML, RDF, XSLT), and ontology construction and reasoning (e.g. RDFS, DAML+OIL, OWL). In order to manage and use information more effectively within the enterprise, a benchmark used to evaluate the mechanism of heterogeneous information integration is needed. In this research, we have developed the XML and ontology benchmark workload model in heterogeneous information integration, and built a workload generation prototype. We have reviewed the XML and ontology related literature to motivate the design of the workload model. The objective of this research is to develop a workload

model to test whether the heterogeneous information integration system under EB environment can overcome the diverse formats of content and derive meaning from this content. In order to apply the workload model to different scenarios easier, it is designed in generic constructs. Finally, we validate the research model through the prototype implementation. The results in this research include:

- Collecting and reviewing the literature on XML standardization, XML benchmarks, ontology standardization, and ontology related benchmarks. Identifying major requirements for a XML or ontology benchmark.
- Developing a generic XML and ontology benchmark workload model in heterogeneous information integration. The workload model consists of XML and ontology data model and query model according to the generic constructs and constrains requirements. Also, a control model is created to set up the benchmark environment.
- Implementing a workload generation prototype based on the workload model in this research. The prototype illustrates the feasibility and validity of the research model.

4.2 Future Research Directions

This research only built a simple prototype of the XML and ontology benchmark workload model of heterogeneous information integration. It still needs more effort to expand its capabilities. We expect this work to continue and evolve in the future. Future research directions include:

- Enhancing the ontology query model. The development of an ontological standard presents many opportunities and challenges. New reasoning tasks may arise in the future. Retrieval (instances of a concept) and realization (most specific class of instance) may not be sufficient. In order to make the ontology query model more comprehensive, further study to keep track of ontology progression is needed.
- Improving the complexity factors of the XML query model. The complexity factors we analyze in the XML query model are still too rough. Each query type can be analyzed more carefully to refine the query model.
- Implementing various data distributions. In this research, only uniform distribution is implemented. It cannot evaluate performance under different distributions. Implementation of diverse data distributions will become a user requirement.
- Applying the workload model to other applications. Ontology and XML are complementary technologies, and there are other applications that can apply. In this research, we assume the heterogeneous information integration system is used on Intranets, such as enterprise information integration (EII), electronic business (EB), and enterprise application integration (EAI). There are other applications between enterprises that may need to integrate heterogeneous information, such as business-to-business integration (B2Bi), collaborative commerce (C-Commerce), and electronic commerce

(EC). We can modify the workload model of this research to create other benchmarks that are based on XML and ontology with different characteristics.

References

1. Andersen, B. (2001). What is an ontology. Retrieved February 5, 2004, from http://www.ontologyworks.com/docs/what-is-ontology.pdf
2. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., & Raghavan, S. (2001). Searching the Web. ACM Transactions on Internet Technology, 1(1), 2–43.
3. Böhme, T., & Rahm, E. (2001). XMach-1: A Benchmark for XML Data Management. *Proceedings of German database conference BTW2001*, Oldenburg, Germany, 264-273.
4. Böhme, T., & Rahm, E. (2003). Multi-User Evaluation of XML Data Management Systems with XMach-1. *Lecture Notes in Computer Science (LNCS), 2590*, 148-159.
5. Bos, B. (1997). *The XML Datamodel*. Retrieved January 30, 2004, from http://www.w3.org/XML/Datamodel.html
6. Beech, D., Malhotra, A., & Rys, M. (1999). A Formal Data Model and Algebra for XML. W3C XML Query working group note.
7. Bray, T., Paoli, J., Sperberg-McQueen, C. M., & Maler, E. (2000). *Extensible Markup Language (XML) 1.0 (Second Edition)*. Retrieved January 30, 2004, from http://www.w3.org/TR/REC-xml
8. Baader, F., Horrocks, I., & Sattler, U. (2003). Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan (Ed.), *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. Springer.
9. Chamberlin, D., Fankhauser, P., Marchiori, M., & Robie, J. (2003). *XML Query Requirements.* Retrieved January 8, 2004, from http://www.w3.org/TR/xquery-requirements/
10. Cui, Z., Jones, D., & O'Brien, P. (2001). Issues in ontology-based information integration. *Proceedings of IJCAI-01 Workshop on E-Business & the Intelligent Web.*
11. Elhaik, Q., Rousset, M-C, & Ycart., B. (1998). Generating Random Benchmarks for Description Logics. *Proceedings of DL'98.*
12. Fernández, M., Malhotra, A., Marsh, J., Nagy, M., & Walsh, N. (2003). *XQuery 1.0 and XPath 2.0 Data Model.* Retrieved January 30, 2004, from http://www.w3.org/TR/xpath-datamodel/
13. Gray, J. (1993). *The Benchmark Handbook* (2nd ed.) . Morgan Kaufmann, San Mateo, CA, Retrieved January 8, 2004, from http://www.benchmarkresources.com/handbook/index.asp
14. Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Workshop on Formal Ontology,* Padova, Italy.

15. Guo, Y., Heflin, J., & Pan, Z. (2003). Benchmarking DAML+OIL Repositories. *Proceedings of the 2nd International Semantic Web Conference, LNCS, 2870*, 613-627.
16. Gómez-Pérez, A. (1994). Some Ideas and Examples to Evaluate Ontologies. *Technical Report KSL-94-65*, Knowedge Systems Laboratory, Stanford University.
17. Gruninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *Proceedings of IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing.*
18. Horrocks, I. (2002). DAML+OIL: A Reason-able Web Ontology Language, *Proceedings of the 8th International Conference on Extending Database Technology: Advances in Database Technology*, 2-13.
19. Horrocks, I. (2002). *DAML+OIL and Description Logic Reasoning.* Retrieved February 12, 2004, from http://www.dcs.shef.ac.uk/~angus/daml-oil-workshop/presentations/horrocks.pdf
20. Horrocks, I., & Patel-Schneider, P. (1998). DL systems comparison. *Proceedings of DL'98.*
21. Heflin, J. (2003). *OWL Web Ontology Language Use Cases and Requirements.* Retrieved February 5, 2004, from http://www.w3.org/TR/webont-req/
22. Jiang, H., Lu, H., Wang, W., & Yu, J. X. (2002). Path Materialization Revisited: An Efficient Storage Model for XML Data. *The 13th Australasian Database Conference (ADC 2002), Melbourne*, Australia, 85-94.
23. Li, Y. G., Bressan, S., Dobbie, G., Lacroix, Z., Lee, M. L., Nambiar, U., & Wadhwa, B. (2001). XOO7: applying OO7 benchmark to XML query processing tool. *Proceedings of the tenth international conference on Information and knowledge management (CIKM)*, Atlanta, Georgia, USA, 167-174.
24. Lehti, P. (2001). *Design and implementation of a data manipulation processor for an xml query processor.* Technical University of Darmstadt, Darmstadt, Germany, Diplomarbeit.
25. Maier, A., Aguado, J., Bernaras, A., Laresgoiti, I., Pedinaci, C., Pena, N., & Smithers, T. (2003). Integration with Ontologies. *Wissensmanagement 2003*, 21-24.
26. Manolescu, I., Florescu, D., & Kossmann, D. (2001). Answering XML Queries over Heterogeneous Data Sources. *Proceedings of the 27th VLDB Conference*, Roma, Italy.
27. Nambiar, U., Lacroix, Z., Bressan, S., Lee, M. L., & Li, Y. G. (2002). Efficient XML Data Management: An Analysis. *Proceedings of the 3rd International Conference on Electronic Commerce and Web Technologies (ECWeb)*, Aix en

Provence, France, 87-98.

28. Nambiar, U., Lacroix, Z., Bressan, S., Lee, M. L., & Li, Y. G. (2002). Current Approaches to XML Management. *IEEE Internet Computing Journal, 6(4),* 43-51.
29. 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.*
30. Noy, N. F., & Musen, M. A. (2002). Evaluating Ontology-Mapping Tools: Requirements and Experience. *Proceedings of the OntoWeb-SIG3 Workshop EON 2002 at EKAW 2002,* Siguenza, Spain, 1-14.
31. Omelayenko B. (2002). Ontology-Mediated Business Integration. *Proceedings of the 13-th EKAW 2002 Conference,* Siguenza, Spain, LNAI 2473, 264-269.
32. Rys, M. (2002). *Proposal for an xml data modification language.* Microsoft Corp., Redmond, WA, Proposal.
33. Schmidt, A., Waas, F., Manegold, S., & Kersten, M. (2003). A Look Back on the XML Benchmark Project. *Lecture Notes in Computer Science (LNCS), 2818,* 263-278.
34. Schmidt, A. R., Waas, F., Kersten, M. L., Florescu, D., Manolescu, I., Carey, M. J., & Busse, R. (2001). The XML Benchmark Project. *Technical Report INS-R0103,* CWI, Amsterdam, The Netherlands.
35. Schmidt, A., Waas, F., Kersten, M., Florescu, D., Carey, M. J., Manolescu, I., & Busse, R. (2001). Why and how to benchmark XML databases. *ACM SIGMOD Record, 30(3),* 27-32.
36. Schmidt, A. R., Waas, F., Kersten, M. L., Carey, M. J., Manolescu, I., & Busse, R. (2002). XMark: A Benchmark for XML Data Management. *Proceedings of the International Conference on Very Large Data Bases (VLDB),* Hong Kong, China, 974-985.
37. Sengupta, A., & Mohan, S. (2003). *Formal and conceptual models for XML structures - the past, present and future.* Retrieved January 30, 2004, from http://www.indiana.edu/~isdept/research/papers/tr137-1.pdf
38. Stevens, R., Goble, C.A., & Bechhofer, S. (2000). Ontology-based Knowledge Representation for Bioinformatics. *Briefings in Bioinformatics, 1(4),* 398-414.
39. Suarez-Figueroa, M. C., & Gomez-Perez, A. (2003). Results of Taxonomic Evaluation of RDF(S) and DAML+OIL ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms import services. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003),* Sanibel Island, Florida, USA.

40. Staab, S., Schnurr, H. P., Studer, R., & Sure, Y. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems, 16(1)*, 26-34.
41. Simov, K., & Jordanov, S. (2002). BOR: a pragmatic DAML+OIL reasoner. *On-To-Knowledge deliverable D-40*, OntoText Lab.
42. Sullivan, D. (2003). Search Engine Sizes. Retrieved May 6, 2004 from http://searchenginewatch.com/reports/article.php/2156481
43. Tempich, C., & Volz, R. (2003). Towards a benchmark for Semantic Web reasoners - an analysis of the DAML ontology library. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, Sanibel Island, Florida, USA.
44. Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1998). The Enterprise Ontology. *The Knowledge Engineering Review, 13(1)*, 31-89.
45. Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review, 11(2)*, 122-147.
46. Weißenberg, N., & Gartmann, R. (2003). Ontology Architecture for Semantic Geo Services for Olympia 2008. *In: Bernard, L., A. Sliwinski and C. Senkler (Eds). Münsteraner GI-Tage, Münster. IfGIprints 18.* 267-283.
47. Weinberger, H., Te'eni, D., & Frank, A. J. (2003). Ontologies of Organizational Memory as a Basis for Evaluation. *11th ECIS'03 European Conference on Information Systems*, Naples, Italy.
48. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, 108-117.
49. Weiss, S. (1997). Glossary for Information Retrieval. Retrieve February 22, 2004, from http://www.cs.jhu.edu/~weiss/glossary.html
50. YoshiKawa, M., & Amagasa, T. (2001). XRel: A path-based approach to storage and retrieval of XML documents using relational databases. *ACM Transactions on Internet Technology, 1(1)*, 110-141.

赴國外研究心得報告

計畫編號	NSC 95－2416－H－004－006 -
計畫名稱	本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(2/2)
出國人員姓名 服務機關及職稱	國立政治大學會計學系所 謝家蘭教授
出國時間地點	2006.08.01-2007.03.31
國外研究機構	UC Berkeley

工作記要：

1. RESEARCH BACKGROUND

Internet has changed the way business conducted between companies worldwide. Firms are now used to exchange business information electronically over Internet. Since the mid-1990s, wave after wave of web technology standards emerge to support the electronic business information exchange. Standards like Extensible Markup Language (XML), Internet Electronic Data Exchange (I-EDI), RosettaNet[1], ebXML[2], Web Ontology Language (OWL), and Semantic Web (SW) surge and sweep electronic commerce worldwide (W3C 2006) (RosettaNet 2006) (ebXML 2006) (OWL 2004). These standards impact on contemporary corporations in many aspects. These standards are proposed to provide a uniform way of business information exchange mechanisms. Semantic not syntactic integration emerges to be the issue that hinders the plan and progress of business-to-business integration electronic commerce (B2Bi EC), which in turn causes time, cost, and reinvention every time there is a change in the public process, there is a change in the standard, and there is a change in the partnership.

The traditional method to tackle the issue can be divided into the programming (ad hoc) approach and the mapping table (syntactic) method. The programming approach solves the problem in a one to one fashion but the result easily becomes the unmanageable "spaghetti" chaos. The mapping table seems to be an easy and convenient approach. However, it only deals with the specific data values not the data definition. An exponentially growing number of trading partners emerge in B2Bi EC. Programming is no longer an effective and flexible way. Mapping table is too primitive and inadequate. The new complexity of data semantic in the business information exchange makes both approaches even harder to tackle the problem (Stojanovic et al, 2002) (Trastour et al, 2003). We believe that Internet growth makes B2Bi climb to a higher level of exchange, that is, the exchange of business meanings and business constraints. A knowledge-intensive and system-to-system semantic integration model and method is in need.

[1] RosettaNet is a consortium of major computer and consumer electronics, electronic components, semiconductor manufacturing, telecommunications and logistics companies working to create and implement industry-wide, electronic commerce and business process standards. RosettaNet is a subsidiary of GS1 US, formerly the Uniform Code Council, Inc. (UCC).

[2] ebXML is a worldwide project initiated and driven by the Organization for the Advancement of Structured Information Standards (OASIS) and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). ebXML is to map out a common framework to enable interoperable electronic commerce and business expressed in XML.

2. RESEARCH ISSUE

Business-to-business integration is to exchange business information between different firms and interoperate the public processes over Internet. The traditional ways of trading include telephone, fax, and email. These approaches introduce faults, redundancies, and wastes. Electronic data interchange is a 1990s and transaction-based approach. However, the change of EDI specification is neither on line or real time. EDI lacks the ability to quickly respond to business changes and suffers from the scalability in the presence of an exponentially growing number of users. Internet EDI is the next stage of B2Bi development. And new B2Bi standards have been proposed based on XML. They indeed provide a more on line and real time method than traditional EDI. However, companies still struggle with the difficulty of heterogeneity and interoperability in the exchange and execution of processes and protocols. In essence, an enhanced approach needs to provide the technology compatibility and the knowledge representation.

Electronic commerce within and across national boundaries is universal. Most firms if not all have problems in one way or another with business process integration and business model interoperability. On both methodological and pragmatic levels, due to increasing diversity in web pages, web services, data sources, and programming languages in all countries, developing an analysis framework of cross national B2Bi resolution is important at international, national, and intra-national levels. This study will develop an analysis framework and a method to explore the way integration and interoperability over schema and semantics can be achieved in B2Bi EC. The dynamics of Internet and intelligence of XML and ontology interplay with inter-organizational context, making it a base for exploring the model and method.

Various approaches have been proposed to study B2Bi issues. However, they lack the process perspective and the semantic representation. Their interoperability is based on adhocracy. Much is needed in the systematic and methodological enhancement. This research intends to tackle the inadequacy of B2Bi standard implementation in forms. An ontology-assisted analysis framework is created to reconcile and represent the conflicts and correspondences in the B2Bi EC issue. Based on the literature, in general, B2Bi framework has three fundamental layers to deal with (Cut et al, 2002) (Falkovych et al, 2003) (Gasevic et al, 2004). They are the communication layer, the content layer, and the process layer. These layers represent the important mechanism and management in B2Bi such as the coupling among partners, the autonomy, and the security. In essence, they mean the specifications of the message formats, the transport protocol, the procedure, and the security mechanism.

3. RESEARCH METHOD

3.1 Analysis-driven Ontology Modeling

The research structure depicted in Figure 1 is the analysis framework we present in the paper to illustrate the model and the method to be developed and deployed in the B2Bi EC standards implementation. The framework is made up of the Unified Modeling Language, the Extensible Markup Language, and the Ontology technologies. Business process interoperability and business data integration are considered the antecedents to B2Bi strategies. More in the framework, a set of analysis procedures are proposed. We analyze the cross national business partners' data schema and process model. We examine the electronic commerce standard in the aspect of data semantics and process semantics. A set of heuristics and rules will be created to represent the above analyzed process models and data schema in form of syntax and semantics. The partners' and the standards' ontologies will be separately developed using the rules and the heuristics. We will merge these ontologies in order to reconcile their conflicts and correspondences. The resulting merged ontologies are tested by the prototype system.

In the end, we hope there is an evolution step to be undertaken to reuse the resulting ontologies. The trading partners can share the domain knowledge in the future standard implementation. The following subsections describe the procedures of the analysis framework and are divided into Step A through Step D. Step A develops the domain ontology of the firm and of the trading partners. Step B creates the domain ontology of the standards. Step C focuses on the ontology knowledge representation for the firm and for the trading partners. Step D creates the ontology knowledge representation of the standards.

[Insert Figure 1 here]

3.2 Step A – Firm Public Process Ontology, "as-is"

A. to analyze the current business process, "as-is"

If we want to analyze the current process, in general, we initiate a meeting. The meeting participants include the process owners and the process users. Through interviewing users, we discover detailed information about the current processes. The detail information contains the process goal, the process flow, the process user role, the process input, the process output and others. This information should be minuted. According to the meeting minutes, we draw the UML diagrams. If we understand the current processes more, we can represent the process as in UML without losing its semantics.

A.1 to design the use case diagram

Before we draw a use case diagram, we have to gather data. We analyze the process actors, the process preconditions, and the process flow to fill out an analysis form. Take the purchase order (PO) as an example. There should be two actors in the purchase order process: buyer and seller. Before the buyer orders something, the seller makes a request for a quote document from the seller first. Then, if the buyer accepts the quote, he sends a purchase order to the seller. When the seller receives the purchase order, the seller confirms the order. This scenario is the common and simple one.

A.2 to design the sequence diagram

In a sequence diagram, we try to discover all messages that are exchanged in a business process and in the purchase order. It can be extracted from the use case diagram and the meeting minutes. In the purchase order example, the PO Request is the first message to be sent from the buyer to the seller. When the seller receives the order request, the seller should check the inventory to determine whether the firm can fulfill that purchase order or not. Then the PO Confirmation is the next message to be sent from the seller to the buyer.

A.3 to design the activity diagram
An activity diagram can show the flow from one activity to another activity. It can represent the detailed process flow. We should find the information from discussion at the meetings so as to develop the activity diagram. We need to discover the detailed actions in the flow, the initial state, and the final state. We then continue the PO example and finish the activity diagram. In this example, we have three actions: request a purchase order, check inventory for this order, and confirm this purchase order.

A.4 to design the class diagram
We try to extract a generic class construct from the use case diagram, the sequence diagram, and the activity diagram. Again, we move on with the PO example. First, we work on the use case diagram. We discover four components: the two actors and the two use cases. We take the two major elements in the use case diagram, Actor and Use Case, to form the two classes: Actor and Activity. Next, we extract the class Message from the sequence diagram, because the sequence diagram describes the message flow and the order flow between the objects. Then, we work on the activity diagram which consists of several actions as described above. The class Action can be extracted.

3.3 Step B – Standard Public Process Ontology, "to-be"

B. to develop the EC-standard-compliant business process
We use four UML diagrams to perform the work such as the use case diagram, the sequence diagram, the activity diagram, and the class diagram. They are utilized to model an EC-standard-compliant business process. The mapping methods between the four diagrams are the same as in Step A. The difference between Step A and Step B is the source of analysis. Step A focuses on the firm existing and current public processes. We have to collect and examine them through interviews and observations. We model the standard processes from B2Bi EC standard specifications at Step B. Some B2B standards have the concept of process, but some do not. If they do not, we should discuss this issue with the trading partners in order to develop a new standard process specification based on the B2Bi EC recommendation. Of course, some B2B standards have adopted UML diagrams to present their standard processes in the specification. We can directly use them.

B.1 to design the use case diagram
We develop the use case diagram based on the B2B standard specification. A B2B standard specification often describes the process purpose and the process definition in the statements. We search and extract the basic components for a use case from the process statements.

B.2 to design the sequence diagram
The B2B standards should specify the sequence of the exchanged messages. The latest standards often adopt the sequence diagram to represent the sequence. Therefore, we use the diagram provided by the standards. If the standards do not use UML diagrams, we still can analyze the sequence of messages in the generic control constructs.

B.3 to design the activity diagram
A B2B standard should formalize the public process flow. Such formalization allows the partners to follow. We do not expect to manage many different process flows with our trading partners in the real world. A B2B standard provides the well-defined process flows. We can extract and formalize the defined process flow from B2B standard specification.

3.4 Step C – Firm Ontology Representation

C.1 to capture the current B2B ontologies
In this study, we propose a heuristics approach to model the ontologies for the firms and the B2Bi EC process and message. We build the ontology so as to describe the firms' B2B domain knowledge. This domain ontology contains the basic classes and properties. Every business process should fit in an ontology definition. We define the basic B2B components and properties.

C.2 to model the current business document ontology
We analyze the core of the public processes performed between B2B partners. The core means the message analysis. We then need to develop a process ontology based on the semantics of the message analysis. The semantics refer to the context, the meaning, the terminology, and the relationship in the business document exchange process.

C.3 to reconcile the current business constraints
We may have constraints on each entity, each message, and each process. These constraints have to be converted into OWL. After business process and document ontology being created, we move on to build the EC standard ontology.

3.5 Step D – Standard Ontology Representation

D. to capture the EC standard's ontologies
To build the EC standard ontology, we need to find out the B2B process specifications and their business documents. The definition of each business document is often encoded as DTD or XML Schema. We use the schema to create these EC standard ontology.

D.1 to design the EC standard's process ontology
Notice that not all EC standards require implementing all elements in the specifications. Only the standards that are required in the partnership will be converted into OWL classes. In this section, we develop a set of heuristics to address the issue.

D.2 to model the EC standard's document ontology per partner
The way to model the standard document ontology is the same as above. We extract the

data definition in standard to do the conversion.

D.3 to reconcile the EC standard's constraints
The standard may have constraints on each entity, each message, and each process. The trading partners in between may have their own practice constraints. We extract to collect them and use the above procedures to convert them into OWL object properties.

3.6 Step E - Ontology Merge

When initiating and implementing a new B2B initiative, we deal with new B2B EC standards. Different business partners and different settings occur. Though we have the existing ontology in the ontology repository, these new differences cause the ontology mismatch and inconsistency. We need to resolve and merge these ontologies including functions of (a) reading in ontologies, ontology updates, and adaptations, (b) viewing a specific version or a variant of an ontology, (c) differentiating ontologies, (c) checking the inconsistency in the ontology combination.

In essence, the key to merge is to discover the differences and to generate the correspondence rules between ontologies. The differences are like the instances of the changes of class name, the addition or deletion of classes, the addition or deletion of properties, and the mergence or split of classes. Though it is common to find new conflicts and differences between new trading partners, there are common parts as well to take advantage of as we discuss on the repeat rate and reuse. The hard part is the more heterogeneous the ontologies are, the larger extent of change to be implemented between the old and the new processes. Analysis gives us the parts of the process to be changed and installed in the coming implementation. We adopt the ontology of the B2B standard and merge the B2B standard ontology based on the merge rules as listed in Table 1.

[Insert Table 1 here]

3.7 Step F – Ontology Representation

We have described the ontology representation in Step C and Step D. The technique is used in the merged ontology.

3.8 Step G – Ontology Test

To verify the ontologies merged, we consider two issues, the syntactic test and the semantic test. We test the syntactic of ontology through the ontology tool. It automatically validates the inconsistency of syntax. The semantic test is to discover the inconsistency between the database schema, the business processes, and the old version of ontology. We extract the database schema and examine the consistency between the business ontology. We compare the consistency between the trading agreements in order to specify the business rules. We analyze the differences between the new ontology and the real environment. The analysis results will be used to adjust the business process to refine the merged ontology.

Figure 2 illustrate the Steps. As described above, we first discover the ontology requirements in Step A and Step B. We then create the ontology from Step C and Step C.

We merge ontologies in Step E. Step F gives a merged representation. Step G tests the syntax and semantics.

[Insert Figure 2 here]

4. AN EXPERIMENTAL STUDY

4.1 A Prototype System

In this research, we have developed an experimental prototype that implements the presented B2Bi ontology development method. This prototype is built to facilitate the illustration of the feasibility and the validity of the method. In this section, we demonstrate an application of the prototype in two main electronic commerce standards, the RosettaNet, a worldwide and vertical B2B standard; and the ebXML, an OASIS and UN sponsored and horizontal B2B standard (RosettaNet 2006) (ebXML 2006) (OASIS 2005) (Hofreiter et al, 2002). Both standards are installed worldwide because they cover the diverse electronic commerce practices. We use these two major standards as the starter experiments in the illustration that our new method is feasible and valid. Preliminary experimental results show that this ontology-assisted method gives a viable resolution to the long-standing semantic and syntactic issue in the implementation of electronic commerce standards.

In both experiments, we choose the purchase order process as the baseline to illustrate a live case study of a large scale semiconductor component distributor. The live case company is called company W. Company W is the number one distributor in the Asia Pacific region since 2004. The purchase order process is the main business process in their B2B EC. Company W since 2004 became quite concerned with the various EC standards to be installed and among its cross-national suppliers and customers. The time and efforts grow exponentially. At the same time, Company W is troubled by a needed lift to the next level of performance of global supply chain management. And B2Bi is the bottleneck of the performance and becomes the compelling reason to reengineer the electronic commerce architecture.

An ontology-assisted B2Bi eCommerce prototype architecture as shown in Figure 3 is developed in the experimental study. The B2Bi platform allows enterprises to exchange business documents over Internet. It provides various and common B2B protocols to connect the trading partners. It provides the ability to streamline the business process and the adapters when linking with the various enterprise information systems.

[Insert Figure 3 here]

We build the research model of Step A through Step G into a prototype system. The layers in the system are illustrated in Figure 4. The system provides a number of main functions such as the DTD Importer, the Ontology Editor, and the Ontology Display. Figure 5 illustrates the structure of the functions.

[Insert Figure 4 here]

[Insert Figure 5 here]

4.1.1 DTD Importer

DTD importer parses the DTD that specifies the document format and transfers DTD to ontology. The user can enter the output OWL file as shown in Figure 6. This feature will transfer the file automatically. We will produce two groups of class and one group object property. The classes are B2B_DataEntity and B2B_ComposedDataEntity. The object property is B2B_BusinessProperty. The DTD Importer will differentiate all entities from DTD file base on the nature.

[Insert Figure 6 here]

The DTD Importer also provides an ability to parse the entity's metadata. Through parsing the metadata, we can enrich our document ontology. This program will read the entity information using a batch approach as shown in Figure 7.

[Insert Figure 7 here]

4.1.2 Ontology Editor

We build a process ontology template. The basic classes and properties of the B2B process can use this template to develop the ontology as shown in Figure 8 and Figure 9. Business constraints also can be edited through the ontology.

[Insert Figure 8 here]

[Insert Figure 9 here]

4.2 A RosettaNet Experiment

RosettaNet consortium (RosettaNet Consortium, 2004) is a non-profit consortium of more than 500 organizations working to create, implement and promote open eBusiness standards and services. RosettaNet tries to establish a common language and a standard processes for the electronic sharing of business information. In order to implement the experimental scenario, we install a set of RosettaNet core specifications. We will explain each in the following sections. These core specifications include RNIF, PIP, and Dictionary.

We chose the RosettaNet Partner Interface Process™ (PIP) 3A4, the purchase order request process, to be the experimental public process, which is mostly implemented and installed. Purchase order process is corresponding to Company W's sales order flow. The PIP3A4 specification provides the details of the purchase order process property and constraint. The structure and content of the business documents to be exchanged in RosettaNet is specified in DTD and XML Schema. We need to build the process ontology from the PIP specification and to create the document ontology from the DTD and message guidelines.

Step A - to analyze the existing business process

We analyze the current business process of purchase order between Company W and its buyers. In the use case diagram, we see two kinds of participants: Company W and buyers. Company W has the partner role of “Seller” and customers play the role of “Buyer”. In the sequence diagram, we see two business documents that are exchanged. They are the “Customer Order” and “Customer Order Ack”. There are three main activities in the sales order flow such as “Send A Customer Order”, “Check Inventory” and “Confirm Customer Order”.

Step B - to develop the B2B EC-standard-compliant business process

The RosettaNet specifies PIP3A4 exchanging three messages such as “PurchaseOrderRequest”, “PurchaseOrderConfirmation” and “ReceiptAcknowledgment”. PIP3A4 further specifies two more activities, that is, “Request Purchase Order” and “Confirm Purchase Order”. They represent the standards to be complied with.

Step C - to represent the existing firm ontology

C.1 to design current business process ontology

Company W and its buyers are B2B_Partner. We add them as the instances of B2B_Partner. The instance’s naming rule should be pre-established. We create the domain ontology of each business process for Company W and its buyers. In Figure 10, we show the B2B_Process “Customer Order”.

[Insert Figure 10 here]

C.2 to model current business document ontology

We use the DTD importer in the prototype to convert and store the DTD files into document ontology repository. The converted data result is shown in Figure 11.

[Insert Figure 11 here]

Step D - to represent B2B EC standard ontology

D.1 to design RosettaNet process ontology

The PIP3A4’s ontology is built from the standard specification. The PIP restrictions must be considered. For example, the PIP3A4 specification defines when to enable a buyer to issue a purchase order and when to enable a seller to acknowledge the receipt of order, and even down to the line item level. No matter how and if the order is accepted, rejected, or pending. The provider's acknowledgment must include the information about delivery expectation. Further, when a provider acknowledges that a product line item on the purchase order document is “pending”, the provider must later use PIP3A7, "Notify of Purchase Order Acknowledgment" to notify the buyer when the product line item is either finally accepted or rejected. The PIP specification also describes the process start state to be one of the following: Purchase Order Request Exists, TPA Exists, Requesting Partner Approved, Responding Partner Approved, Buyer Authorized, Purchase Order Request Valid, or Purchase Order Request Non-Repudiated as shown in Figure 12. One of the above must be returned as the initial date started.

[Insert Figure 12 here]

D.2 to model RosettaNet document ontology
The PIP3A4's DTD files and message guideline give the details of each entity in the standard document. We again use the DTD importer to convert and load the specification. We edit the constraints for each entity. The DTD importer can intelligently determine and set up the business property domain and domain range properly. A RosettaNet PIP definition can be added as the instance's comment as shown in Figure 13.

[Insert Figure 13 here]

Step E - to merge ontology
In the merge of ontologies, the purchase order has a unique number as the identity of the order. In this experiment, the current purchase order number is named "orno". However PIP3A4 uses the field "ProprietaryDocumentIdentitfer" as the purchase order number. We resolve the inconsistency via link analysis.

When we generate the current business ontology and the B2B standard ontology, we can merge them in the system. Although Protégé provides the merge, we need to adjust the detail correspondence rules to link the relationship between two ontologies. The ontology editor provides the test function helps us check the ontology consistency. The purchase order usually has a unique number as the identity of the purchase order. The current order number is named "orno". However, PIP3A4 names the field as "ProprietaryDocumentIdentitfer". We use the *owl:sameAs* to link these two classes as equivalent.

Step F - to represent ontology
We generate a set of HTML files that contain the content of ontology. Users browse the ontology in a user-friendly interface. It minimizes the risk of ontology to be altered. With the hyperlink, we can trace any related classes and properties.

Step G - to test ontology
We validate the ontology with the Protégé test ontology function.

4.3 An ebXML Experiment
The second experiment we have conducted is a realization of ebXML in the case of purchase order. ebXML (ebXML, 2004) was started in 1999 and developed by the Organization for the Advancement of Structured Information Standards (OASIS). OASIS is a non-profit, international consortium that drives the development, convergence, and adoption of eBusiness standards. The consortium produces more Web services standards than any other organization along with standards for security, eBusiness, and standardization efforts in the public sector and for application-specific markets. Founded in 1993, OASIS has more than 3,500 participants representing over 600 organizations and individual members in 100 countries (OASIS Consortium, 2004). ebXML is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. By using ebXML, companies can exchange business messages, conduct trading relationships, communicate data in

common terms and define and register business processes. In order to implement the experimental scenario, we install a set of ebXML core specifications. We will explain each in the following sections. These core specifications include the business process, the core component, and the collaboration protocol profile and agreement.

Steps A, B, C from the method are similarly applied in the ebXML experiment. The main difference is in Step D where we explain the procedure of transition. The difference is when we try to model the ebXML standard specification in ontology.

D.1 to design ebXML process ontology

ebXML uses Business Process Specification Schema (BPSS) to model business processes. In modeling the process ontology, we utilize the sequence diagram and the activity diagram. The BPSS specification specifies a Business Transaction, a Business Document flow for the Business Transaction, a Binary Collaboration, and then a Choreography for the Binary Collaboration. A Business Transaction in ebXML is the basic transaction unit between two partners. It consists of a Requesting Business Activity and a Responding Business Activity. A Binary Collaboration is always executed between two roles. They are called the Authorized Roles because they represent the actors that are authorized to participate in the collaboration. A Binary Collaboration consists of one or more Business Activities. These Business Activities must be conducted between the two Authorized Roles in the Binary Collaboration. A Choreography is an ordering and sequencing of Business Activities within a Binary Collaboration. The choreography is specified in terms of the Business States and the Transitions between these Business States. In the purchase order process example, we know it has two activities: Purchase Order Request Action and Purchase Order Confirmation Action. Both can be extracted from the BPSS sample file. We further know that if the activity is authorization required or non repudiation required. Below is the converted OWL scripts.

D.2 to design ebXML document ontology

The ebXML document ontology is created in use of the ebXML core component specification. The core component can be used to create the classes and the properties and be converted into ontology. The ebXML document ontology needs to incorporate existing DTD or existing XML Schema in order to support each component conversion. The ebXML core component in the standard ontology corresponds to the basic data entity in the domain ontology. The core component represents the same meaning as the data entity in the domain ontology. The ebXML aggregate information entity, on the other hand, stands for the composite data entity in the domain ontology. Because BPSS is also UML-based, the set of UML diagram such as the use case diagram, the class diagram, the sequence diagram, and the activity diagram our method recommends are adopted. Hence, the correspondence and reconciliation will occur early at the analysis phase and will be carried out in the merge of standard ontology and domain ontology. For ebXML, the actual document definition is achieved using the ebXML core component specifications or by some methodology external to ebXML. They in turn are converted into a DTD or a XML Schema. BPSS specifies the specification name as “PurchaseOrderReques.dtd”.

We only need to get this DTD file and import it. A business document has three types of

components. They are the basic data entity, the composite data entity, and the business property. The ebXML consists of: Core Component Type (CCT), Basic Information Entity, and Aggregate Information Entity. CCT are core components that carry the actual value and have no business meaning on their own. A Basic Information Entity is a singular concept that has a unique business semantic definition. A Basic Information Entity adds a semantic meaning to a single datatype or a CCT. When CCTs are reused in a business context, they become Basic Information Entities. An Aggregate Information Entity contains two or more Basic Information Entities or Aggregate Information Entities that together form a single business concept. Each Aggregate Information Entity has its own business semantic definition.

5. DISCUSSION

5.1 Research Implication

The first set of literature investigating the issues of aligning the business processes with the business-to-business electronic commerce standards are described in (Omelayenko, 2001) (Stojanovic, Maedche, Motik, and Stojanovic2002) (Bussler, Fensel, and Maedche, 2002). They represent the earlier efforts working on the programming to approach the interface between the trading processes and new standards. This ad hoc programming approach was primitive and exploratory. Later in the literature survey, (Ding, Fensel, Klein, Omelayenko, and Schulten, 2004), the time when there were more business-to-business middleware systems in the marketplace, the researches gear toward to solve the issue of conversion and hub in the middleware system. Another important stream of literature presented in (Choy and Kim, 2004) (Cao, Zhang, and Seydel, 2005) addresses the integration aspect of the alignment. These studies relate to ours. They perceive the issue as a process and performance issue in the supply chain management. In (Hsieh, Lai, and Shi, 2006) (Iyer, Gupta, Johri, 2005) the process issue is further examined with mathematics to model the business operation.

In 2004 and 2005, we tested new PIPs including PIP3B18, PIP3B2, and PIP4C1 in the live case experiment. Each PIP took one month to three months instead of six months as in the prior years. The IT division assigned four full time system engineers instead of six to deliver the implementations. Figure 14 shows the equivalent classes generated from the PIP3A4 standard. Figure 15 gives the equivalent properties in the test.

[Insert Figure 14 here]

[Insert Figure 15 here]

5.2 Managerial and Technical Implications

We summarize the managerial and technical implications into five points.

Ontology is a more powerful technology on semantics and context. A mapping table is simple but lacks the ability to scale. It is a way of “mapping of terms” not an approach to “mapping of sense”. Ontology allows systems to discern the “one to one”, the “one too many”, and the “many to many” correspondences. It allows the systems to undertake the complex conversion such as the situation when there are same terms but with different meanings; different terms but with same meaning; different terms

but with different meanings yet close. Ontology can support an automation of the evolution of the terms, the concepts, and the relationships. The relationships between the new terms and the corresponding old terms can update automatically. Ontology is the base of reasoning.

The analysis framework and ontology enables the deployment of a new B2Bi EC standard initiative to be installed and operated in an effective and efficient fashion. Though RosettaNet PIP message defines many business entities, but there are many repeating entities in different PIPs. RosettaNet PIP3B2 is an example of the shipment notification process which specifies 120 business entities and properties. PIP3A4 is another example that has 143 business entities. However, there are 59 repeating business entities between these two PIPs. The repeat rate is above 49%. In fact, the more related the processes are, the higher the repeat rate will be. We must take advantage of the repeat rates. The repeat rate of entities between PIP3A4 and PIP3A8 is as high as 92%, almost identical. As new processes are continuously and constantly added to our ontology, the ontology must become more robust. The work of ontology creation becomes more automatic and less labor intensive. At the same time, the new knowledge extracted from the new ontology can be captured. Trading partners regularly collaborate and contribute to the reconciled and merged ontology which in turn forms a semantically rich repository in support of the reasoning, the inference, and the search of organizational learning. By enhancing and enriching the shared ontologies, we can deploy a new B2Bi EC initiative in a more effective and efficient manner.

A special function to transfer data model from DTD and XML Schema to OWL is useful. A prototype of parser and converter to handle XML documents in the creation of ontology is needed.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this paper, we have presented an ontology-assisted analysis method and alignment model in the implementation of the business-to-business electronic commerce standard specification over the existing trading partners' public processes in the syntactic and semantic integration and interoperability. An application of the Unified Modeling Language is made to analyze the public process in the domain and in the standard. Terms, concepts, relations, and links are created from the analysis results and converted into an ontology representation. Web ontology language is introduced to formulate the analyzed knowledge and experience to align the domain and the standard. There are correspondences and conflicts in the process of alignment. They are resolved via the shared and reusable ontology which is a convergence of the domain ontology and the standard ontology. The converged and shared ontology is achieved via a set of rules and heuristics that are created in the research. The key of success in the business-to-business integration electronic commerce lies on the ability to accomplish the process interoperability and the schema comparability. Three main tasks have to be achieved to fulfill the requirements. In this research, we have constructed a prototype to implement the method. The prototype is used to illustrate the feasibility and validity of the method. A set of starter experiments has been conducted in use of a straight through example of a

purchase order process in the alignment with the RosettaNet standard and the ebXML standard. The starter experiment serves as the baseline to demonstrate the method is feasible and valid. The three main things we have accomplished in the research are:

Identifying the main components of knowledge and experience to be reconciled and to be represented in the alignment of the standard ontology and the domain ontology.

Developing the set of rules and heuristics for the ontology correspondence and reconciliation.

Designing an experimental prototype to implement the method and to demonstrate the feasibility and the validity by selecting two main electronic commerce standards as the baseline test. The RosettaNet experiment represents the vertical electronic commerce standard. The ebXML experiment stands for the horizontal electronic commerce standard.

6.2 Future Research Work

The future research work will continue to explore the complex issues of the alignment and automation between domains and standards. Some of the immediate tasks to be undertaken in our study include:

Enhancing the ontology search and inference capability. As the rule base and heuristics base grow, search and inference engine become slow in the ontology management. Tuning and enhanced rules must be developed.

Upgrading the DTD importer to import XML Schema and to enable the conversion between XML Schema and OWL representation. This will solve the version control issue.

Conducting more diverse and complex experiments in terms of scale and scope. More experiments will be conducted in the public processes of receiving and payment that are closely related with the public process of purchase order. RosettaNet and ebXML will still be the main standards.

REFERENCES

1. Baclawski, K., Kokar, M.K., Kogut, P.A., Hart, L., Smith, J., Letkowski, J., & Emery, P. (2002) “Extending the Unified Modeling Language for Ontology Development”, Software and Systems Modeling, Vol 1 No 2, pp. 142-156.
2. Berners-Lee, T., Hendler, J., & Lassila, O. (2001) “The Semantic Web”, Scientific American, Vol 284 No 5, pp. 34-43.
3. Bird, Linda, Andrew G., & Terry H. (2000) “Object Role Modelling and XML-Schema”, ER2000.
4. Bose, R. (2006) “ Understanding Management Data Systems For Enterprise Performance Management ”, Industrial Management and Data Systems, Vol 106 No 1, pp. 43-59.
5. Bussler, C. (2001) “Semantic B2B integration”, ACM SIGMOD Record, Vol 30 No 2, pp. 625.
6. Bussler, C., Fensel, D., & Maedche, A. (2002) “A Conceptual Architecture for Semantic Web Enabled Web Services”, ACM SIGMOD Record, Vol 31 No 4.
7. Cao, M., Zhang, Q.Y., Seydel, J. (2005) “B2C e-commerce web site quality: an

empirical examination", Industrial Management & Data Systems, Vol 105 No 5, pp. 645-661.
8. Choy, L. Y., Kim, H. T. (2004) "A process and tool for supply network analysis", Industrial Management & Data Systems, Vol 104 No 4, pp. 355-363.
9. Cranefield, S., & Purvis, M. (1999) "UML as an Ontology Modelling Language", In Proceedings of 16th International Joint Conference on Artificial Intelligence on Workshop on Intelligent Information Integration.
10. Cranefield, S. (2001) "Networked Knowledge Representation and Exchange Using UML and RDF", Journal of Digital Information, Vol 1 No 8.
11. Cut, Z., Jones, D., & O'Brien, P. (2002) "Semantic B2B Integration: Issues in Ontology-based Approaches", ACM SIGMOD Record, Vol 31 No 1, pp. 43-48.
12. Decker, Stefan, Sergey M., Frank V. H., Dieter F., Michel K., et al. (2000) "The Semantic Web: The Roles of XML and RDF", IEEE Internet Computing, Vol 4 No 5, pp.63-64.
13. Ding, Y., Fensel, D., Klein, M., Omelayenko, B., & Schulten, E. (2004) "The Role of Ontologies in eCommerce", Handbook on Ontologies, Staab S. and Studer R., Springer.
14. ebXML BPSS (2006) http://www.ebxml.org/
15. Gasevic, D., Djuric, D., Devedzic, V., & Damjanovic, V. (2004) "Converting UML to OWL Ontologies", Proceedings of the 13th international World Wide Web Conference, pp. 488-489.
16. Gulledge, T. (2006) "What is integration?," Industrial Management & Data Systems, Vol 106 No 1, pp. 5-20.
17. Helo, P., Szekely, B. (2005) "Logistics information systems: An analysis of software solutions for supply chain co-ordination", Industrial Management & Data Systems, Vol 105 No 1, pp. 5-18.
18. Hunag, C.J., Amy J.C. Trappy, Yao, Y.H. (2006) "Developing an agent-based workflow management system for collaborative product design", Industrial Management & Data Systems, Vol 106 No 5, pp. 680-699.
19. Hsieh, C.T., Lai, F.J., Shi, W.H. (2006) "Information orientation and its impacts on information asymmetry and e-business adoption: Evidence from China's international trading industry", Industrial Management & Data Systems, Vol 106 No 6, pp. 825-840.
20. Iyer ,L. S., Gupta, B., Johri, N. (2005) "Performance, scalability and reliability issues in web applications", Industrial Management & Data Systems, Vol 105 No 5, pp. 561-576.
21. Kogut P., Cranefield S., Hart L, Dutra M., Baclawski K., Kokar M., & Smith J. (2002) "UML for Ontology Development", The Knowledge Engineering Review, Vol 17 No 1, pp. 61-64.
22. Lesjak ,D., Vehovar, V. (2005) "Factors affecting evaluation of e-business projects", Industrial Management & Data Systems, Vol 105 No 4, pp. 409-428.
23. Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A.H.H, & Elmagarmid, A.K. (2003) "Business-to-Business Interactions: Issues and Enabling Technologies", The VLDB Journal, Vol 12 No 1, pp. 59-85.
24. OASIS Consortium (2005) http://www.oasis-open.org
25. Object Management Group (2005) http://www.omg.org

26. Omelayenko, B. (2001) “Preliminary Ontology Modeling for B2B Content Integration”, Proceedings of the 12th International Workshop on Database and Expert Systems Applications, pp. 7-13.
27. Rahm, Erhard & Philip A. B. (2001) “A survey of approaches to automatic schema matching”, The VLDB Journal, Vol 10, pp. 334-350.
28. RosettaNet Consortium (2006) http://www.rosettanet.org/
29. Stojanovic L., Maedche A., Motik B., & Stojanovic N. (2002) “User-Driven Ontology Evolution Management”, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pp. 285-300.
30. Web Ontology Language (2004) http://www.w3c.org/2004/OWL/
31. Zhao, F. (2004) “Management of information technology and business process re-engineering: a case study”, Industrial Management & Data Systems, Vol 104 No 8, pp.674-680.

Figure 1: An Ontology-assisted B2Bi EC Alignment and Management Framework (This Research)

Reuse, Manage, and Evolution

Cross Trading Partners’ Biz Model and Process

Analyze EC Standards’ Data and Process using UML

Merge Ontologies, “to-be”

Test XML and Ontologies

Model Standard Ontology using UML

Model Public Ontology using UML

Analyze Partners’ Data and Process, “as-is” using UML

Represent XML and Ontologies

EC Standards’ Biz Model and Process

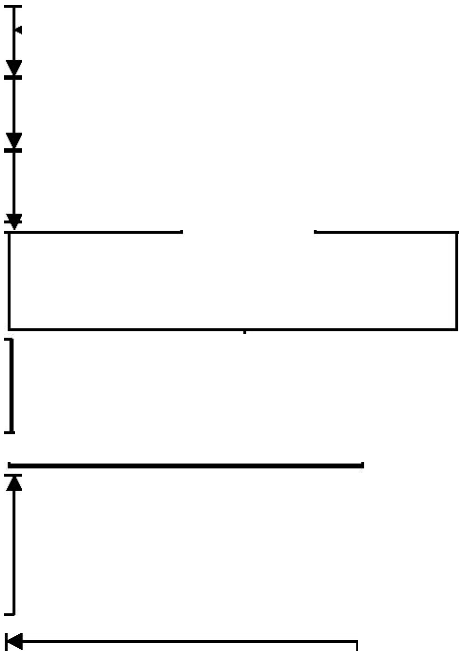


Figure 2: An Ontology-assisted B2Bi eCommerce Alignment Framework (This Research)

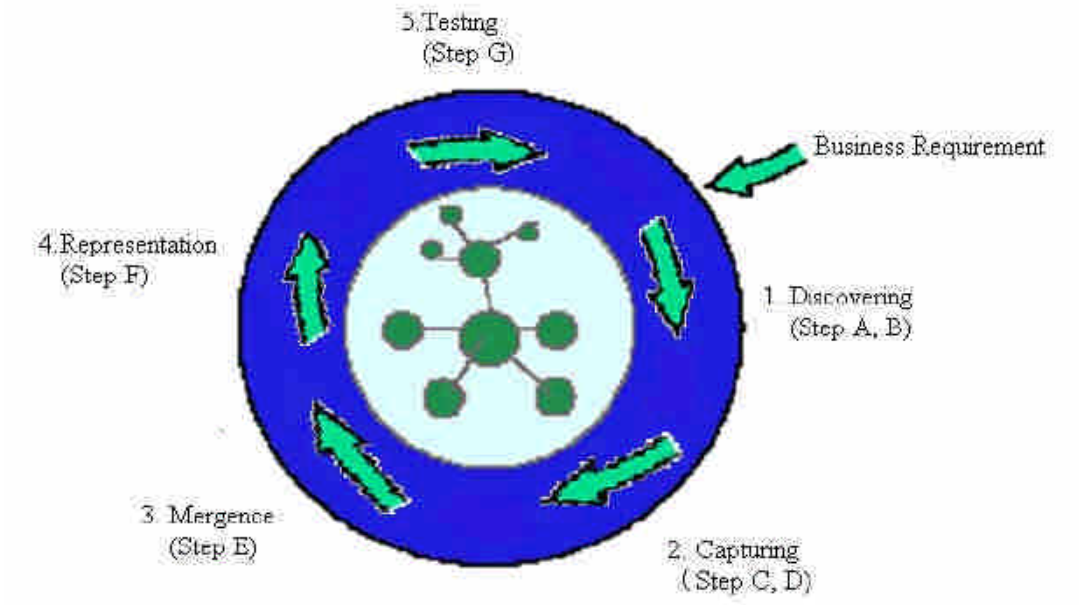


Figure 3: A Prototype Architecture of An Ontology-assisted B2Bi eCommerce Platform (This Research)

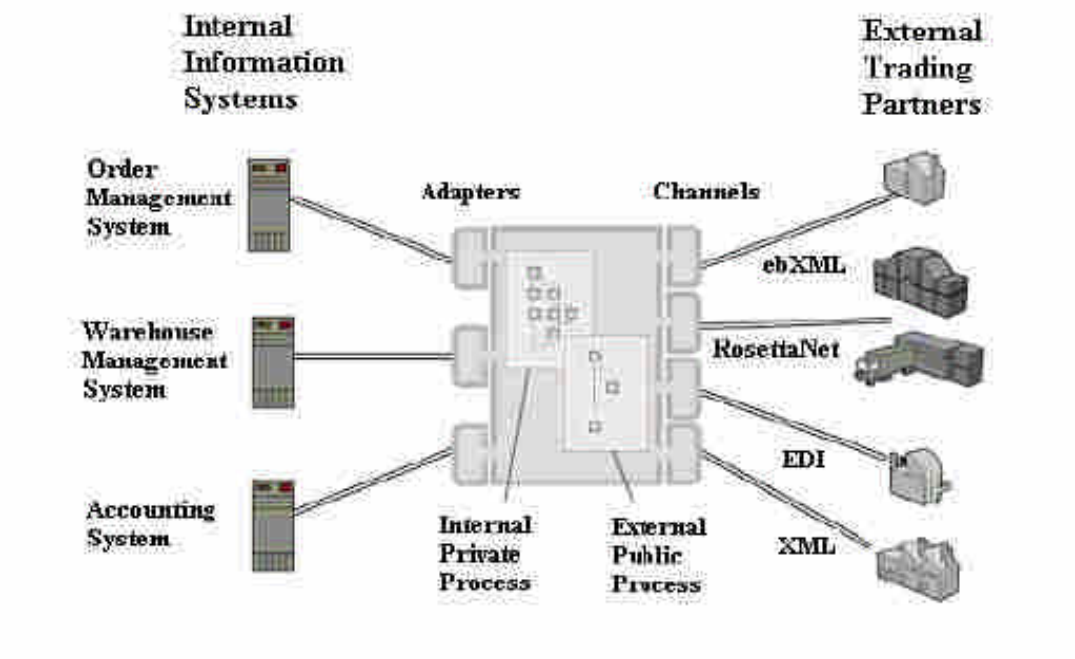


Figure 4: Layers in the Prototype System (This Research)

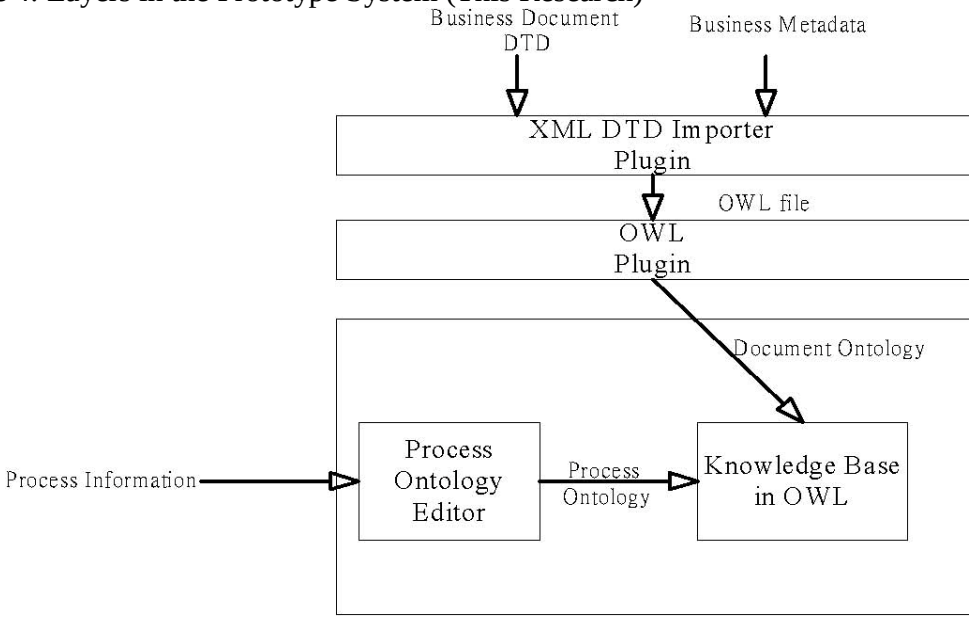


Figure 5: Main Functions of the Prototype (This Research)

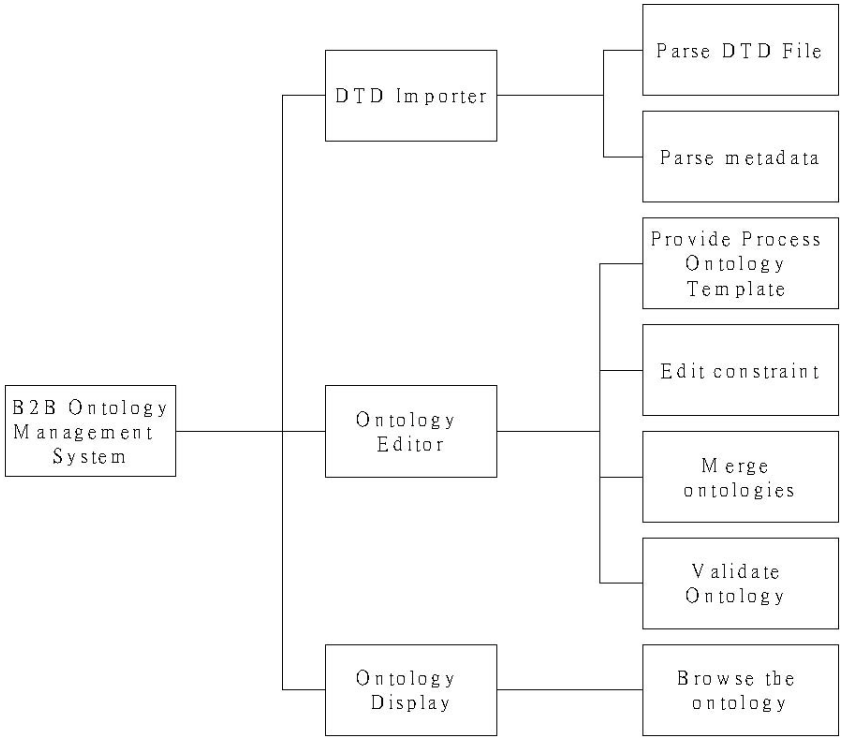


Figure 6: The B2B DTD Plug-in (This Research)

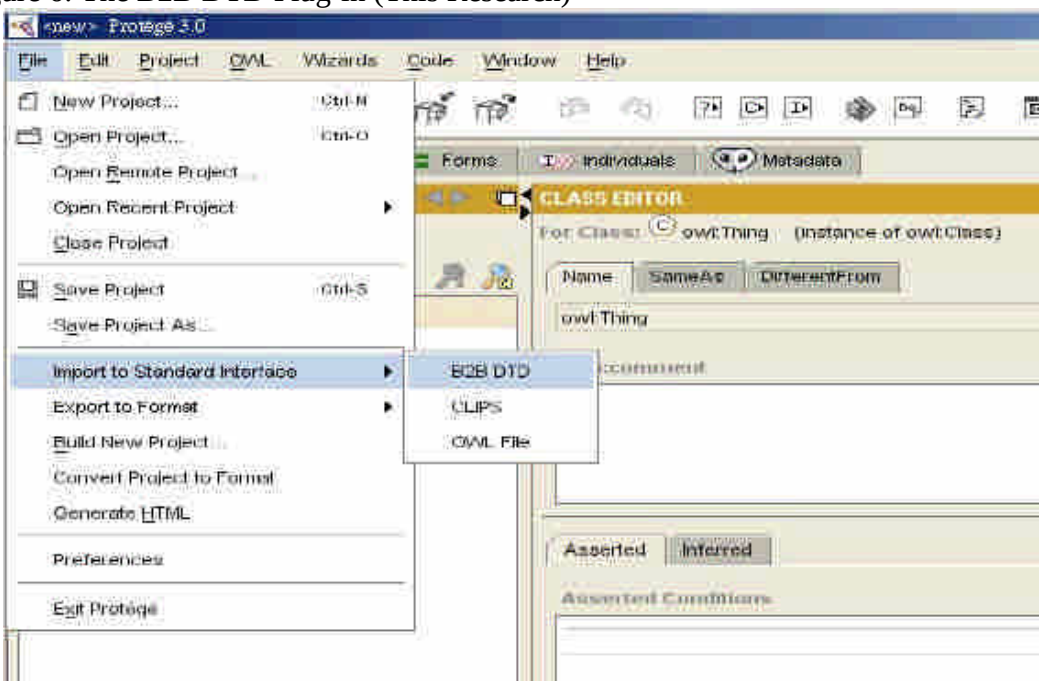


Figure 7: The B2B DTD Importer (This Research)

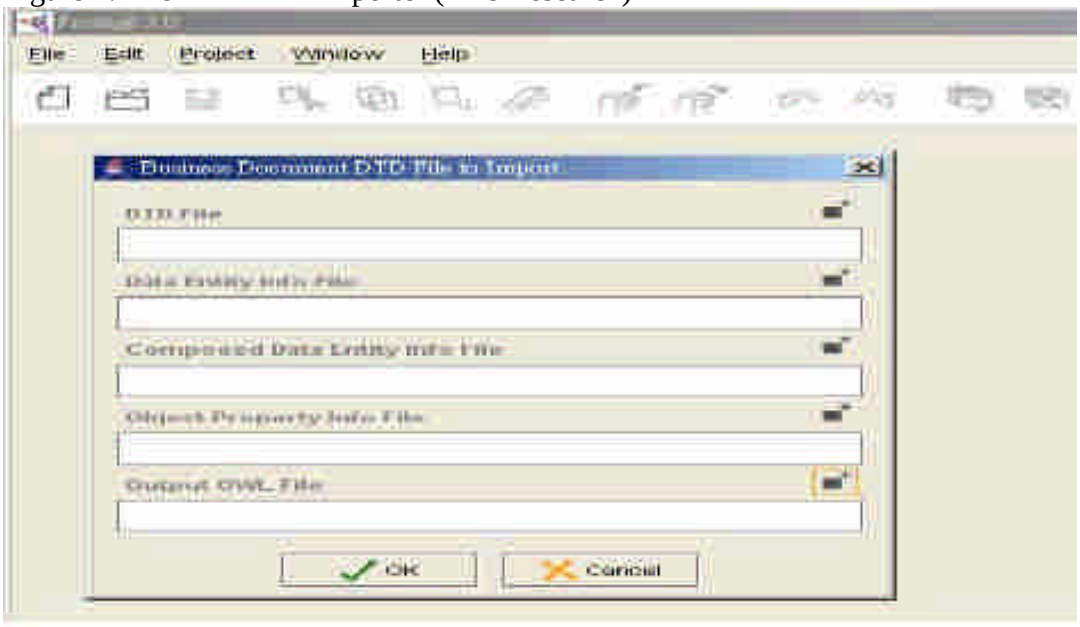


Figure 8: The Basic Classes of a Process Ontology (This Research)

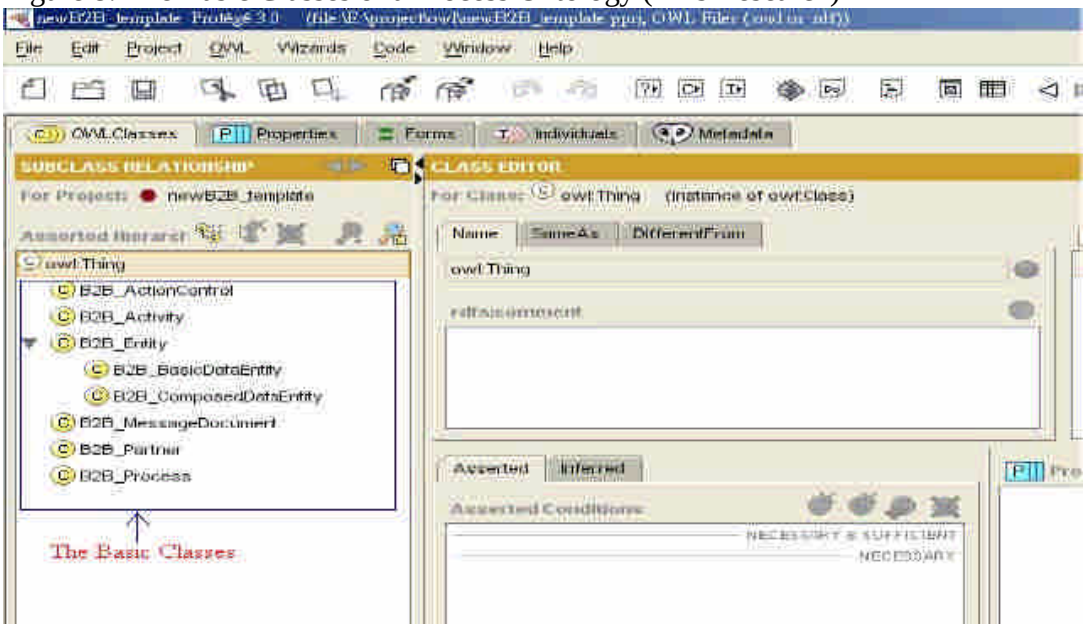


Figure 9: The Basic Properties of a Process Ontology (This Research)

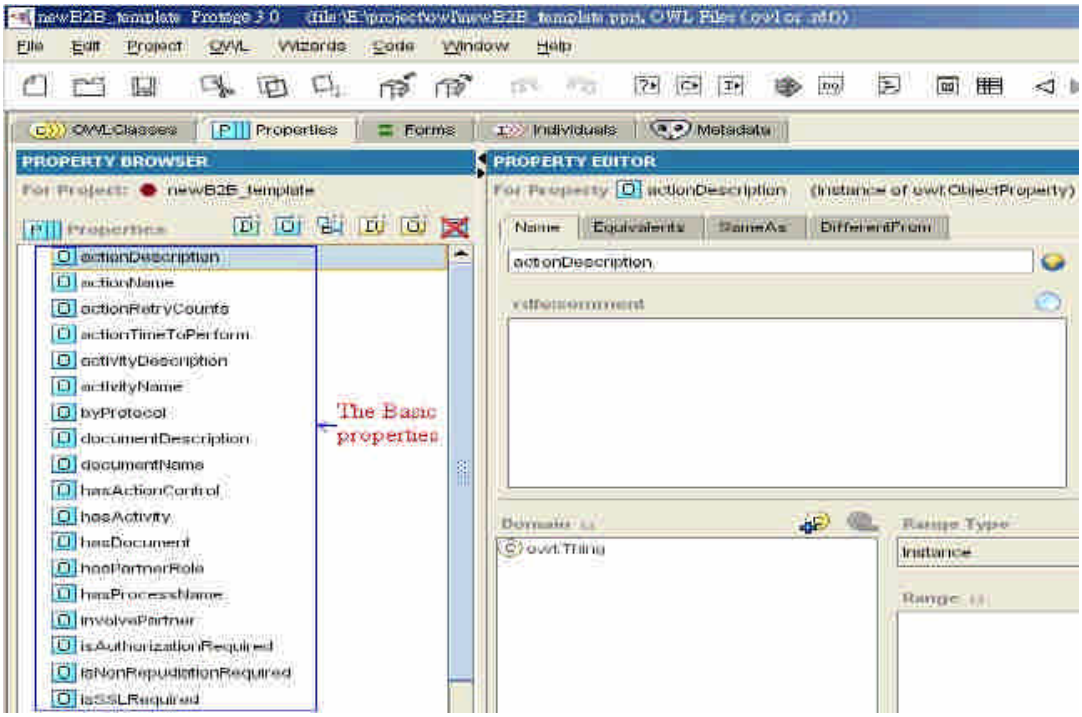


Figure 10: Ontology Instance Creation (This Research)

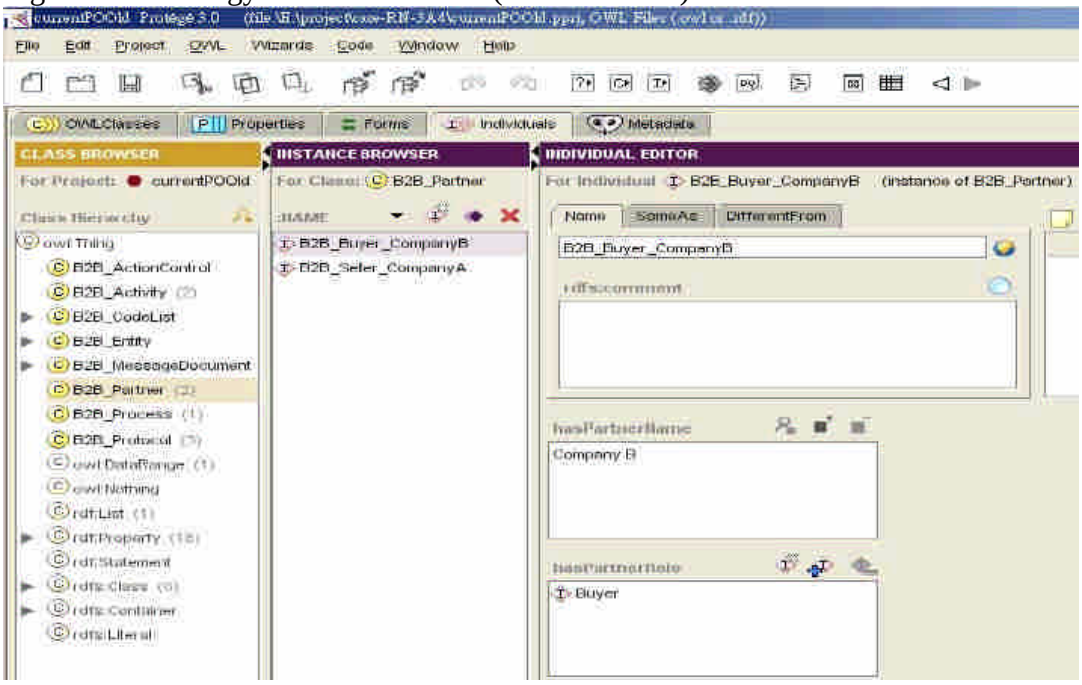


Figure 11: Existing Public Process Ontology (This Research)

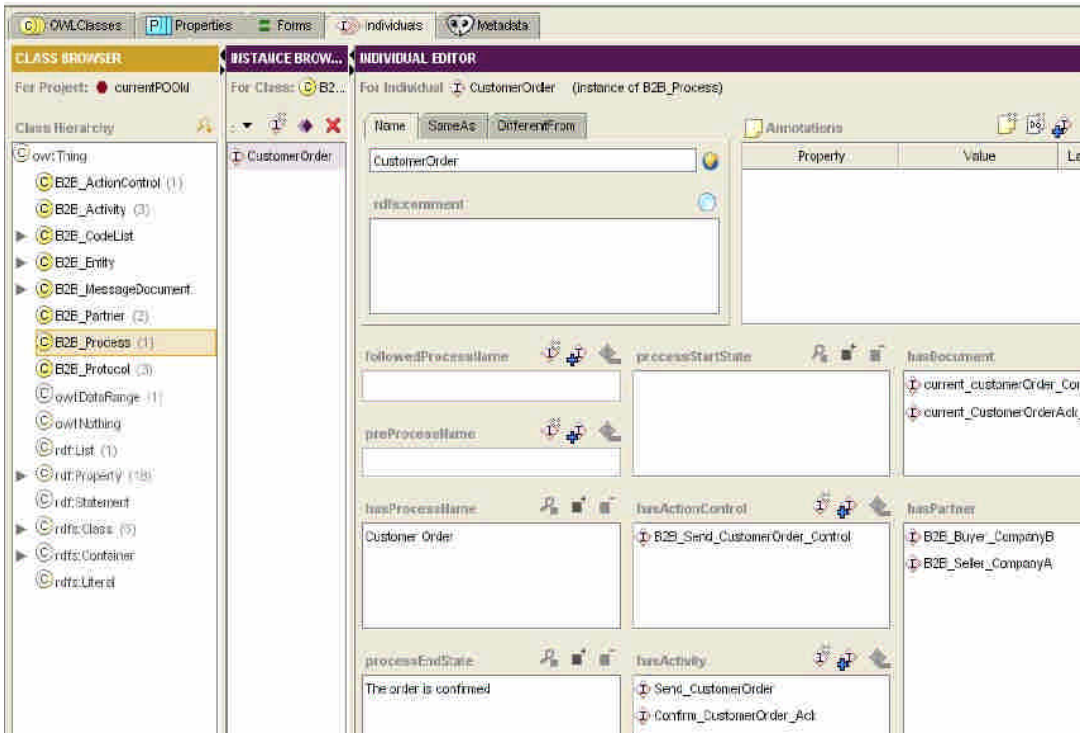


Figure12: Newly Generated Classes and Properties (This Research)

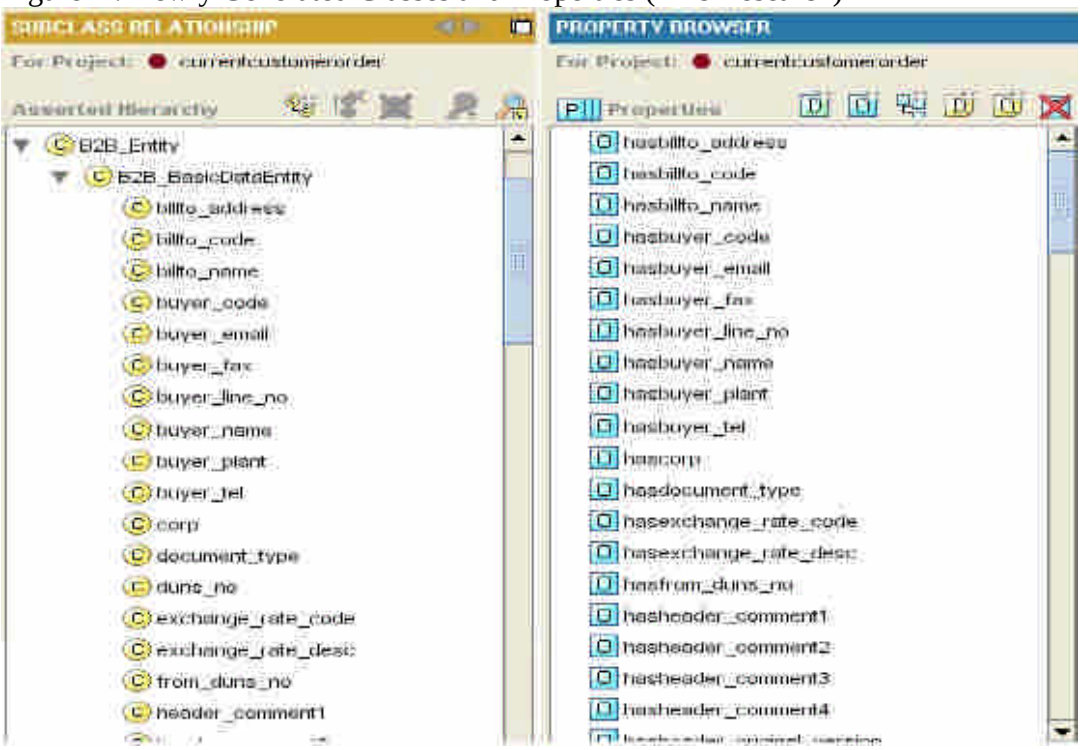


Figure 13: Created Instances of PIP3A4

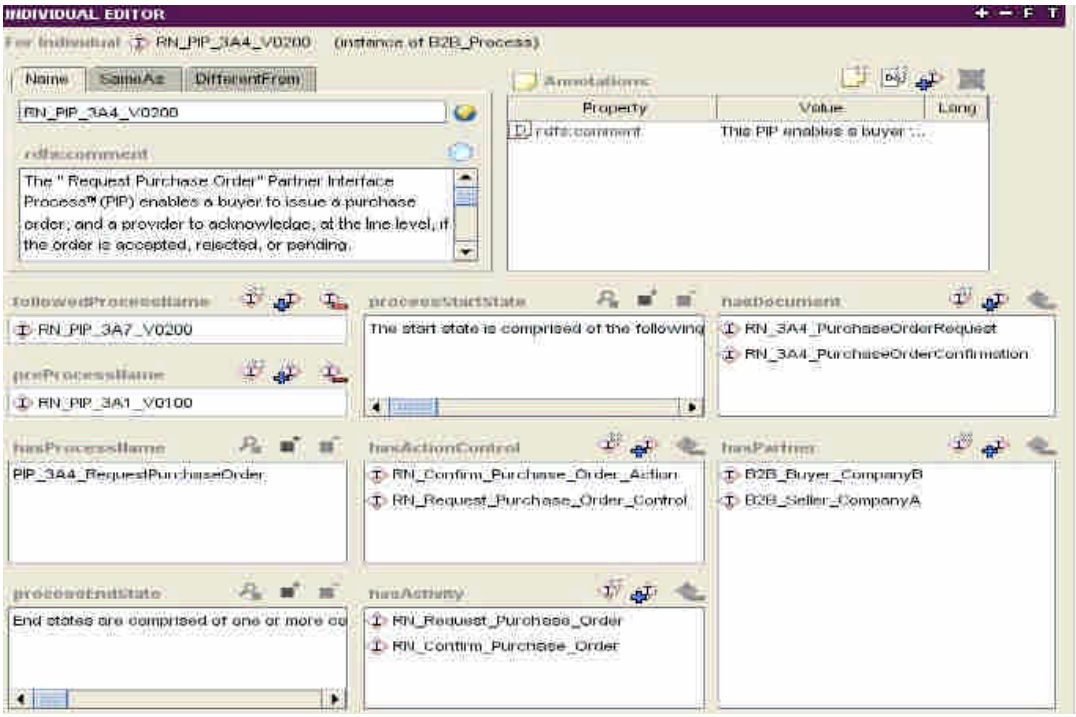


Figure 14: Equivalent Classes

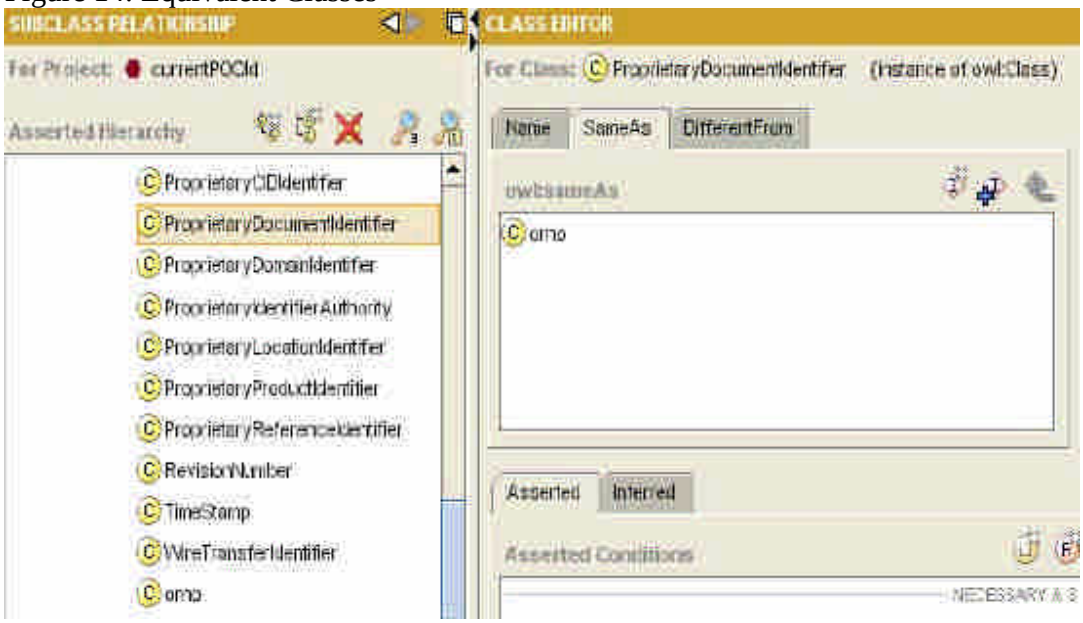
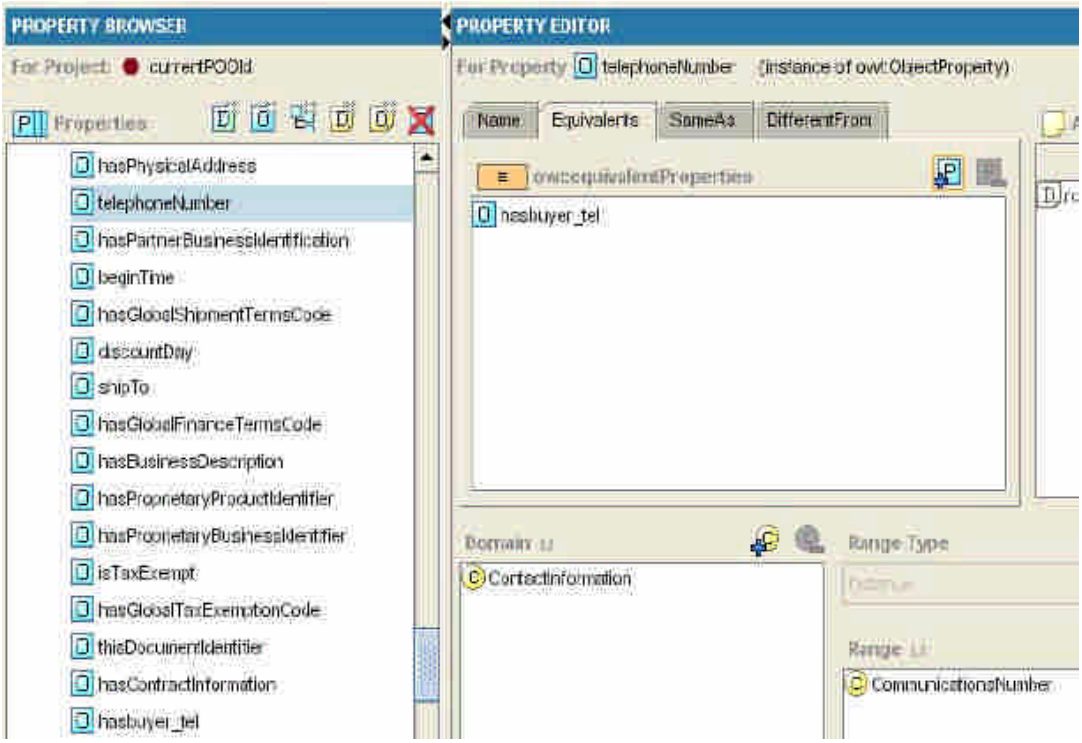
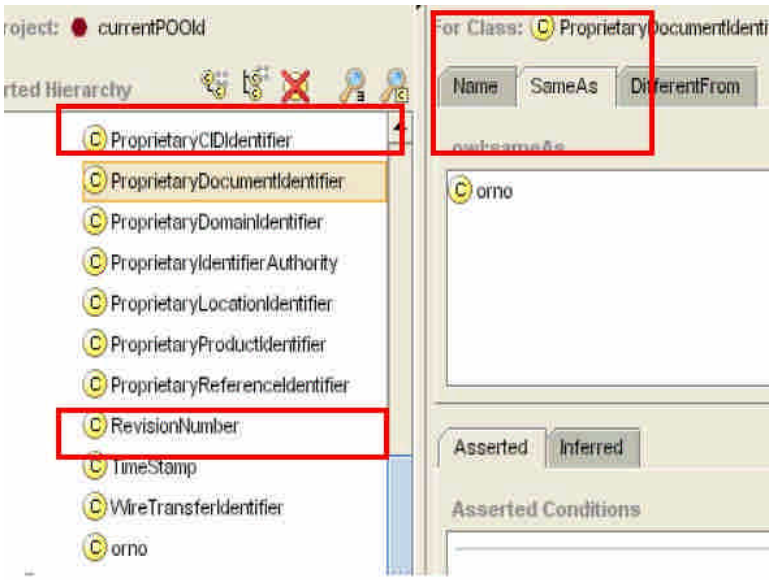


Figure 15: Equivalent Properties


PROPERTY BROWSER
For Project: currentPOOId
Properties
hasPhysicalAddress
telephoneNumber
hasPartnerBusinessIdentification
beginTime
hasGlobalShipmentTermsCode
discountDay
shipTo
hasGlobalFinanceTermsCode
hasBusinessDescription
hasProprietaryProductIdentifier
hasProprietaryBusinessIdentifier
isTaxExempt
hasGlobalTaxExemptionCode
thisDocumentIdentifier
hasContractInformation
hasbuyer_tel
PROPERTY EDITOR
For Property: telephoneNumber (instance of owl:ObjectProperty)
Name
Equivalents
SameAs
DifferentFrom
owl:equivalentProperties
hasbuyer_tel
Domain
ContactInformation
Range Type
Range
CommunicationsNumber



re 15: Equivalent Properties

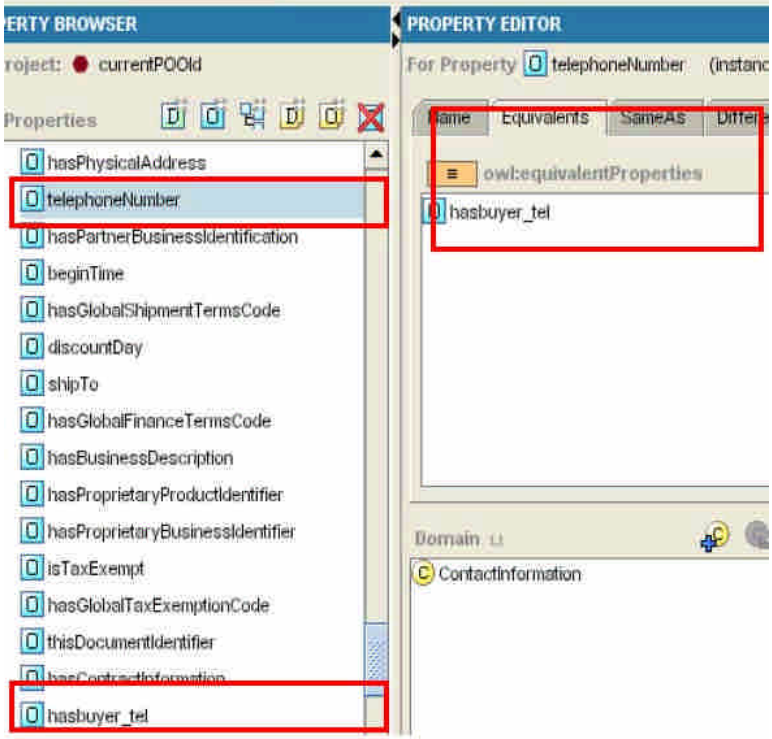


Table 1: Ontology Merge Rules (This Research)

Level	Type	Current (Old)	Standard (New)	Conflict Description	Merge Rules

Class Level	Schematic conflicts	None	New	Standard has a new class, which doest not exist in current process.	We keep the new class in the ontology. All the properties of the new class should be retained, too.
		Existed	None	The current process exist an old class, which does not appear in standard process.	If the old class will no longer exist in the future, we discard them; else we should add the old class to the new ontology.
	Semantic conflicts	Existed Class	New Class	They are with the different class names but the same meaning	We reserve the old class A and add it to new ontology. Then, we use the *owl:sameAs* to state the two classes are equivalent. However, we use the class B usually.
		Existed Class	New Class	They are with the same class name but different meanings.	We keep the name of the new class. However we change the name of old class to another new name.
Property Level	Schematic conflicts	None	New	There are additional properties in a class.	We use and adopt these properties in the new ontology.

		Existed	None	There are deletion properties in a class.	We have to determine whether the properties are no longer useful. If we do not use these properties any more, we discard them. We adjust the minimum cardinality of these old properties to 0 becausethey are not necessary properties in the new class.
	Semantic conflicts	Existed Property	New Property	They are with the different property names but the same meaning	We reserve the old property A and add it to new ontology. Then, we use the *owl:equivalentProperty* to state the two properties are equivalent.
		Existed Property	New Property	They are with the same property name but different meanings.	We keep the name of the new property. However we change the name of old property to another new name.

赴國外研究心得報告

計畫編號	NSC 95－2416－H－004－006 -
計畫名稱	本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(2/2)
出國人員姓名 服務機關及職稱	國立政治大學會計學系所 謝家蘭教授
出國時間地點	2006.08.01-2007.03.31
國外研究機構	UC Berkeley

工作記要：

1. RESEARCH BACKGROUND

Internet has changed the way business conducted between companies worldwide. Firms are now used to exchange business information electronically over Internet. Since the mid-1990s, wave after wave of web technology standards emerge to support the electronic business information exchange. Standards like Extensible Markup Language (XML), Internet Electronic Data Exchange (I-EDI), RosettaNet[1], ebXML[2], Web Ontology Language (OWL), and Semantic Web (SW) surge and sweep electronic commerce worldwide (W3C 2006) (RosettaNet 2006) (ebXML 2006) (OWL 2004). These standards impact on contemporary corporations in many aspects. These standards are proposed to provide a uniform way of business information exchange mechanisms. Semantic not syntactic integration emerges to be the issue that hinders the plan and progress of business-to-business integration electronic commerce (B2Bi EC), which in turn causes time, cost, and reinvention every time there is a change in the public process, there is a change in the standard, and there is a change in the partnership.

The traditional method to tackle the issue can be divided into the programming (ad hoc) approach and the mapping table (syntactic) method. The programming approach solves the problem in a one to one fashion but the result easily becomes the unmanageable "spaghetti" chaos. The mapping table seems to be an easy and convenient approach. However, it only deals with the specific data values not the data definition. An exponentially growing number of trading partners emerge in B2Bi EC. Programming is no longer an effective and flexible way. Mapping table is too primitive and inadequate. The new complexity of data semantic in the business information exchange makes both approaches even harder to tackle the problem (Stojanovic et al, 2002) (Trastour et al, 2003). We believe that Internet growth makes B2Bi climb to a higher level of exchange, that is, the exchange of business meanings and business constraints. A knowledge-intensive and system-to-system semantic integration model and method is in need.

[1] RosettaNet is a consortium of major computer and consumer electronics, electronic components, semiconductor manufacturing, telecommunications and logistics companies working to create and implement industry-wide, electronic commerce and business process standards. RosettaNet is a subsidiary of GS1 US, formerly the Uniform Code Council, Inc. (UCC).

[2] ebXML is a worldwide project initiated and driven by the Organization for the Advancement of Structured Information Standards (OASIS) and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). ebXML is to map out a common framework to enable interoperable electronic commerce and business expressed in XML.

2. RESEARCH ISSUE

Business-to-business integration is to exchange business information between different firms and interoperate the public processes over Internet. The traditional ways of trading include telephone, fax, and email. These approaches introduce faults, redundancies, and wastes. Electronic data interchange is a 1990s and transaction-based approach. However, the change of EDI specification is neither on line or real time. EDI lacks the ability to quickly respond to business changes and suffers from the scalability in the presence of an exponentially growing number of users. Internet EDI is the next stage of B2Bi development. And new B2Bi standards have been proposed based on XML. They indeed provide a more on line and real time method than traditional EDI. However, companies still struggle with the difficulty of heterogeneity and interoperability in the exchange and execution of processes and protocols. In essence, an enhanced approach needs to provide the technology compatibility and the knowledge representation.

Electronic commerce within and across national boundaries is universal. Most firms if not all have problems in one way or another with business process integration and business model interoperability. On both methodological and pragmatic levels, due to increasing diversity in web pages, web services, data sources, and programming languages in all countries, developing an analysis framework of cross national B2Bi resolution is important at international, national, and intra-national levels. This study will develop an analysis framework and a method to explore the way integration and interoperability over schema and semantics can be achieved in B2Bi EC. The dynamics of Internet and intelligence of XML and ontology interplay with inter-organizational context, making it a base for exploring the model and method.

Various approaches have been proposed to study B2Bi issues. However, they lack the process perspective and the semantic representation. Their interoperability is based on adhocracy. Much is needed in the systematic and methodological enhancement. This research intends to tackle the inadequacy of B2Bi standard implementation in forms. An ontology-assisted analysis framework is created to reconcile and represent the conflicts and correspondences in the B2Bi EC issue. Based on the literature, in general, B2Bi framework has three fundamental layers to deal with (Cut et al, 2002) (Falkovych et al, 2003) (Gasevic et al, 2004). They are the communication layer, the content layer, and the process layer. These layers represent the important mechanism and management in B2Bi such as the coupling among partners, the autonomy, and the security. In essence, they mean the specifications of the message formats, the transport protocol, the procedure, and the security mechanism.

3. RESEARCH METHOD

3.1 Analysis-driven Ontology Modeling

The research structure depicted in Figure 1 is the analysis framework we present in the paper to illustrate the model and the method to be developed and deployed in the B2Bi EC standards implementation. The framework is made up of the Unified Modeling Language, the Extensible Markup Language, and the Ontology technologies. Business process interoperability and business data integration are considered the antecedents to B2Bi strategies. More in the framework, a set of analysis procedures are proposed. We analyze the cross national business partners' data schema and process model. We examine the electronic commerce standard in the aspect of data semantics and process semantics. A set of heuristics and rules will be created to represent the above analyzed process models and data schema in form of syntax and semantics. The partners' and the standards' ontologies will be separately developed using the rules and the heuristics. We will merge these ontologies in order to reconcile their conflicts and correspondences. The resulting merged ontologies are tested by the prototype system.

In the end, we hope there is an evolution step to be undertaken to reuse the resulting ontologies. The trading partners can share the domain knowledge in the future standard implementation. The following subsections describe the procedures of the analysis framework and are divided into Step A through Step D. Step A develops the domain ontology of the firm and of the trading partners. Step B creates the domain ontology of the standards. Step C focuses on the ontology knowledge representation for the firm and for the trading partners. Step D creates the ontology knowledge representation of the standards.

[Insert Figure 1 here]

3.2 Step A – Firm Public Process Ontology, "as-is"

A. to analyze the current business process, "as-is"

If we want to analyze the current process, in general, we initiate a meeting. The meeting participants include the process owners and the process users. Through interviewing users, we discover detailed information about the current processes. The detail information contains the process goal, the process flow, the process user role, the process input, the process output and others. This information should be minuted. According to the meeting minutes, we draw the UML diagrams. If we understand the current processes more, we can represent the process as in UML without losing its semantics.

A.1 to design the use case diagram

Before we draw a use case diagram, we have to gather data. We analyze the process actors, the process preconditions, and the process flow to fill out an analysis form. Take the purchase order (PO) as an example. There should be two actors in the purchase order process: buyer and seller. Before the buyer orders something, the seller makes a request for a quote document from the seller first. Then, if the buyer accepts the quote, he sends a purchase order to the seller. When the seller receives the purchase order, the seller confirms the order. This scenario is the common and simple one.

A.2 to design the sequence diagram

In a sequence diagram, we try to discover all messages that are exchanged in a business process and in the purchase order. It can be extracted from the use case diagram and the meeting minutes. In the purchase order example, the PO Request is the first message to be sent from the buyer to the seller. When the seller receives the order request, the seller should check the inventory to determine whether the firm can fulfill that purchase order or not. Then the PO Confirmation is the next message to be sent from the seller to the buyer.

A.3 to design the activity diagram
An activity diagram can show the flow from one activity to another activity. It can represent the detailed process flow. We should find the information from discussion at the meetings so as to develop the activity diagram. We need to discover the detailed actions in the flow, the initial state, and the final state. We then continue the PO example and finish the activity diagram. In this example, we have three actions: request a purchase order, check inventory for this order, and confirm this purchase order.

A.4 to design the class diagram
We try to extract a generic class construct from the use case diagram, the sequence diagram, and the activity diagram. Again, we move on with the PO example. First, we work on the use case diagram. We discover four components: the two actors and the two use cases. We take the two major elements in the use case diagram, Actor and Use Case, to form the two classes: Actor and Activity. Next, we extract the class Message from the sequence diagram, because the sequence diagram describes the message flow and the order flow between the objects. Then, we work on the activity diagram which consists of several actions as described above. The class Action can be extracted.

3.3 Step B – Standard Public Process Ontology, "to-be"

B. to develop the EC-standard-compliant business process
We use four UML diagrams to perform the work such as the use case diagram, the sequence diagram, the activity diagram, and the class diagram. They are utilized to model an EC-standard-compliant business process. The mapping methods between the four diagrams are the same as in Step A. The difference between Step A and Step B is the source of analysis. Step A focuses on the firm existing and current public processes. We have to collect and examine them through interviews and observations. We model the standard processes from B2Bi EC standard specifications at Step B. Some B2B standards have the concept of process, but some do not. If they do not, we should discuss this issue with the trading partners in order to develop a new standard process specification based on the B2Bi EC recommendation. Of course, some B2B standards have adopted UML diagrams to present their standard processes in the specification. We can directly use them.

B.1 to design the use case diagram
We develop the use case diagram based on the B2B standard specification. A B2B standard specification often describes the process purpose and the process definition in the statements. We search and extract the basic components for a use case from the process statements.

B.2 to design the sequence diagram
The B2B standards should specify the sequence of the exchanged messages. The latest standards often adopt the sequence diagram to represent the sequence. Therefore, we use the diagram provided by the standards. If the standards do not use UML diagrams, we still can analyze the sequence of messages in the generic control constructs.

B.3 to design the activity diagram
A B2B standard should formalize the public process flow. Such formalization allows the partners to follow. We do not expect to manage many different process flows with our trading partners in the real world. A B2B standard provides the well-defined process flows. We can extract and formalize the defined process flow from B2B standard specification.

3.4 Step C – Firm Ontology Representation

C.1 to capture the current B2B ontologies
In this study, we propose a heuristics approach to model the ontologies for the firms and the B2Bi EC process and message. We build the ontology so as to describe the firms' B2B domain knowledge. This domain ontology contains the basic classes and properties. Every business process should fit in an ontology definition. We define the basic B2B components and properties.

C.2 to model the current business document ontology
We analyze the core of the public processes performed between B2B partners. The core means the message analysis. We then need to develop a process ontology based on the semantics of the message analysis. The semantics refer to the context, the meaning, the terminology, and the relationship in the business document exchange process.

C.3 to reconcile the current business constraints
We may have constraints on each entity, each message, and each process. These constraints have to be converted into OWL. After business process and document ontology being created, we move on to build the EC standard ontology.

3.5 Step D – Standard Ontology Representation

D. to capture the EC standard's ontologies
To build the EC standard ontology, we need to find out the B2B process specifications and their business documents. The definition of each business document is often encoded as DTD or XML Schema. We use the schema to create these EC standard ontology.

D.1 to design the EC standard's process ontology
Notice that not all EC standards require implementing all elements in the specifications. Only the standards that are required in the partnership will be converted into OWL classes. In this section, we develop a set of heuristics to address the issue.

D.2 to model the EC standard's document ontology per partner
The way to model the standard document ontology is the same as above. We extract the

data definition in standard to do the conversion.

D.3 to reconcile the EC standard's constraints
The standard may have constraints on each entity, each message, and each process. The trading partners in between may have their own practice constraints. We extract to collect them and use the above procedures to convert them into OWL object properties.

3.6 Step E - Ontology Merge

When initiating and implementing a new B2B initiative, we deal with new B2B EC standards. Different business partners and different settings occur. Though we have the existing ontology in the ontology repository, these new differences cause the ontology mismatch and inconsistency. We need to resolve and merge these ontologies including functions of (a) reading in ontologies, ontology updates, and adaptations, (b) viewing a specific version or a variant of an ontology, (c) differentiating ontologies, (c) checking the inconsistency in the ontology combination.

In essence, the key to merge is to discover the differences and to generate the correspondence rules between ontologies. The differences are like the instances of the changes of class name, the addition or deletion of classes, the addition or deletion of properties, and the mergence or split of classes. Though it is common to find new conflicts and differences between new trading partners, there are common parts as well to take advantage of as we discuss on the repeat rate and reuse. The hard part is the more heterogeneous the ontologies are, the larger extent of change to be implemented between the old and the new processes. Analysis gives us the parts of the process to be changed and installed in the coming implementation. We adopt the ontology of the B2B standard and merge the B2B standard ontology based on the merge rules as listed in Table 1.

[Insert Table 1 here]

3.7 Step F – Ontology Representation

We have described the ontology representation in Step C and Step D. The technique is used in the merged ontology.

3.8 Step G – Ontology Test

To verify the ontologies merged, we consider two issues, the syntactic test and the semantic test. We test the syntactic of ontology through the ontology tool. It automatically validates the inconsistency of syntax. The semantic test is to discover the inconsistency between the database schema, the business processes, and the old version of ontology. We extract the database schema and examine the consistency between the business ontology. We compare the consistency between the trading agreements in order to specify the business rules. We analyze the differences between the new ontology and the real environment. The analysis results will be used to adjust the business process to refine the merged ontology.

Figure 2 illustrate the Steps. As described above, we first discover the ontology requirements in Step A and Step B. We then create the ontology from Step C and Step C.

We merge ontologies in Step E. Step F gives a merged representation. Step G tests the syntax and semantics.

[Insert Figure 2 here]

4. AN EXPERIMENTAL STUDY

4.1 A Prototype System

In this research, we have developed an experimental prototype that implements the presented B2Bi ontology development method. This prototype is built to facilitate the illustration of the feasibility and the validity of the method. In this section, we demonstrate an application of the prototype in two main electronic commerce standards, the RosettaNet, a worldwide and vertical B2B standard; and the ebXML, an OASIS and UN sponsored and horizontal B2B standard (RosettaNet 2006) (ebXML 2006) (OASIS 2005) (Hofreiter et al, 2002). Both standards are installed worldwide because they cover the diverse electronic commerce practices. We use these two major standards as the starter experiments in the illustration that our new method is feasible and valid. Preliminary experimental results show that this ontology-assisted method gives a viable resolution to the long-standing semantic and syntactic issue in the implementation of electronic commerce standards.

In both experiments, we choose the purchase order process as the baseline to illustrate a live case study of a large scale semiconductor component distributor. The live case company is called company W. Company W is the number one distributor in the Asia Pacific region since 2004. The purchase order process is the main business process in their B2B EC. Company W since 2004 became quite concerned with the various EC standards to be installed and among its cross-national suppliers and customers. The time and efforts grow exponentially. At the same time, Company W is troubled by a needed lift to the next level of performance of global supply chain management. And B2Bi is the bottleneck of the performance and becomes the compelling reason to reengineer the electronic commerce architecture.

An ontology-assisted B2Bi eCommerce prototype architecture as shown in Figure 3 is developed in the experimental study. The B2Bi platform allows enterprises to exchange business documents over Internet. It provides various and common B2B protocols to connect the trading partners. It provides the ability to streamline the business process and the adapters when linking with the various enterprise information systems.

[Insert Figure 3 here]

We build the research model of Step A through Step G into a prototype system. The layers in the system are illustrated in Figure 4. The system provides a number of main functions such as the DTD Importer, the Ontology Editor, and the Ontology Display. Figure 5 illustrates the structure of the functions.

[Insert Figure 4 here]

[Insert Figure 5 here]

4.1.1 DTD Importer

DTD importer parses the DTD that specifies the document format and transfers DTD to ontology. The user can enter the output OWL file as shown in Figure 6. This feature will transfer the file automatically. We will produce two groups of class and one group object property. The classes are B2B_DataEntity and B2B_ComposedDataEntity. The object property is B2B_BusinessProperty. The DTD Importer will differentiate all entities from DTD file base on the nature.

[Insert Figure 6 here]

The DTD Importer also provides an ability to parse the entity's metadata. Through parsing the metadata, we can enrich our document ontology. This program will read the entity information using a batch approach as shown in Figure 7.

[Insert Figure 7 here]

4.1.2 Ontology Editor

We build a process ontology template. The basic classes and properties of the B2B process can use this template to develop the ontology as shown in Figure 8 and Figure 9. Business constraints also can be edited through the ontology.

[Insert Figure 8 here]

[Insert Figure 9 here]

4.2 A RosettaNet Experiment

RosettaNet consortium (RosettaNet Consortium, 2004) is a non-profit consortium of more than 500 organizations working to create, implement and promote open eBusiness standards and services. RosettaNet tries to establish a common language and a standard processes for the electronic sharing of business information. In order to implement the experimental scenario, we install a set of RosettaNet core specifications. We will explain each in the following sections. These core specifications include RNIF, PIP, and Dictionary.

We chose the RosettaNet Partner Interface Process™ (PIP) 3A4, the purchase order request process, to be the experimental public process, which is mostly implemented and installed. Purchase order process is corresponding to Company W's sales order flow. The PIP3A4 specification provides the details of the purchase order process property and constraint. The structure and content of the business documents to be exchanged in RosettaNet is specified in DTD and XML Schema. We need to build the process ontology from the PIP specification and to create the document ontology from the DTD and message guidelines.

Step A - to analyze the existing business process

We analyze the current business process of purchase order between Company W and its buyers. In the use case diagram, we see two kinds of participants: Company W and buyers. Company W has the partner role of “Seller” and customers play the role of “Buyer”. In the sequence diagram, we see two business documents that are exchanged. They are the “Customer Order” and “Customer Order Ack”. There are three main activities in the sales order flow such as “Send A Customer Order”, “Check Inventory” and “Confirm Customer Order”.

Step B - to develop the B2B EC-standard-compliant business process

The RosettaNet specifies PIP3A4 exchanging three messages such as “PurchaseOrderRequest”, “PurchaseOrderConfirmation” and “ReceiptAcknowledgment”. PIP3A4 further specifies two more activities, that is, “Request Purchase Order” and “Confirm Purchase Order”. They represent the standards to be complied with.

Step C - to represent the existing firm ontology

C.1 to design current business process ontology

Company W and its buyers are B2B_Partner. We add them as the instances of B2B_Partner. The instance’s naming rule should be pre-established. We create the domain ontology of each business process for Company W and its buyers. In Figure 10, we show the B2B_Process “Customer Order”.

[Insert Figure 10 here]

C.2 to model current business document ontology

We use the DTD importer in the prototype to convert and store the DTD files into document ontology repository. The converted data result is shown in Figure 11.

[Insert Figure 11 here]

Step D - to represent B2B EC standard ontology

D.1 to design RosettaNet process ontology

The PIP3A4’s ontology is built from the standard specification. The PIP restrictions must be considered. For example, the PIP3A4 specification defines when to enable a buyer to issue a purchase order and when to enable a seller to acknowledge the receipt of order, and even down to the line item level. No matter how and if the order is accepted, rejected, or pending. The provider's acknowledgment must include the information about delivery expectation. Further, when a provider acknowledges that a product line item on the purchase order document is “pending”, the provider must later use PIP3A7, "Notify of Purchase Order Acknowledgment" to notify the buyer when the product line item is either finally accepted or rejected. The PIP specification also describes the process start state to be one of the following: Purchase Order Request Exists, TPA Exists, Requesting Partner Approved, Responding Partner Approved, Buyer Authorized, Purchase Order Request Valid, or Purchase Order Request Non-Repudiated as shown in Figure 12. One of the above must be returned as the initial date started.

[Insert Figure 12 here]

D.2 to model RosettaNet document ontology
The PIP3A4's DTD files and message guideline give the details of each entity in the standard document. We again use the DTD importer to convert and load the specification. We edit the constraints for each entity. The DTD importer can intelligently determine and set up the business property domain and domain range properly. A RosettaNet PIP definition can be added as the instance's comment as shown in Figure 13.

[Insert Figure 13 here]

Step E - to merge ontology
In the merge of ontologies, the purchase order has a unique number as the identity of the order. In this experiment, the current purchase order number is named "orno". However PIP3A4 uses the field "ProprietaryDocumentIdentitfer" as the purchase order number. We resolve the inconsistency via link analysis.

When we generate the current business ontology and the B2B standard ontology, we can merge them in the system. Although Protégé provides the merge, we need to adjust the detail correspondence rules to link the relationship between two ontologies. The ontology editor provides the test function helps us check the ontology consistency. The purchase order usually has a unique number as the identity of the purchase order. The current order number is named "orno". However, PIP3A4 names the field as "ProprietaryDocumentIdentitfer". We use the *owl:sameAs* to link these two classes as equivalent.

Step F - to represent ontology
We generate a set of HTML files that contain the content of ontology. Users browse the ontology in a user-friendly interface. It minimizes the risk of ontology to be altered. With the hyperlink, we can trace any related classes and properties.

Step G - to test ontology
We validate the ontology with the Protégé test ontology function.

4.3 An ebXML Experiment
The second experiment we have conducted is a realization of ebXML in the case of purchase order. ebXML (ebXML, 2004) was started in 1999 and developed by the Organization for the Advancement of Structured Information Standards (OASIS). OASIS is a non-profit, international consortium that drives the development, convergence, and adoption of eBusiness standards. The consortium produces more Web services standards than any other organization along with standards for security, eBusiness, and standardization efforts in the public sector and for application-specific markets. Founded in 1993, OASIS has more than 3,500 participants representing over 600 organizations and individual members in 100 countries (OASIS Consortium, 2004). ebXML is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. By using ebXML, companies can exchange business messages, conduct trading relationships, communicate data in

common terms and define and register business processes. In order to implement the experimental scenario, we install a set of ebXML core specifications. We will explain each in the following sections. These core specifications include the business process, the core component, and the collaboration protocol profile and agreement.

Steps A, B, C from the method are similarly applied in the ebXML experiment. The main difference is in Step D where we explain the procedure of transition. The difference is when we try to model the ebXML standard specification in ontology.

D.1 to design ebXML process ontology

ebXML uses Business Process Specification Schema (BPSS) to model business processes. In modeling the process ontology, we utilize the sequence diagram and the activity diagram. The BPSS specification specifies a Business Transaction, a Business Document flow for the Business Transaction, a Binary Collaboration, and then a Choreography for the Binary Collaboration. A Business Transaction in ebXML is the basic transaction unit between two partners. It consists of a Requesting Business Activity and a Responding Business Activity. A Binary Collaboration is always executed between two roles. They are called the Authorized Roles because they represent the actors that are authorized to participate in the collaboration. A Binary Collaboration consists of one or more Business Activities. These Business Activities must be conducted between the two Authorized Roles in the Binary Collaboration. A Choreography is an ordering and sequencing of Business Activities within a Binary Collaboration. The choreography is specified in terms of the Business States and the Transitions between these Business States. In the purchase order process example, we know it has two activities: Purchase Order Request Action and Purchase Order Confirmation Action. Both can be extracted from the BPSS sample file. We further know that if the activity is authorization required or non repudiation required. Below is the converted OWL scripts.

D.2 to design ebXML document ontology

The ebXML document ontology is created in use of the ebXML core component specification. The core component can be used to create the classes and the properties and be converted into ontology. The ebXML document ontology needs to incorporate existing DTD or existing XML Schema in order to support each component conversion. The ebXML core component in the standard ontology corresponds to the basic data entity in the domain ontology. The core component represents the same meaning as the data entity in the domain ontology. The ebXML aggregate information entity, on the other hand, stands for the composite data entity in the domain ontology. Because BPSS is also UML-based, the set of UML diagram such as the use case diagram, the class diagram, the sequence diagram, and the activity diagram our method recommends are adopted. Hence, the correspondence and reconciliation will occur early at the analysis phase and will be carried out in the merge of standard ontology and domain ontology. For ebXML, the actual document definition is achieved using the ebXML core component specifications or by some methodology external to ebXML. They in turn are converted into a DTD or a XML Schema. BPSS specifies the specification name as “PurchaseOrderReques.dtd”.

We only need to get this DTD file and import it. A business document has three types of

components. They are the basic data entity, the composite data entity, and the business property. The ebXML consists of: Core Component Type (CCT), Basic Information Entity, and Aggregate Information Entity. CCT are core components that carry the actual value and have no business meaning on their own. A Basic Information Entity is a singular concept that has a unique business semantic definition. A Basic Information Entity adds a semantic meaning to a single datatype or a CCT. When CCTs are reused in a business context, they become Basic Information Entities. An Aggregate Information Entity contains two or more Basic Information Entities or Aggregate Information Entities that together form a single business concept. Each Aggregate Information Entity has its own business semantic definition.

5. DISCUSSION

5.1 Research Implication

The first set of literature investigating the issues of aligning the business processes with the business-to-business electronic commerce standards are described in (Omelayenko, 2001) (Stojanovic, Maedche, Motik, and Stojanovic2002) (Bussler, Fensel, and Maedche, 2002). They represent the earlier efforts working on the programming to approach the interface between the trading processes and new standards. This ad hoc programming approach was primitive and exploratory. Later in the literature survey, (Ding, Fensel, Klein, Omelayenko, and Schulten, 2004), the time when there were more business-to-business middleware systems in the marketplace, the researches gear toward to solve the issue of conversion and hub in the middleware system. Another important stream of literature presented in (Choy and Kim, 2004) (Cao, Zhang, and Seydel, 2005) addresses the integration aspect of the alignment. These studies relate to ours. They perceive the issue as a process and performance issue in the supply chain management. In (Hsieh, Lai, and Shi, 2006) (Iyer, Gupta, Johri, 2005) the process issue is further examined with mathematics to model the business operation.

In 2004 and 2005, we tested new PIPs including PIP3B18, PIP3B2, and PIP4C1 in the live case experiment. Each PIP took one month to three months instead of six months as in the prior years. The IT division assigned four full time system engineers instead of six to deliver the implementations. Figure 14 shows the equivalent classes generated from the PIP3A4 standard. Figure 15 gives the equivalent properties in the test.

[Insert Figure 14 here]

[Insert Figure 15 here]

5.2 Managerial and Technical Implications

We summarize the managerial and technical implications into five points.

Ontology is a more powerful technology on semantics and context. A mapping table is simple but lacks the ability to scale. It is a way of “mapping of terms” not an approach to “mapping of sense”. Ontology allows systems to discern the “one to one”, the “one too many”, and the “many to many” correspondences. It allows the systems to undertake the complex conversion such as the situation when there are same terms but with different meanings; different terms but with same meaning; different terms

but with different meanings yet close. Ontology can support an automation of the evolution of the terms, the concepts, and the relationships. The relationships between the new terms and the corresponding old terms can update automatically. Ontology is the base of reasoning.

The analysis framework and ontology enables the deployment of a new B2Bi EC standard initiative to be installed and operated in an effective and efficient fashion. Though RosettaNet PIP message defines many business entities, but there are many repeating entities in different PIPs. RosettaNet PIP3B2 is an example of the shipment notification process which specifies 120 business entities and properties. PIP3A4 is another example that has 143 business entities. However, there are 59 repeating business entities between these two PIPs. The repeat rate is above 49%. In fact, the more related the processes are, the higher the repeat rate will be. We must take advantage of the repeat rates. The repeat rate of entities between PIP3A4 and PIP3A8 is as high as 92%, almost identical. As new processes are continuously and constantly added to our ontology, the ontology must become more robust. The work of ontology creation becomes more automatic and less labor intensive. At the same time, the new knowledge extracted from the new ontology can be captured. Trading partners regularly collaborate and contribute to the reconciled and merged ontology which in turn forms a semantically rich repository in support of the reasoning, the inference, and the search of organizational learning. By enhancing and enriching the shared ontologies, we can deploy a new B2Bi EC initiative in a more effective and efficient manner.

A special function to transfer data model from DTD and XML Schema to OWL is useful. A prototype of parser and converter to handle XML documents in the creation of ontology is needed.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this paper, we have presented an ontology-assisted analysis method and alignment model in the implementation of the business-to-business electronic commerce standard specification over the existing trading partners' public processes in the syntactic and semantic integration and interoperability. An application of the Unified Modeling Language is made to analyze the public process in the domain and in the standard. Terms, concepts, relations, and links are created from the analysis results and converted into an ontology representation. Web ontology language is introduced to formulate the analyzed knowledge and experience to align the domain and the standard. There are correspondences and conflicts in the process of alignment. They are resolved via the shared and reusable ontology which is a convergence of the domain ontology and the standard ontology. The converged and shared ontology is achieved via a set of rules and heuristics that are created in the research. The key of success in the business-to-business integration electronic commerce lies on the ability to accomplish the process interoperability and the schema comparability. Three main tasks have to be achieved to fulfill the requirements. In this research, we have constructed a prototype to implement the method. The prototype is used to illustrate the feasibility and validity of the method. A set of starter experiments has been conducted in use of a straight through example of a

purchase order process in the alignment with the RosettaNet standard and the ebXML standard. The starter experiment serves as the baseline to demonstrate the method is feasible and valid. The three main things we have accomplished in the research are:

Identifying the main components of knowledge and experience to be reconciled and to be represented in the alignment of the standard ontology and the domain ontology.

Developing the set of rules and heuristics for the ontology correspondence and reconciliation.

Designing an experimental prototype to implement the method and to demonstrate the feasibility and the validity by selecting two main electronic commerce standards as the baseline test. The RosettaNet experiment represents the vertical electronic commerce standard. The ebXML experiment stands for the horizontal electronic commerce standard.

6.2 Future Research Work

The future research work will continue to explore the complex issues of the alignment and automation between domains and standards. Some of the immediate tasks to be undertaken in our study include:

Enhancing the ontology search and inference capability. As the rule base and heuristics base grow, search and inference engine become slow in the ontology management. Tuning and enhanced rules must be developed.

Upgrading the DTD importer to import XML Schema and to enable the conversion between XML Schema and OWL representation. This will solve the version control issue.

Conducting more diverse and complex experiments in terms of scale and scope. More experiments will be conducted in the public processes of receiving and payment that are closely related with the public process of purchase order. RosettaNet and ebXML will still be the main standards.

REFERENCES

1. Baclawski, K., Kokar, M.K., Kogut, P.A., Hart, L., Smith, J., Letkowski, J., & Emery, P. (2002) “Extending the Unified Modeling Language for Ontology Development”, Software and Systems Modeling, Vol 1 No 2, pp. 142-156.
2. Berners-Lee, T., Hendler, J., & Lassila, O. (2001) “The Semantic Web”, Scientific American, Vol 284 No 5, pp. 34-43.
3. Bird, Linda, Andrew G., & Terry H. (2000) “Object Role Modelling and XML-Schema”, ER2000.
4. Bose, R. (2006) “ Understanding Management Data Systems For Enterprise Performance Management ”, Industrial Management and Data Systems, Vol 106 No 1, pp. 43-59.
5. Bussler, C. (2001) “Semantic B2B integration”, ACM SIGMOD Record, Vol 30 No 2, pp. 625.
6. Bussler, C., Fensel, D., & Maedche, A. (2002) “A Conceptual Architecture for Semantic Web Enabled Web Services”, ACM SIGMOD Record, Vol 31 No 4.
7. Cao, M., Zhang, Q.Y., Seydel, J. (2005) “B2C e-commerce web site quality: an

empirical examination", Industrial Management & Data Systems, Vol 105 No 5, pp. 645-661.
8. Choy, L. Y., Kim, H. T. (2004) "A process and tool for supply network analysis", Industrial Management & Data Systems, Vol 104 No 4, pp. 355-363.
9. Cranefield, S., & Purvis, M. (1999) "UML as an Ontology Modelling Language", In Proceedings of 16th International Joint Conference on Artificial Intelligence on Workshop on Intelligent Information Integration.
10. Cranefield, S. (2001) "Networked Knowledge Representation and Exchange Using UML and RDF", Journal of Digital Information, Vol 1 No 8.
11. Cut, Z., Jones, D., & O'Brien, P. (2002) "Semantic B2B Integration: Issues in Ontology-based Approaches", ACM SIGMOD Record, Vol 31 No 1, pp. 43-48.
12. Decker, Stefan, Sergey M., Frank V. H., Dieter F., Michel K., et al. (2000) "The Semantic Web: The Roles of XML and RDF", IEEE Internet Computing, Vol 4 No 5, pp.63-64.
13. Ding, Y., Fensel, D., Klein, M., Omelayenko, B., & Schulten, E. (2004) "The Role of Ontologies in eCommerce", Handbook on Ontologies, Staab S. and Studer R., Springer.
14. ebXML BPSS (2006) http://www.ebxml.org/
15. Gasevic, D., Djuric, D., Devedzic, V., & Damjanovic, V. (2004) "Converting UML to OWL Ontologies", Proceedings of the 13th international World Wide Web Conference, pp. 488-489.
16. Gulledge, T. (2006) "What is integration?," Industrial Management & Data Systems, Vol 106 No 1, pp. 5-20.
17. Helo, P., Szekely, B. (2005) "Logistics information systems: An analysis of software solutions for supply chain co-ordination", Industrial Management & Data Systems, Vol 105 No 1, pp. 5-18.
18. Hunag, C.J., Amy J.C. Trappy, Yao, Y.H. (2006) "Developing an agent-based workflow management system for collaborative product design", Industrial Management & Data Systems, Vol 106 No 5, pp. 680-699.
19. Hsieh, C.T., Lai, F.J., Shi, W.H. (2006) "Information orientation and its impacts on information asymmetry and e-business adoption: Evidence from China's international trading industry", Industrial Management & Data Systems, Vol 106 No 6, pp. 825-840.
20. Iyer ,L. S., Gupta, B., Johri, N. (2005) "Performance, scalability and reliability issues in web applications", Industrial Management & Data Systems, Vol 105 No 5, pp. 561-576.
21. Kogut P., Cranefield S., Hart L, Dutra M., Baclawski K., Kokar M., & Smith J. (2002) "UML for Ontology Development", The Knowledge Engineering Review, Vol 17 No 1, pp. 61-64.
22. Lesjak ,D., Vehovar, V. (2005) "Factors affecting evaluation of e-business projects", Industrial Management & Data Systems, Vol 105 No 4, pp. 409-428.
23. Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A.H.H, & Elmagarmid, A.K. (2003) "Business-to-Business Interactions: Issues and Enabling Technologies", The VLDB Journal, Vol 12 No 1, pp. 59-85.
24. OASIS Consortium (2005) http://www.oasis-open.org
25. Object Management Group (2005) http://www.omg.org

26. Omelayenko, B. (2001) “Preliminary Ontology Modeling for B2B Content Integration”, Proceedings of the 12th International Workshop on Database and Expert Systems Applications, pp. 7-13.
27. Rahm, Erhard & Philip A. B. (2001) “A survey of approaches to automatic schema matching”, The VLDB Journal, Vol 10, pp. 334-350.
28. RosettaNet Consortium (2006) http://www.rosettanet.org/
29. Stojanovic L., Maedche A., Motik B., & Stojanovic N. (2002) “User-Driven Ontology Evolution Management”, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pp. 285-300.
30. Web Ontology Language (2004) http://www.w3c.org/2004/OWL/
31. Zhao, F. (2004) “Management of information technology and business process re-engineering: a case study”, Industrial Management & Data Systems, Vol 104 No 8, pp.674-680.

Figure 1: An Ontology-assisted B2Bi EC Alignment and Management Framework (This Research)

Reuse, Manage, and Evolution

Cross Trading Partners’ Biz Model and Process

Analyze EC Standards’ Data and Process using UML

Merge Ontologies, “to-be”

Test XML and Ontologies

Model Standard Ontology using UML

Model Public Ontology using UML

Analyze Partners’ Data and Process, “as-is” using UML

Represent XML and Ontologies

EC Standards’ Biz Model and Process

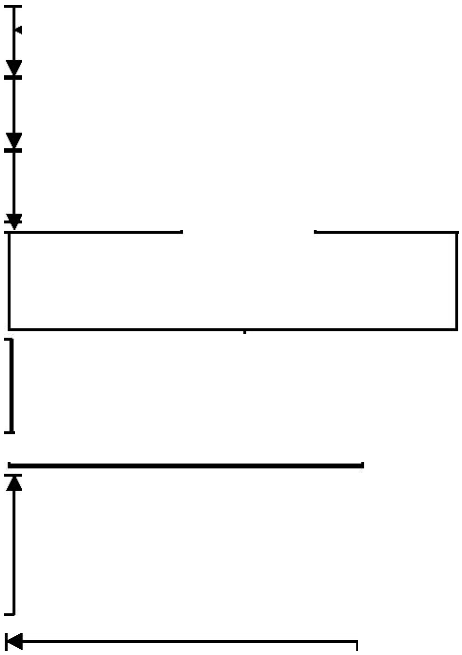


Figure 2: An Ontology-assisted B2Bi eCommerce Alignment Framework (This Research)

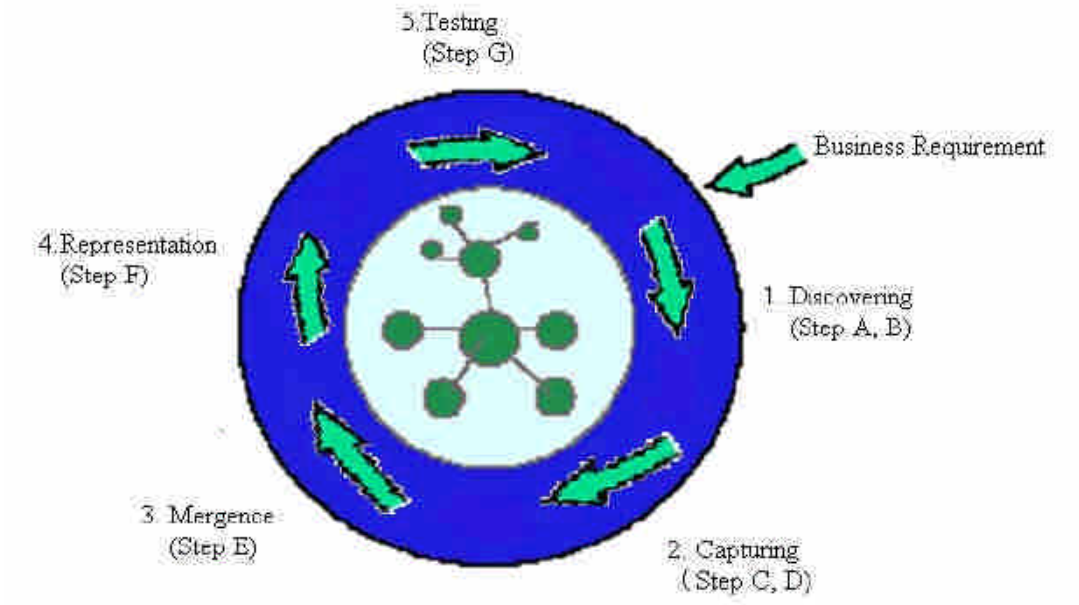


Figure 3: A Prototype Architecture of An Ontology-assisted B2Bi eCommerce Platform (This Research)

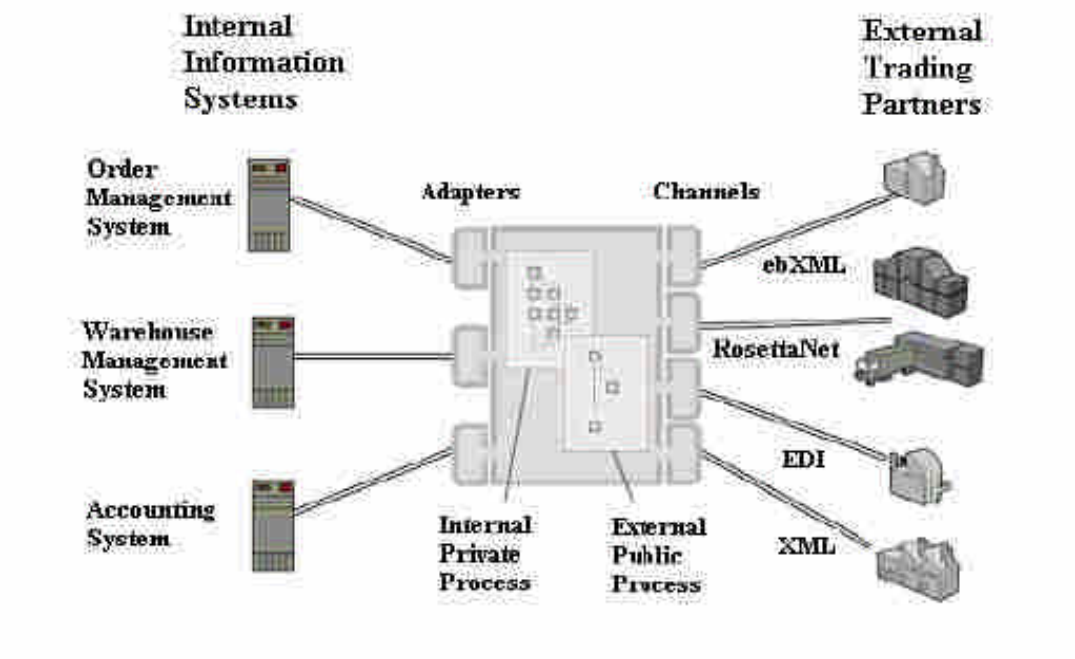


Figure 4: Layers in the Prototype System (This Research)

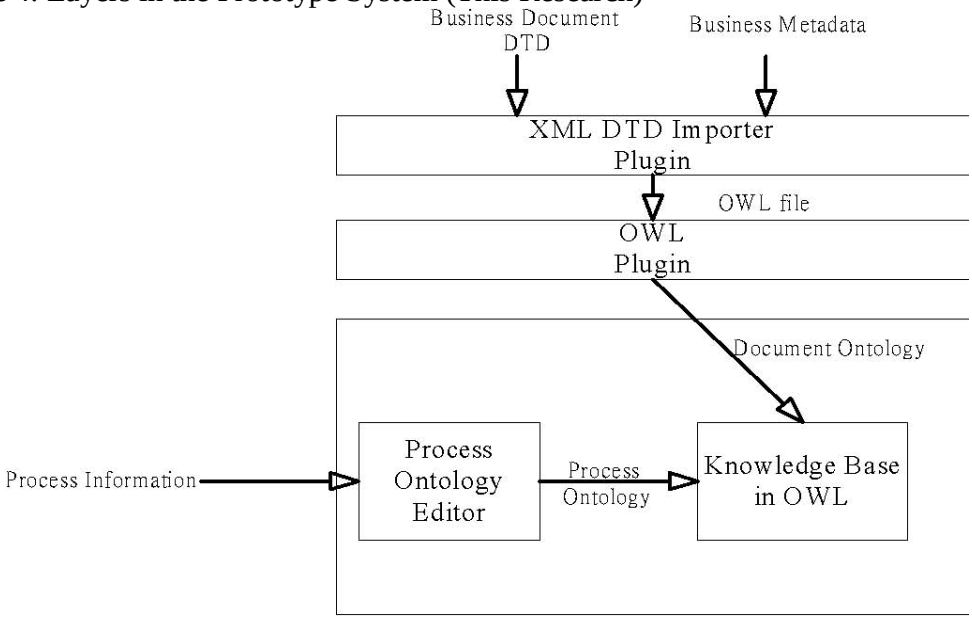


Figure 5: Main Functions of the Prototype (This Research)

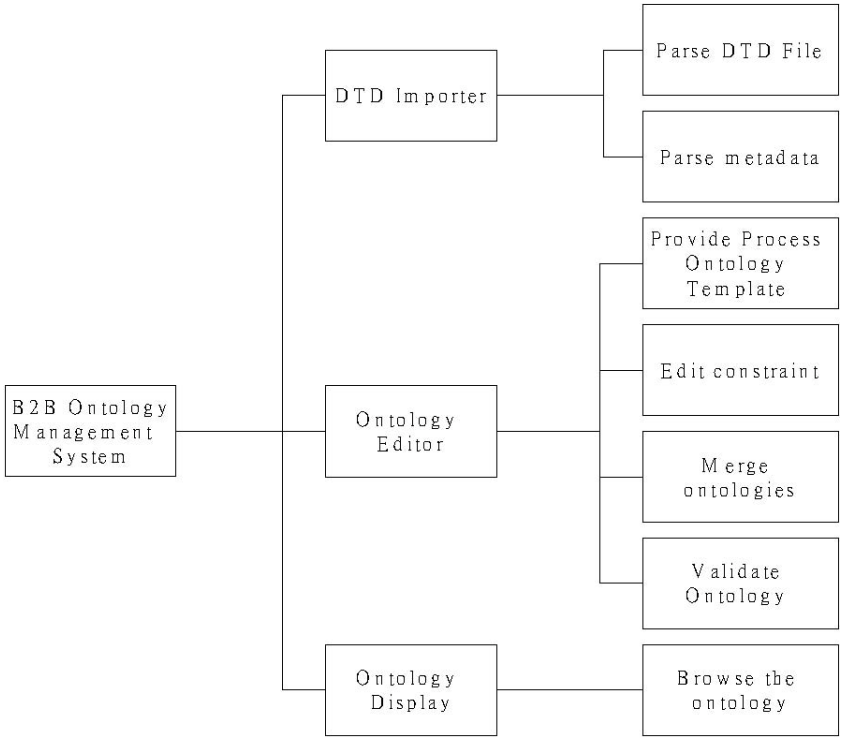


Figure 6: The B2B DTD Plug-in (This Research)

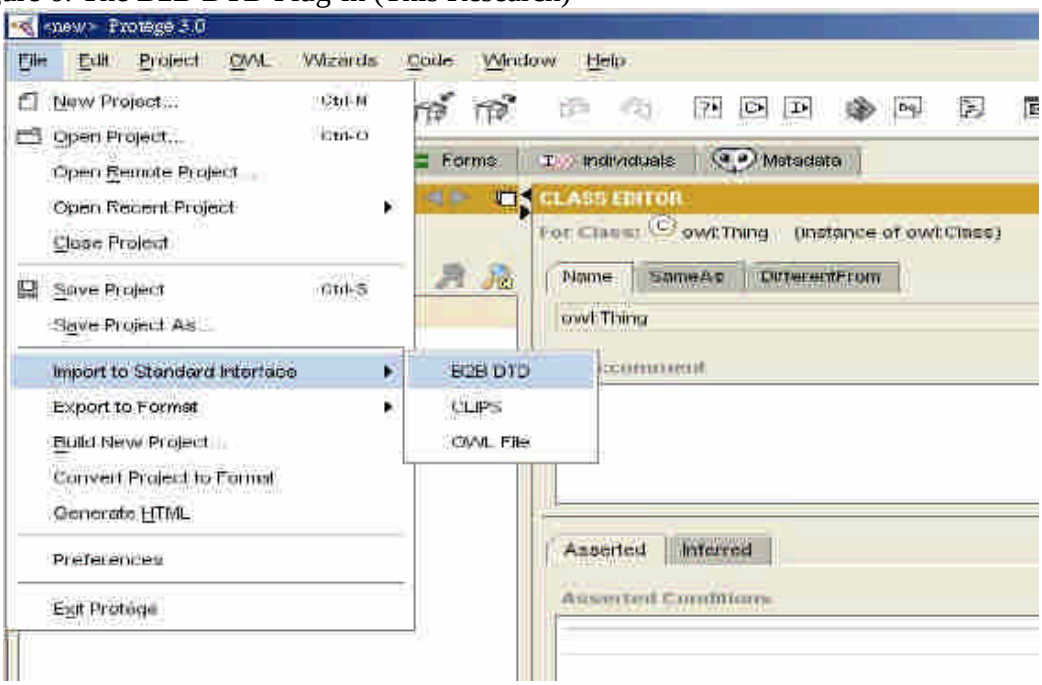


Figure 7: The B2B DTD Importer (This Research)

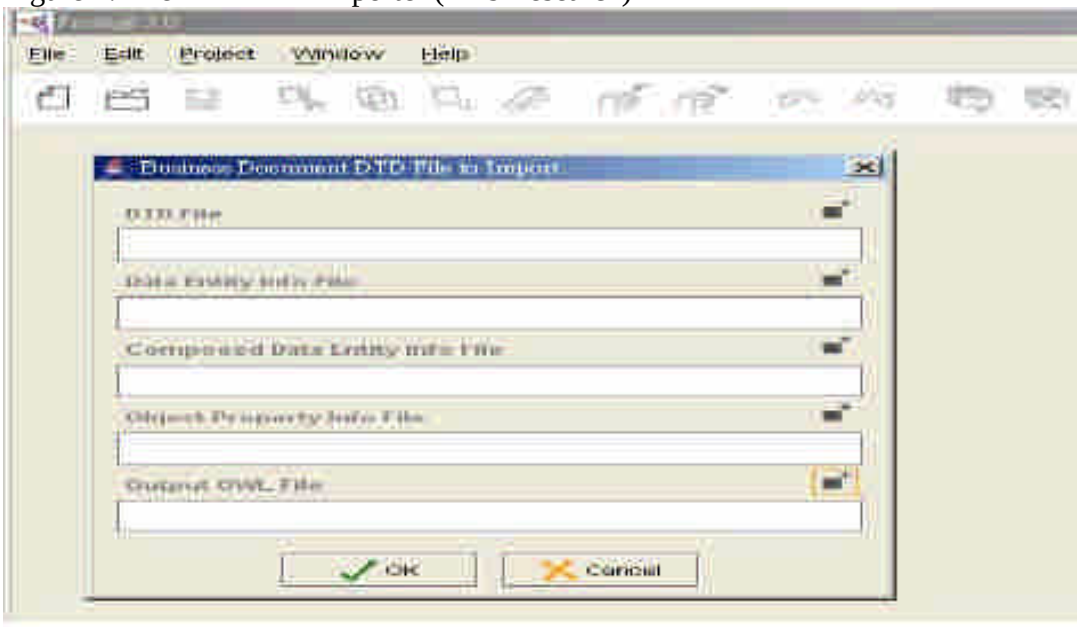


Figure 8: The Basic Classes of a Process Ontology (This Research)

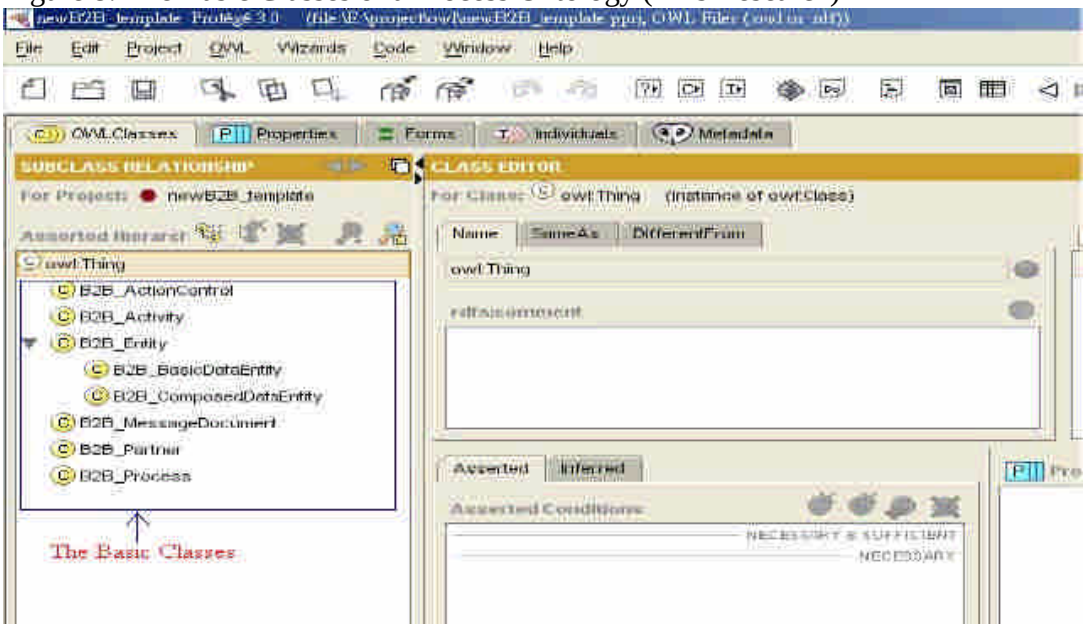


Figure 9: The Basic Properties of a Process Ontology (This Research)

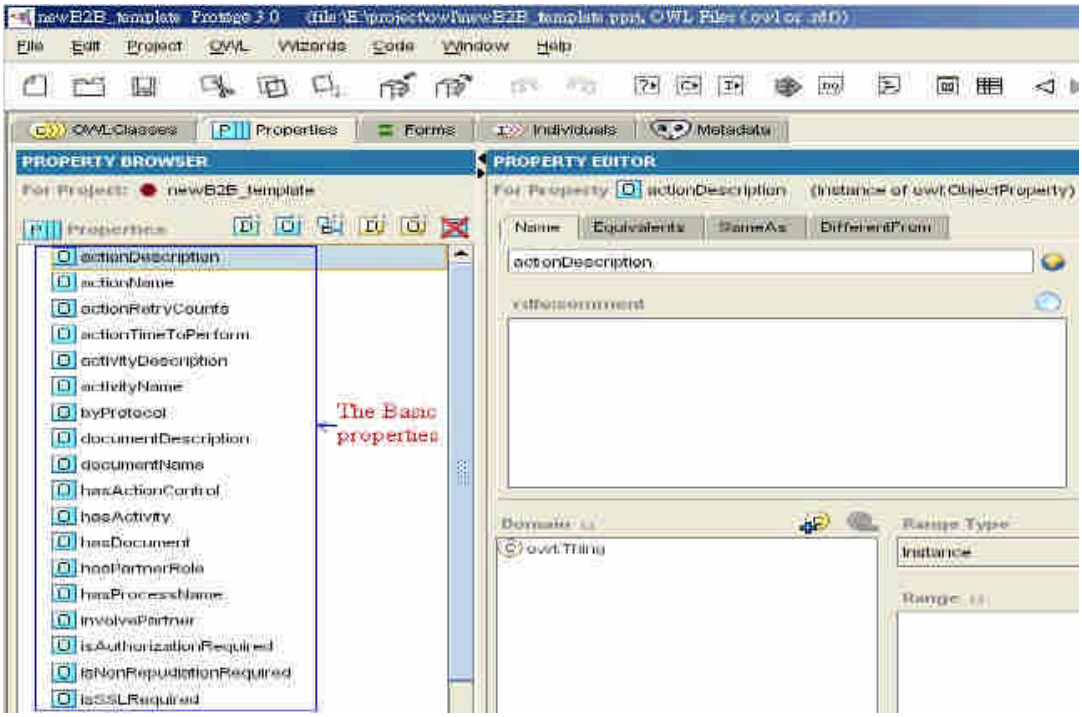


Figure 10: Ontology Instance Creation (This Research)

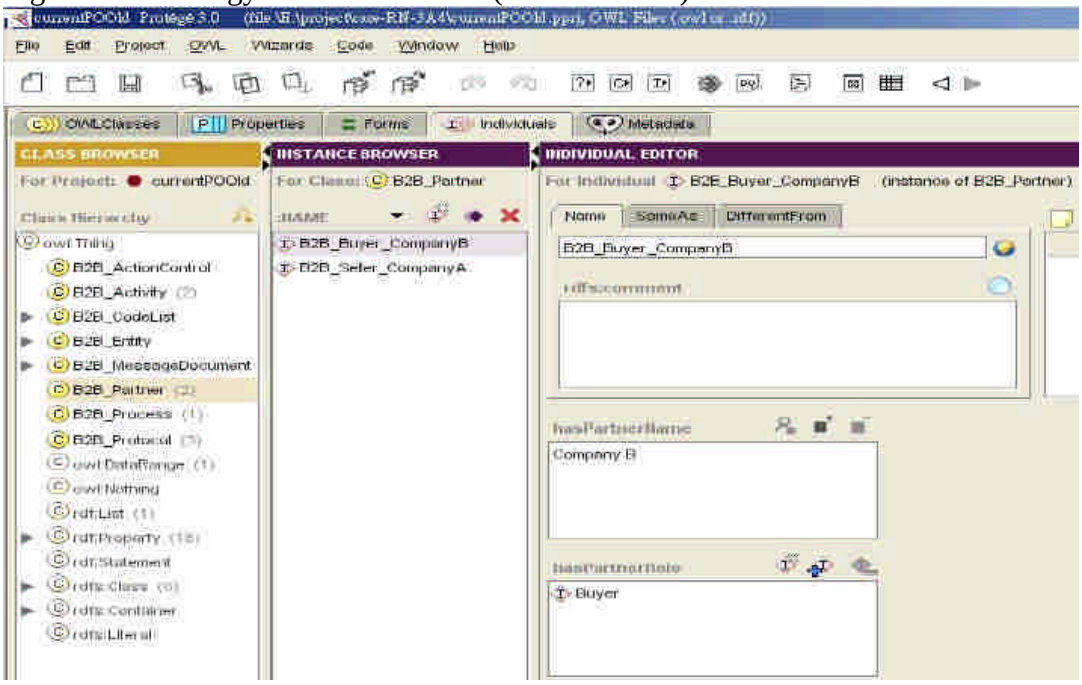


Figure 11: Existing Public Process Ontology (This Research)

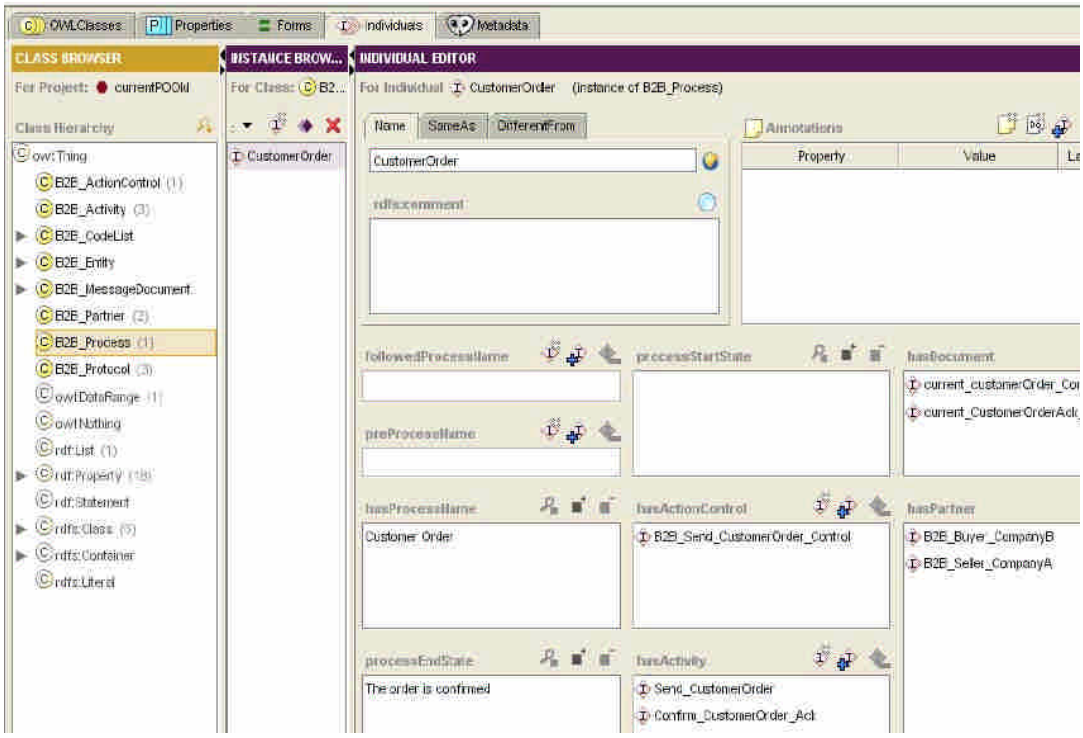


Figure12: Newly Generated Classes and Properties (This Research)

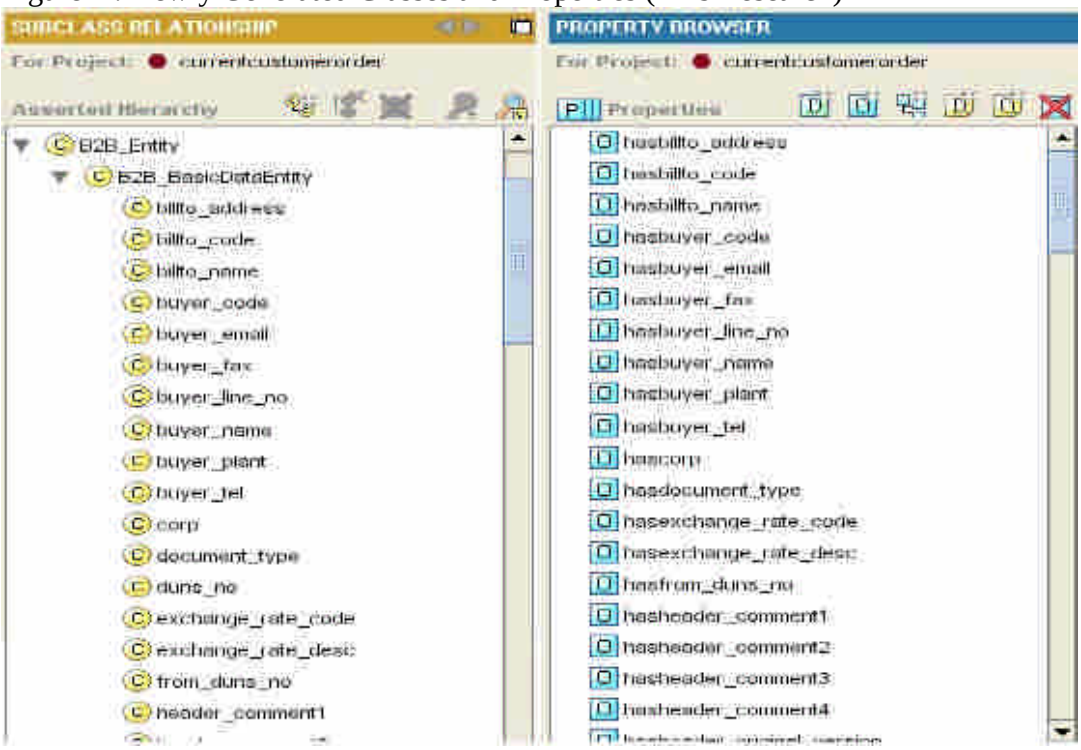


Figure 13: Created Instances of PIP3A4

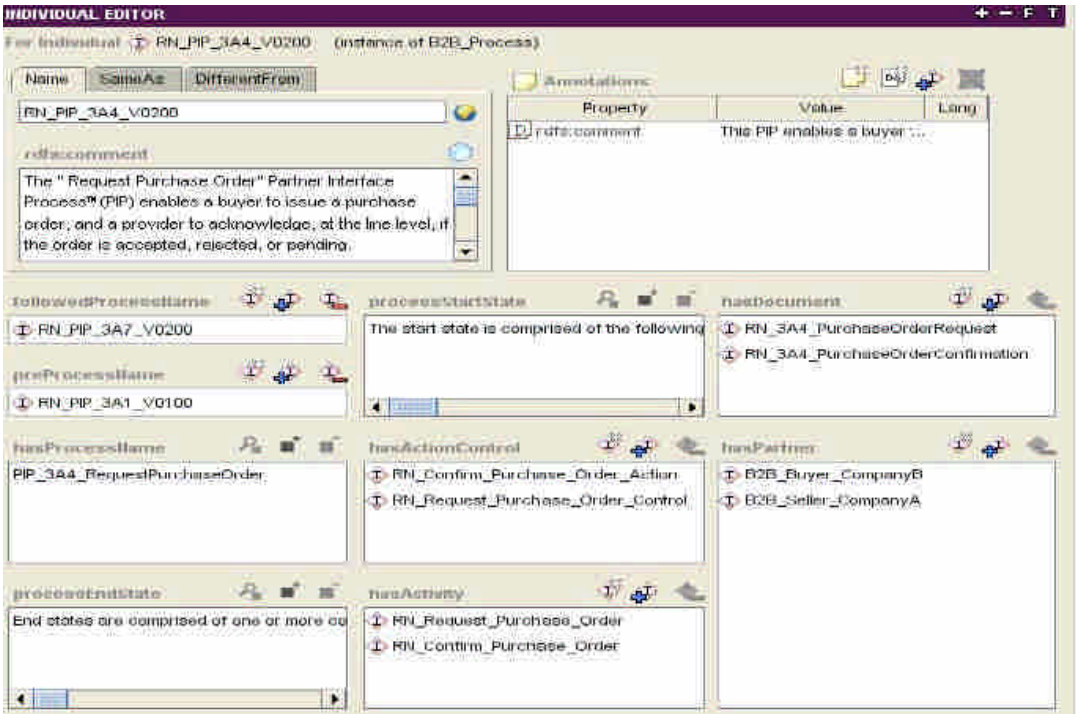


Figure 14: Equivalent Classes

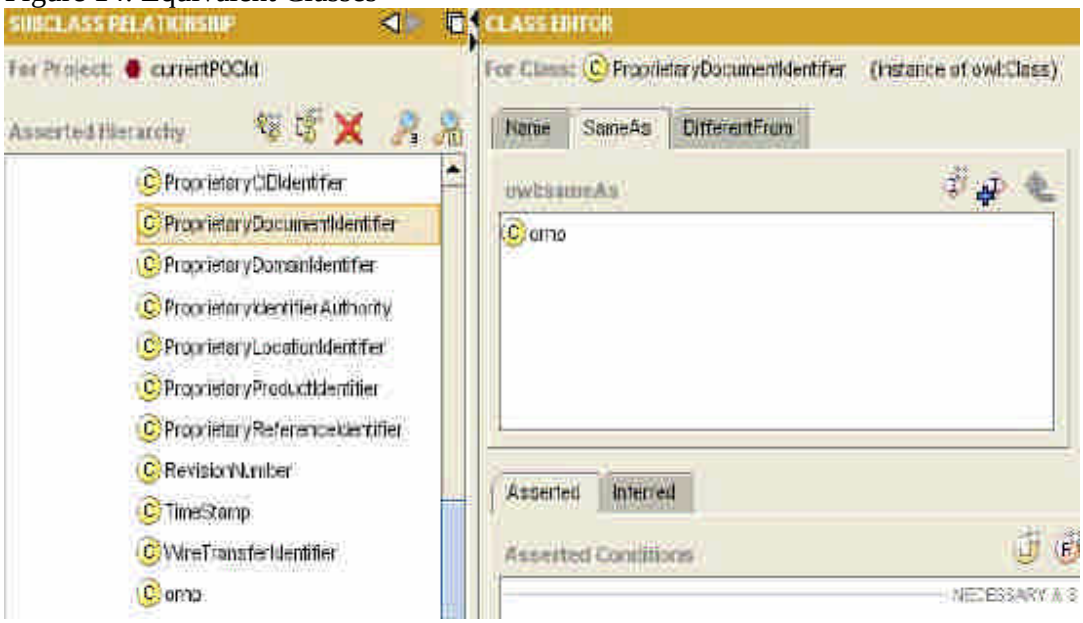
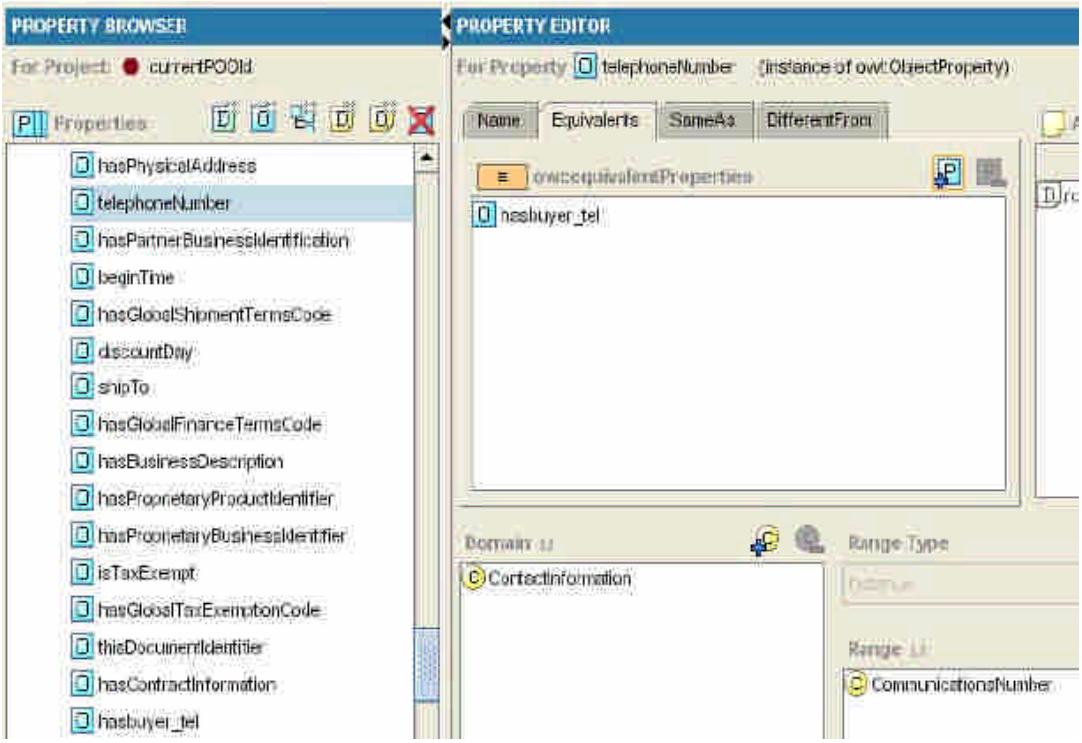
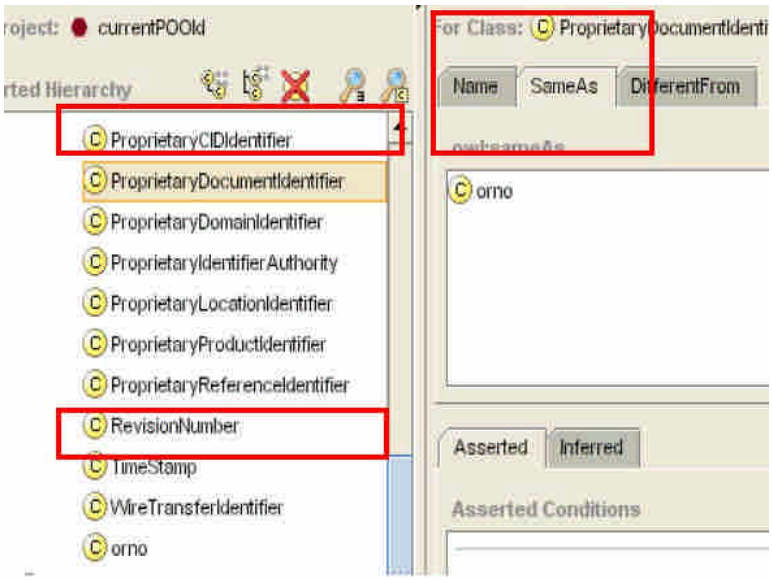


Figure 15: Equivalent Properties


PROPERTY BROWSER
For Project: currentPOOId
Properties
hasPhysicalAddress
telephoneNumber
hasPartnerBusinessIdentification
beginTime
hasGlobalShipmentTermsCode
discountDay
shipTo
hasGlobalFinanceTermsCode
hasBusinessDescription
hasProprietaryProductIdentifier
hasProprietaryBusinessIdentifier
isTaxExempt
hasGlobalTaxExemptionCode
thisDocumentIdentifier
hasContractInformation
hasbuyer_tel
PROPERTY EDITOR
For Property: telephoneNumber (instance of owl:ObjectProperty)
Name
Equivalents
SameAs
DifferentFrom
owl:equivalentProperties
hasbuyer_tel
Domain
ContactInformation
Range Type
Range
CommunicationsNumber



re 15: Equivalent Properties

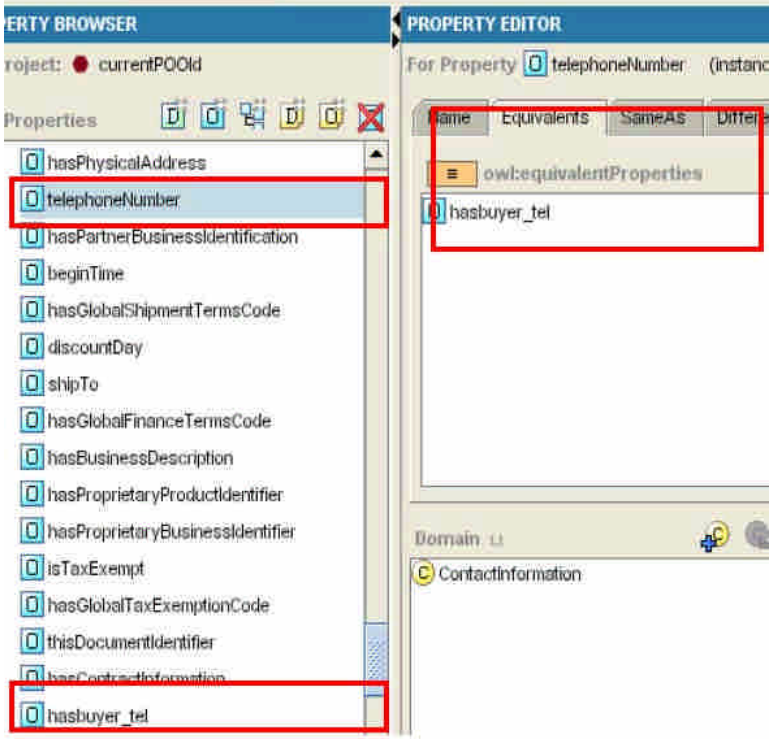


Table 1: Ontology Merge Rules (This Research)

Level	Type	Current (Old)	Standard (New)	Conflict Description	Merge Rules

Class Level	Schematic conflicts	None	New	Standard has a new class, which doest not exist in current process.	We keep the new class in the ontology. All the properties of the new class should be retained, too.
		Existed	None	The current process exist an old class, which does not appear in standard process.	If the old class will no longer exist in the future, we discard them; else we should add the old class to the new ontology.
	Semantic conflicts	Existed Class	New Class	They are with the different class names but the same meaning	We reserve the old class A and add it to new ontology. Then, we use the *owl:sameAs* to state the two classes are equivalent. However, we use the class B usually.
		Existed Class	New Class	They are with the same class name but different meanings.	We keep the name of the new class. However we change the name of old class to another new name.
Property Level	Schematic conflicts	None	New	There are additional properties in a class.	We use and adopt these properties in the new ontology.

		Existed	None	There are deletion properties in a class.	We have to determine whether the properties are no longer useful. If we do not use these properties any more, we discard them. We adjust the minimum cardinality of these old properties to 0 becausethey are not necessary properties in the new class.
	Semantic conflicts	Existed Property	New Property	They are with the different property names but the same meaning	We reserve the old property A and add it to new ontology. Then, we use the *owl:equivalentProperty* to state the two properties are equivalent.
		Existed Property	New Property	They are with the same property name but different meanings.	We keep the name of the new property. However we change the name of old property to another new name.