Chapter 2 Literature Review

2.1 B2B Integration

2.1.1 Introduction

Business-to-Business (B2B) integration is for exchanging information between different companies and the integration of cross-enterprise processes over the Internet. The traditional ways to exchange information between companies are through telephone, fax, or email. However, they are not convenient. These ways have the potential to introduce faults, like erroneous typing. Although the development of Electronic Data Interchange (EDI) provides a convenient way to exchange information, the EDI still has several defects. EDI is a transactional-based approach. It is not a real time or a dynamic system. EDI lacks the ability to quickly respond to business changes, and lacks scalability to provide for a large numbers of users. The EDI transaction is through a Value-added Network (VAN) which requires many transaction fees. As such, EDI is unsuitable for the twenty-first century.

The emergence of the Internet and XML is a reason to widen the next phase of B2B development. Many B2B standards have been proposed for XML and the Internet. They indeed provide a more dynamic and more flexible model than EDI. However, we will face difficulty when increasing the number of trading partners, processes and B2B protocols. These B2B standards lack the integration.

B2B integration platform (shown in Figure2-1) has been developed to solve the integration issue. The B2B integration platform is an information system that allows enterprises to exchange business document over networks and provide the functionality required by the B2B standard specification. It should provide the latest technology including Web-based services and XML, as well as the ability to transport through popular B2B protocols. This kind of software usually provides various B2B protocols for connecting to numerous trading partners. It also provides the ability to streamline the business process and adapters for linking various enterprise information systems. Furthermore, it needs to provide for mapping between business data.

Medjahed and Others (2003) mention that there are three layers in the B2B integration framework. They are the communication layer, the content layer and the process layer. They also identify a set of dimensions for studying the interaction issue in B2B e-Commerce. The dimensions are: coupling among partners, heterogeneity, autonomy, external manageability, adaptability, security, and scalability. Next, they compare several technologies and standards with the layers and dimensions.

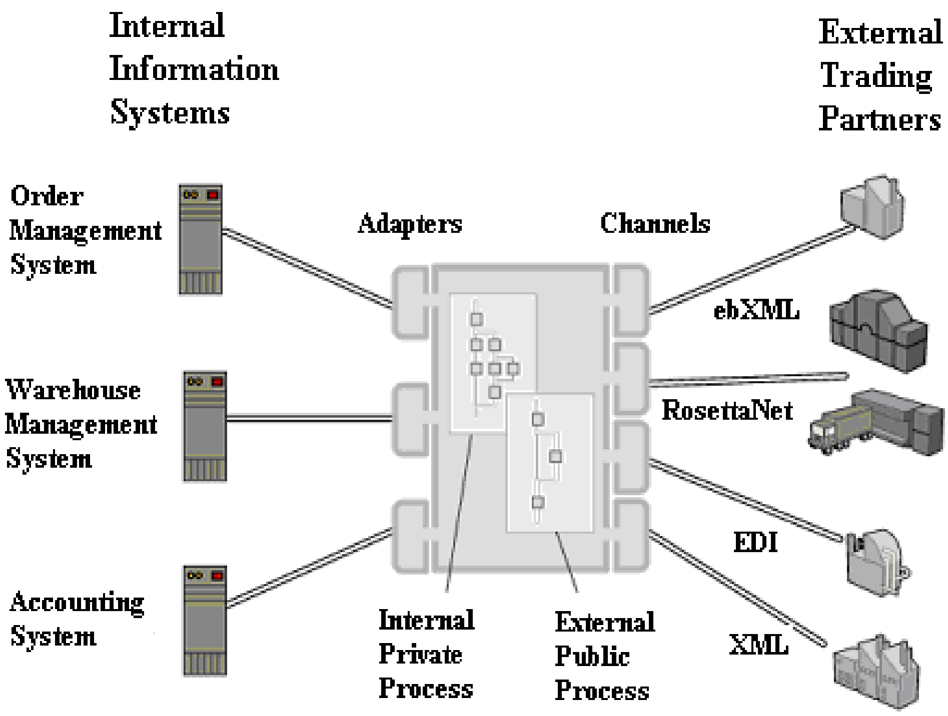

Figure 2-1: A Simple B2B Integration Platform (This Research)

We summarize the dimensions that need to be integrated in the B2B integration framework as follows:

- The B2B protocols
- The trading partners
- The business processes
- The enterprise applications
- The business data

The B2B Standard defines how information is exchanged between two enterprises. In general, it is the description of the message formats, the transport protocol, the process, and the security mechanism to be provided, to name a few.

Table 2-1: The Comparison of Various XML-Based B2B Standards (Liang, 2001)

	cXML	CBL	RNet	Biztalk	OAG	ebXML
Implementation Framework						
Protocol	◆	◆	◆	◆	◇	◆
Message Structure	◆	◇	◆	◆	◇	◆
Conversation	◆	◔	◆	◆	◇	◆
Secutrity	◆	◇	◆	◆	◇	◆
Partner Agreement	◇	◇	◇	◇	◇	◆
Data Dictionary	◔	◔	◆	◔	◐	◆
Vocabulary	◐	◐	◆	◔	◆	◆
Process						
Process Model	◐	◐	◆	◇	◔	◆
Process MetaModel	◇	◇	◔	◇	◇	◆
Process Repository	◇	◇	◔	◇	◇	◆

Table 2-1 lists several B2B standards and compares them. Liang (2001) argues that a well-designed B2B standard should have four components. They are: implement framework, data dictionary, vocabulary and process. RosettaNet and Electronic Business using eXtensible Markup Language (ebXML) are the most noticeable and famous B2B standards. Furthermore, the design of RosettaNet and ebXML has the aforementioned four components. We discuss them individually in following sections and select them as our main targets in this research.

2.1.2 ebXML

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 e-business standards. The consortium produces more Web services standards than any other organization along with standards for security, e-business, 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.

We use Figure 2-2 (ebXML Technical Architecture Specification, 2001) to depict the interaction between two companies conducting eBusiness using ebXML. Company A has become aware of an ebXML Registry that is accessible on the Internet (step 1). After reviewing the contents of the ebXML Registry, Company A decides to build and deploy its own ebXML compliant application (step 2). Then, Company A submits its own Business Profile information (including implementation details and reference links) to the ebXML Registry (step 3). The business profile submitted to the ebXML Registry describes the company's ebXML capabilities and constraints, as well as its supported business scenarios. After receiving the business profile and verification that the format and usage of a business scenario is correct, ebXML Registry sends an acknowledgment to Company A (step 3). Company B discovers the business scenarios supported by Company A in the ebXML Registry (step 4). Company B sends a request to Company A stating that they would like to engage in a business scenario using ebXML (step 5). Company B acquires an ebXML compliant application. Before engaging in the scenario, Company B submits a proposed business arrangement directly to Company A's ebXML compliant software Interface. The proposed business arrangement outlines the mutually agreed upon business scenarios and specific agreements. The business arrangement also contains information pertaining to the messaging requirements for transactions to take place, contingency plans, and security-related requirements (step 5). Company A then accepts the business agreement. Company A and B are now ready to engage in eBusiness using ebXML (step 6).

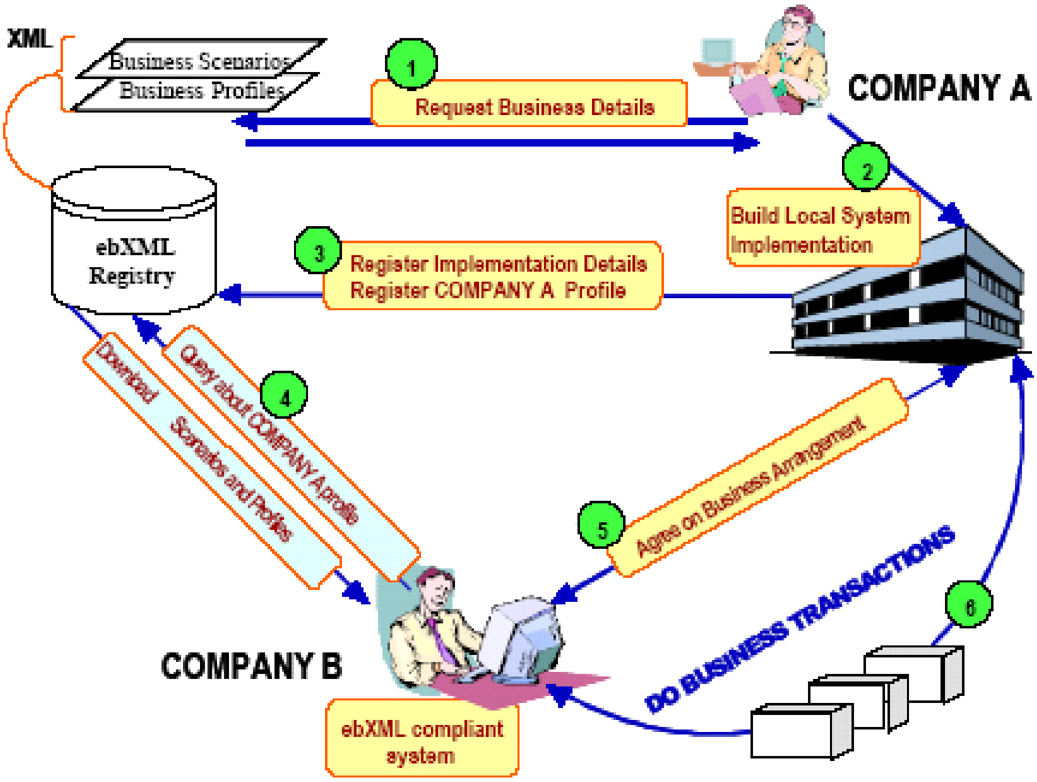

Figure 2-2: A High Level Overview of The Interaction of Two Companies Conducting eBusiness Using ebXML (ebXML Technical Architecture Specification, 2001)

In order to implement the above scenario, ebXML has several foundation specifications:

Business Process

The ebXML Business Process and Information Meta Model (ebXML, 2004) is a mechanism that allows trading partners to capture details for a specific business scenario using a consistent modeling methodology. A Business Process describes in detail how trading partners take on roles, relationships and responsibilities to facilitate interaction with other trading partners in shared collaborations. The interaction between roles takes place as a choreographed set of business transactions. Each business transaction is expressed as an exchange of electronic business documents. Business documents may be composed from re-useable Business Information Objects. At a lower level, Business Processes can be composed of re-useable Core Processes, and Business Information Objects can be composed of re-useable Core Components. The ebXML Business Process and Information Meta Model supports requirements, analysis and design viewpoints that provide a set of semantics (vocabulary) for each viewpoint and forms the basis of specification of the artifacts that are required to facilitate Business Process and information integration and interoperability. An additional view of the Meta Model, the Business Process Specification Schema (BPSS), is also provided to support the direct specification of the set of elements required to configure a runtime system in order to execute a set of ebXML business transactions. By drawing out modeling elements from several of the other views, the Specification Schema forms a semantic subset of the ebXML Business Process and Information Meta Model. The Specification Schema is available in two stand-alone representations, a UML profile, and a DTD.

Core Components

A Core Component (ebXML, 2004) captures information about a real world business concept, and the relationships between that concept, other Business Information Objects, and a contextual description that describes how a Core or Aggregate Information Entity may be used in a particular ebXML eBusiness scenario. A Core Component can be either an individual piece of business information, or a group of Business Information Objects that may be assembled into Aggregate Information Entities.

Registry

An ebXML Registry provides a set of services that enable the sharing of information between Trading Partners. A Registry is a component that maintains an interface to metadata for a registered item. Registry Services provides the interfaces (APIs) to allow trading partners for accessing an ebXML Registry.

Collaboration Protocol Profile and Agreement

To facilitate the process of conducting eBusiness, potential Trading Partners need a mechanism to publish information about the Business Processes they support along with specific technology implementation details about their capabilities for exchanging business information. This is accomplished through the use of a Collaboration Protocol Profile (CPP). The CPP is a document which allows a Trading Partner to express their supported Business Processes and Business Service Interface requirements in a manner where they can be universally understood by other ebXML compliant Trading Partners. A special business agreement called a Collaboration Protocol Agreement (CPA) is derived from the intersection of two or more CPP's. The CPA serves as a formal handshake between two or more Trading Partners wishing to conduct business transactions using ebXML (ebXML, 2004).

Message service

The ebXML Message Service mechanism provides a standard way to exchange business Messages among ebXML Trading Partners. The ebXML Messaging Service provides a reliable means to exchange business Messages without relying on proprietary technologies and solutions. An ebXML Message contains structures for a Message Header (necessary for routing and delivery) and a Payload section (ebXML, 2004).

2.1.3 RosettaNet

RosettaNet consortium (RosettaNet Consortium, 2004) is a non-profit consortium of more than 500 organizations working to create, implement and promote open e-business standards and services. RosettaNet tries to establish a common language, a standard processes for the electronic sharing of business information. Companies adopting RosettaNet standards will strengthen trading-partner relationships, raise productivity and reduce costs. Furthermore, the quick response feature of RosettaNet boosts performance of the global supply chain management. End users enjoy speed and uniformity, extending across various levels of business communications.

The foundation of the RosettaNet standard includes the RosettaNet Implement Framework (RNIF), Partner Interface Processes (PIPs), Dictionaries, and Product & Partner Codes. We extract the content of these foundational components from RosettaNet (RosettaNet Consortium, 2004):

RNIF

The RNIF is a technical-driven specification that specifies how RosettaNet PIPs can be exchanged via the Internet. RNIF specifies the components of a RosettaNet business message, the flow of messages, and how the message is packed and unpacked. It also specifies the RosettaNet security requirements and mechanism. Most importantly, RNIF specifies how to transfer the RosettaNet business message and how to handling the exception.

PIPs

A major part of RosettaNet's standardization effort is alignment of business processes between trading partners in a given supply chain (such as the IT Products and Electronic Component supply chains). RosettaNet specifies these as PIP specifications.

RosettaNet divides the entire e-business supply chain domain for which PIPs are specified into broad classifications called “clusters.” Each cluster is further subdivided into two or more “segments.” Each segment comprises several PIPs. PIPs contain one or more Activities, and Activities in turn specify Actions. An example of this relationship is shown as follows:

- CLUSTER 3: Order Management
    - Segment A: Quote and Order Entry
        - PIP 3A4: Manage Purchase Order
            - Activity: Create Purchase Order
                - Action: Purchase Order Request
    - Segment B: Transportation and Distribution
    - Segment C: Returns and Finance
        - Segment D: Product Configuration

Each PIP in a segment represents a well-defined business process subset, and is named with the cluster, segment, and sequence number of the PIP in the segment. For example the Manage Purchase Order PIP is fourth in sequence in Segment A (Quote and Order Entry) of the Cluster 3 (Order Management). Hence the Manage Purchase Order PIP is identified by the name PIP3A4. PIPs include specification of partner business roles (Buyer, Seller etc.); business activities involved between the roles; and type, content, and sequence of business documents exchanged by the role -interactions while performing these activities. They also specify the time, security, authentication, and performance constraints of these interactions. Structure and content of the business documents exchanged is specified through XML Document Type Definitions (DTDs) or XML Schema, and associated Message Guidelines. Trading partners that participate in the PIP exchange business documents that conform to the DTDs and Message Guidelines in the subject PIP specification, using network protocols that are specified and supported by the RosettaNet Implementation Framework.

Dictionary

RosettaNet dictionaries provide a common set of properties for PIPs. RosettaNet builds two dictionaries. The RosettaNet Business Dictionary (RNBD) defines the Business Properties, Business Data Entities, and Fundamental Business Data Entities in PIP Message Guidelines. The RosettaNet Technical Dictionary (RNTD) provides common properties for defining products for RosettaNet PIPs.

RNTD and RNBD provide a common vocabulary for conducting e-business, eliminating confusion in the procurement process due to many companies' uniquely defined terminology. The RNTD eliminates the need for partners to utilize separate dictionaries when implementing multiple PIPs and is no longer supply chain-specific, allowing it to be used in a variety of supply chain applications.

Product & Partner Codes

Product and partner codes in RosettaNet standards expedite the alignment of business processes between trading partners. RosettaNet specifies three major codes currently.

- GTIN

RosettaNet specifies the Global Trade Item Number (GTIN) for Global Product Identifier in its PIPs. The GTIN is a worldwide multi-industry standard for trade-item identification. GTINs are 14-digit numbers that uniquely and globally identify products and services.

- UN/SPSC

RosettaNet specifies the UN/SPSC for Global Class Identifier in its PIPs. The UN/SPSC is a code standard for classifying products and services. Items are classified using numbers derived from the system's five-level hierarchy in which two digits are assigned at each level. The UN/SPSC allows worldwide trading partners to uniformly classify products and services, resulting in accuracy and efficiency throughout the trading network. Within a RosettaNet PIP, the UN/SPSC is referred to as the *GlobalProductClassificationCode*.

- D-U-N-S

RosettaNet specifies the Data Universal Numbering System (D-U-N-S®) for Global Company Identifier in its PIPs. The nine-digit D-U-N-S Number is a worldwide standard for company identification, distinguishing unique business locations around the globe. D-U-N-S Numbers are assigned and maintained by Dun and Bradstreet (D&B). D-U-N-S Numbers enable organizations to clearly identify trading partners as well as accurately gauge risks and opportunities. Within a RosettaNet PIP, the D-U-N-S Number is referred to as the *GlobalBusinessIdentifier*, and the D-U-N-S+4 Number is referred to as the *GlobalLocationIdentifier*.

2.1.4 Open Issues

Although the appearance of XML brings a great benefit to e-Commerce but XML only provides the syntax in exchanging data not semantics, it does not guarantee interoperability either.

RosettaNet and ebXML are the most advanced B2B standards. The ebXML is more like a horizontal B2B standard and the RosettaNet is a vertical one. A company may implement many B2B standards to meet the requirements of trading partners. How to integrate between the horizontal and the vertical standards? How to integrate between different vertical B2B standards? They will be the critical issue.

RosettaNet plans to integrate support for the ebXML Messaging Services Specification in future releases of RNIF, however, this kind of integration focus on technical and syntax parts. RosettaNet does not mention how to align semantics with ebXML.

Interoperability and heterogeneity of B2B standards is the main issue in B2B integration.

In another respect, most B2B standards define the message syntax through DTD or XML Schema. B2B standards do not represent message semantics of messages or process them in a machine-readable method. Computers can understand message syntax, however they cannot understand the semantic parts.

To understand business messages defined by B2B standard is another issue of B2B integration. The PIP3A4 Version 02_03 messages in RosettaNet are a good example. PIP3A4 Version 02_03 has two messages. One is PurchaseOrderRequest and another is PurchaseOrderConfirmation. The message guideline for PurchaseOrderRequest lists 557 elements and PurchaseOrderConfirmation lists 712. However, not all elements will be used. A large number of elements will increase the effort of B2B initiative implementation. In this situation, trading partners have to select which elements will be used in the business message. The process to select and study elements wastes a lot of time, but trading partners must do it nevertheless. Although RosettaNet specifies the structure of the messages, trading partners must decide the actual details of the messages.

Furthermore, many defined elements are repeated even though the meaning of the elements will sometimes be changed. The change of meaning depends on its positioning in the message. This is not an easy thing to control, especially confronted by hundreds of elements.

2.2 Semantic Web and Ontology

2.2.1 Introduction

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." (Tim Berners-Lee, 2001). Tim Berners-Lee states that a goal of the Semantic Web is it should be useful not only for human-human communication, but also that machines should be able to process and interpret its contents. The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. It is a collaborative effort led by the World Wide Web Consortium (W3C) with participation from a large number of researchers and industrial partners (W3C, 2004). It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming (shown in Figure 2-3). The layer on top of RDF and RDF Schema is the Ontology layer. W3C proposes the Web Ontology Language (OWL) for the Ontology layer. We discuss RDF, RDF Schema and OWL in section 2.2.2. The researches and development of Logic layer, Proof layer and Trust layer are just beginning.

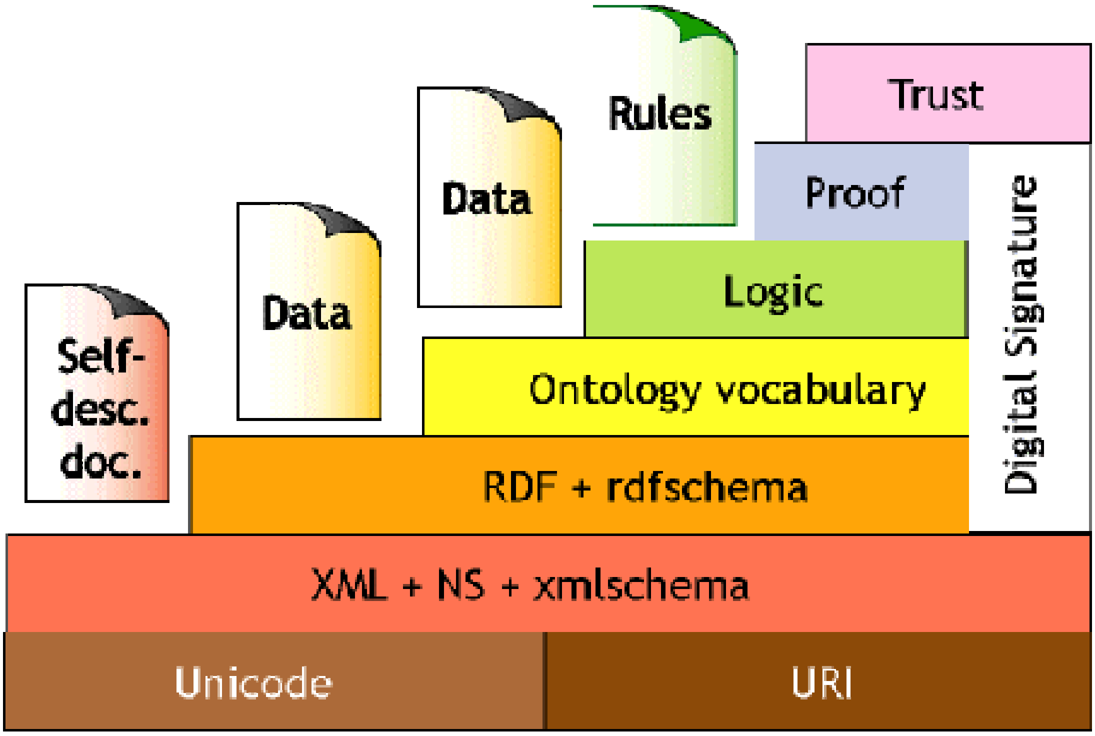

Figure 2-3: The Semantic Web Layer Cake (Tim Berners-Lee, 2000)

Ontology is an explicit specification of a conceptualization (Gruber, 1993). Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related (Web Ontology Language, 2004). For the web, ontology is about the exact description of web information and relationships between web information.

An ontology differs from an XML schema in that it is a knowledge representation, not a message format. Ontology is usually applied to Artificial Intelligence (AI) and knowledge management. Ontology can reason easier and represent the semantics of a domain. Ontology is the key technology of semantic web.

2.2.2 Ontology Languages

RDF and RDFS

The RDF is a framework for representing information in the Web. It is particularly intended for representing metadata about Web resources, such as title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource. However, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web. Examples include information about items available from on-line shopping facilities. RDF is designed to represent information in a minimally constraining, flexible way. It can be used in isolated applications, where individually designed formats might be more direct and easily understood, but RDF's generality offers greater value from sharing (W3C, 2004).

RDF is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs), and describing resources in terms of simple properties and property values. All things described by RDF are called resources. Resources may be divided into groups called classes.

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. Each triple represents a statement of a relationship between the things denoted by the nodes that it links. A set of such triples is called an RDF graph (Figure 2-4). This can be illustrated by a node and directed-arc diagram, in which each triple is represented as a node-arc-node link.

Figure 2-4: The RDF Graph (W3C, 2004)

A RDF triple is conventionally written in the order subject, predicate, object. The subject is a RDF URI reference or a blank node. The predicate is a RDF URI reference and the object is a RDF URI reference, a literal or a blank node. A blank node is a node that is not a URI reference or a literal. A blank node is just a unique node that can be used in one or more RDF statements, but has no intrinsic name. The predicate is also known as the property of the triple.

However, RDF provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. That is the role of the RDF vocabulary description language, RDF Schema. RDF Schema defines classes and properties that may be used to describe classes, properties and other resources. RDF Schema defines a simple modeling language on top of RDF. RDF Schema introduces classes, is-a relationships between classes and properties, and domain and range restrictions for properties as modeling primitives.

The feature of RDF Schema is that it does not attempt to enumerate all the possible vocabulary description that is useful for representing the meaning of RDF classes and properties. Instead, RDF Schema allows other richer vocabulary or Ontology languages such us OWL or other techniques utilizing its framework.

DAML +OIL

RDF is very similar to a directed graph and this simplicity makes it a sort of assembly language on top of which other information modeling methods can be overlaid. However it lacks in catering for data typing, for a consistent expression that defines enumerations and other facilities.

Therefore, the DAML group pooled efforts with OIL to provide a more sophisticated classification, using constructs from frame-based Artificial Intelligence. This resulted in a language called DAML+OIL for expressing far more sophisticated classifications and properties of resources than RDFS. DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modeling primitives. DAML+OIL provides modeling primitives commonly found in frame-based languages (DAML+OIL Language, 2001).

OWL

OWL (Web Ontology Language, 2004) is a language for defining and instantiating Web ontologies. OWL is a revision of the DAML+OIL incorporating lessons learned from the design and application of DAML+OIL. An OWL ontology may include descriptions of classes, properties and their instances.

One advantage of OWL ontologies will be the availability of tools that can reason with them. Tools will provide generic support that is not specific to the particular subject domain, which would be the case if one were to build a system to reason about a specific industry-standard XML schema.

OWL is a component of the Semantic Web project. This effort aims to make Web resources more readily accessible to automated processes by adding information about the resources that describe or provide Web content.

OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema by providing additional vocabulary along with a formal semantics.

The OWL language provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users (Web Ontology Language, 2004).

1. OWL Lite: OWL Lite supports those users primarily needing a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and provide a quick migration path for thesauri and other taxonomies.
2. OWL DL: OWL DL supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL includes all OWL language constructs with restrictions such as type separation (a class can not also be an individual or property, a property can not also be an individual or class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied a particular decidable fragment of first order logic. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems.
3. OWL Full: OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. Another significant difference from OWL DL is that a `owl:DatatypeProperty` can be marked as an `owl:InverseFunctionalProperty`. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

Ontology developers adopting OWL should consider which sublanguage best suits their needs. The choice between OWL Lite and OWL DL depends on the extent to which users require the more-expressive constructs provided by OWL DL. The choice between OWL DL and OWL Full mainly depends on the extent to which users require the meta-modeling facilities of RDF Schema (e.g. defining classes of classes, or attaching properties to classes). When using OWL Full as compared to OWL DL, reasoning support is less predictable since complete OWL Full implementations do not currently exist. OWL Full can be viewed as an extension of RDF, while OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF.

2.3 Ontology Application in e-Commerce

In this section, we discuss several previous researches into the marriage of e-Commerce and semantic web technology.

2.3.1 Ontology and e-Commerce

2.3.1.1 Introduction

An e-Marketplace is a web site to gather vendors and customers conducting business. The main problem of a B2B e-Marketplace is the heterogeneity of information descriptions used by vendors and customers. The heterogeneity arises in three levels: the level of content, the level of product catalogs structures, and the level of document structures (Ding and Fensel et al. 2003).

The content level: This type of mismatch is mainly concerned with the real-world semantics of the exchanged information. People describe the same products in different ways. Historically, many different ways to categorize and describe products have evolved. Often, vendors have their own way of describing their products. Structuring and standardizing the product descriptions is a significant task in B2B e-Commerce, ensuring that different partners can actually communicate with each other, and allowing customers to find the products they are looking for.

The catalog level: E-Commerce is about the electronic exchange of business information in which product descriptions are just one element. The product descriptions are the building blocks of an electronic catalog, together with information about the vendor, the manufacturer, and the lead-time etc. Furthermore, a catalog provider needs to include quality control information, such as the version, date and identification number of the catalog. If two electronic catalogs are involved the structure of these catalogs has to be aligned as well. This type of mismatch arises more in relation to the syntactical structure of the exchanged information.

The document level: A buyer may want to send a purchase order, after picking up the necessary information from a catalog. The vendor has to reply with a confirmation, and the actual buying process begins. In order for the buyer and the vendor to read and process each other's business documents, again a common language is needed. Such common languages are RosettaNet, xCBL, and cXML. This type of mismatch also arises in relation to the syntactical structure of the exchanged information.

Ding and Fensel propose that ontology can align the heterogeneity at the three levels. They use RDF/RDFS and RDF Transformation (RDFT)(Omelayenko, 2002) to align the heterogeneity in document level and content level. The RDFT mapping meta-ontology specifies a small language for mapping XML DTDs to/and RDF Schemas specially targeted for business integration tasks.

They also build a software environment, called GoldenBullet, to support product classification according to certain content standards. This research clearly mentions that ontology technology can apply to the content, catalog structure and document level.

Besides the three levels, RDFT also can be applied to solve the process level (Omelayenko, Fensel, and Bussler, 2002).

2.3.1.2 Content Management

Content management is a critical issue of the B2B e-marketplace. We use the content standard to manage the products. The content standard describe the specification of the products with product classification hierarchies and product attributes. There are many content standards in the world, and they can be classified into horizontal standards and vertical standards. Omelayenko (2001) argues that the e-marketplace should provide the user with different kinds of standards. Furthermore, it should integrate the standard and allow the standards to be transferred each other.

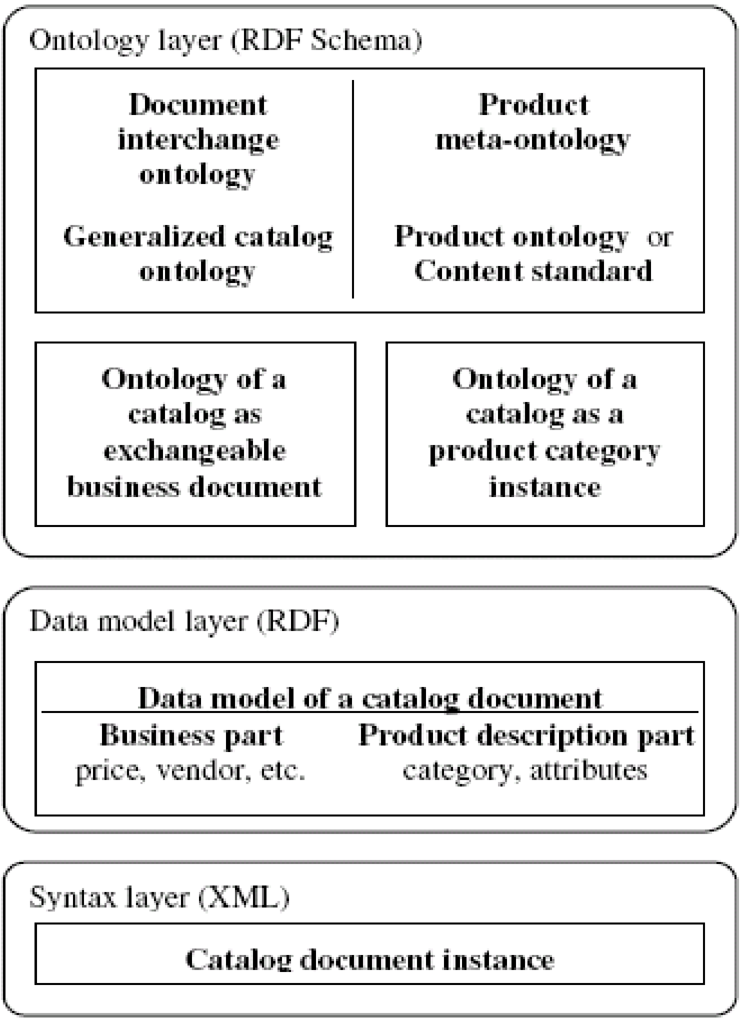

Figure 2-5: The Multi-Layered Model for Content Integration

Omelayenko and Fensel (2001) propose the multi-layered model for content integration. There are three layers in this model. These include a Syntax layer, a Data model layer, and an Ontology layer (shown in Figure 2-5). Omelayenko proposes methods to transfer from each layer and align the content standards. When we try to transfer between the content standards, the standards cannot be transferred directly. They should be transferred in the data model layer or ontology layer (shown in Figure 2-6). The author uses the RDF and RDF Schema in these layers. Such the different standards have the same data model that they can map each other through the RDFT.

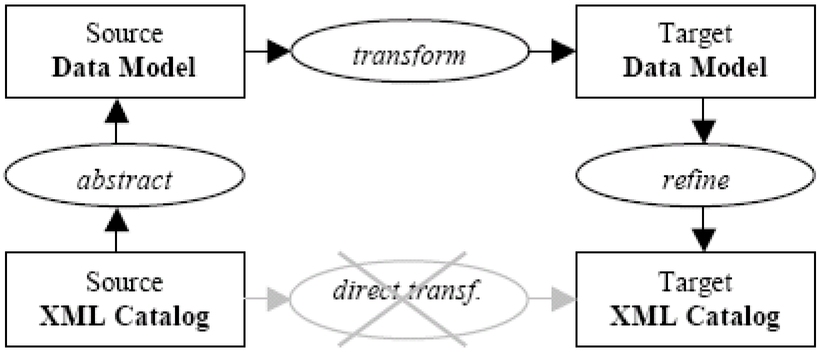

Figure 2-6: The Model for Data Transformation (Omelayenko and Fensel, 2001)

In some types of e-marketplace the content standard integration will not be so important. For example, an e-marketplace established by an industrial giant. In this situation, the company will lead the development of this exchange hub, including the product standard it requires. The customers or suppliers have no choice of standards. Hence the content standard integration will not necessary here. It is only suitable for the e-marketplace built by a third-party company where the bargain power is equivalent between the buyers and the sellers.

2.3.1.3 Using RDFT

The basic class diagram is presented in Figure 2-7, where the classes are represented by their names, and name nesting indicates the relationship.

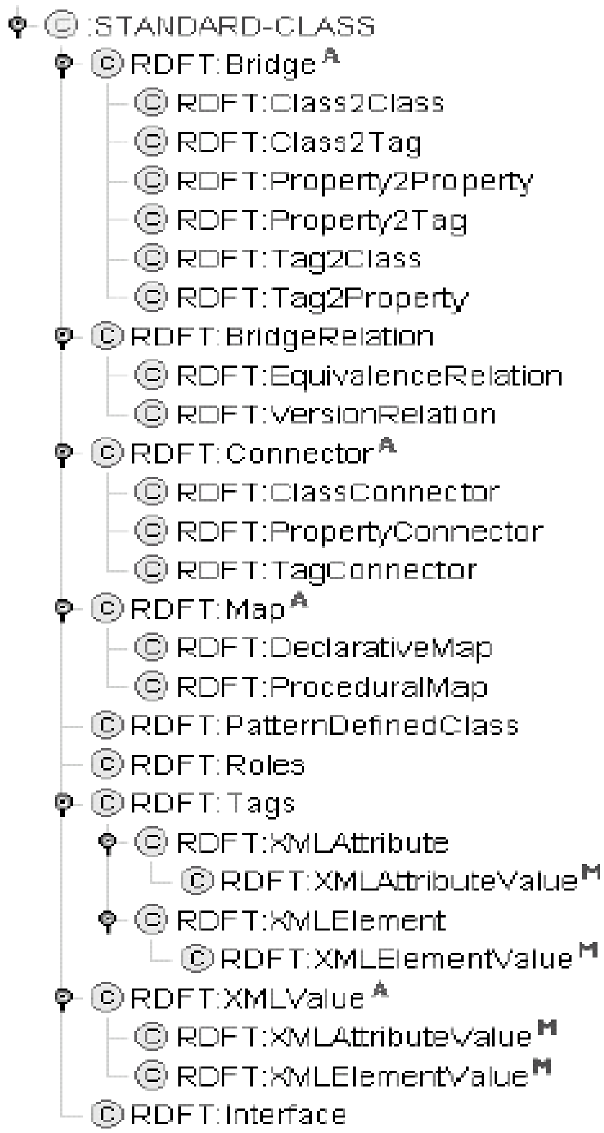

Figure 2-7: RDFT Class Diagram (Omelayenko, Fensel, and Bussler, 2002)

The main concept of RDFT is the bridge between two sets of rdf:Resources (two sets of concepts), one of which is regarded as the source set, and the other one as the target set. The bridges are grouped into maps. Each map is a collection of bridges serving a single purpose.

An abstract class Bridge describes common properties of bridges allowing only one-to-many and many-to-one bridges. Each Bridge contains the *ValueCorrespondance* property linking a map that specifies the necessary transformations of instance values of the source and the target concepts linked by the bridge.

2.3.2 Semantic B2B Integration

In early periods, we take the programming approach to customize each B2B connection for B2B integration. It is not flexibility if there is a lot of partners involve in and lots of B2B protocols, back-end system need to integrate. Bussler (2001) proposes the architecture of semantic B2B integration (shown in Figure 2-8). He believed that B2B integration should contain the following concepts: B2B protocol, application protocol, workflow type, transformation, process binding, trading partner and business event. The integration should be modeled. Any new B2B protocol standard can be dynamically added and supported by a B2B protocol engine. The B2B protocol engines and B2B integration engines are agnostic to B2B protocol standards. Those concepts could be modeled from the modeling languages then a B2B integration server could execute the concepts. From the semantics viewpoint, Bussler (2002) declared that the modeling language provides the abstractions and execution semantic appropriate for modeling integration.

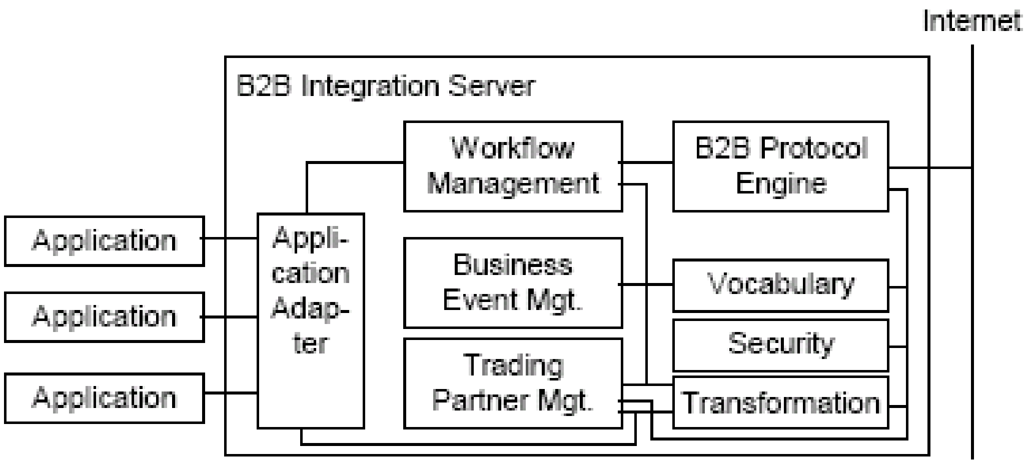

Figure 2-8: B2B Integration Architecture (Bussler, 2001)

Bussler highlights the most important semantic integration concepts in the following list (2001):

- **Trading partner**. Messages are exchanged between trading partners. Each trading partner is uniquely identified.
- **B2B protocol**. Trading partners have to agree on the B2B protocol used that consistently and reliably exchanges messages.
- **Network**. Different types of networks are available to trading partners for exchanging documents across the Internet and VANs.
- **Transformation**. Transformations ensure semantic equivalence between business data in messages over networks and their back end application system

equivalent implementation.

- **Common View**. A common view is the semantic unification of the same business data independent of the format in the different B2B protocols or back end applications.
- **Application Adapter**. An application adapter connects to a back end application extracting and inserting business data.
- **Workflow Management**. Workflow management implements the definition of processes extracting data from back end systems and sending them over a B2B protocol (and vice versa).
- **Business Event**. A business event is the occurrence of a state change indicating that business data are ready to be exchanged with trading partners over networks.

2.3.3 Semantic Web enabled Web Service (SWWS)

Web Service is becoming important technology in e-Commerce. Many applications adopt the Web Service approach to provide B2B functionality. Bussler, Fensel and Maesche (2002) integrate with Semantic Web and Web Service to build up the B2B integration architecture.

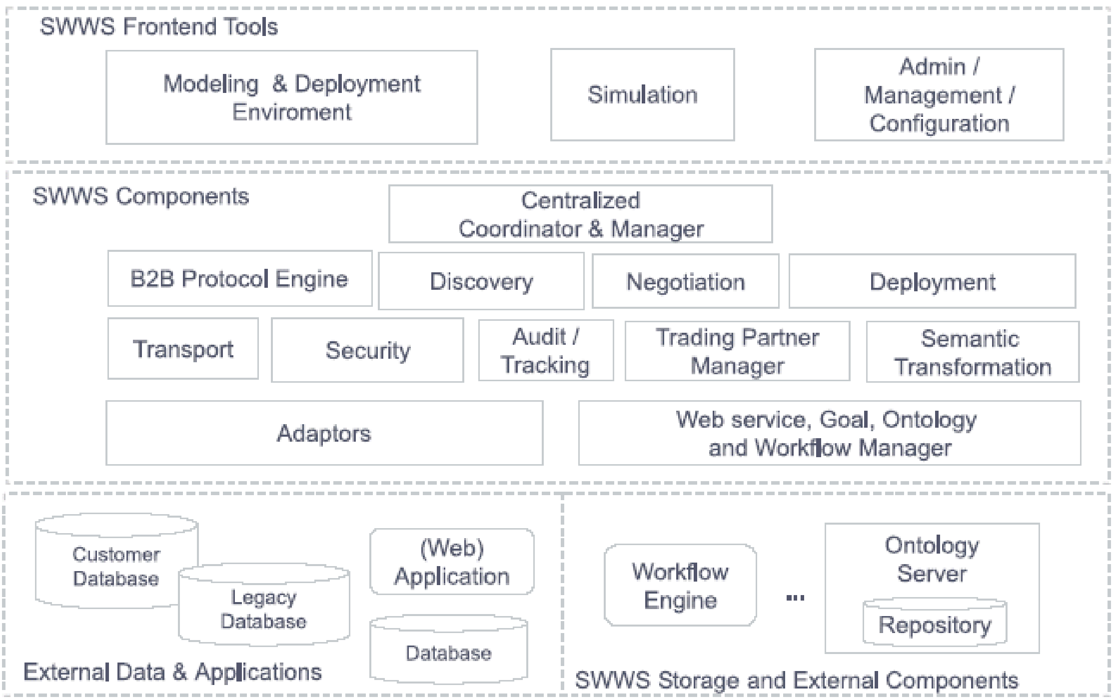

Figure 2-9: SWWS Conceptual Architecture (Bussler, Fensel and Maesche, 2002)

This architecture supports the semantic transformation. It will be supported by this component on different levels: Data, Business Logics, Message Exchange Protocols and Dynamic Service Invocation. Furthermore, Ontology supports the two SWWS concepts: document types and semantics. Document types are the definition of the structure of the business data that are communicated between trading partners. Document types refer to concepts, attributes of concepts and relationship between concepts that are ideally managed by and represented within ontologies. Ontologies can in a very precise manner define as well as manage concepts, attributes and relationships between concepts. Semantics refers to the correct population of attributes with correct domain values. In the architecture ontology servers and associated repositories are used for storing and accessing domain and application concepts (e.g. product names, measures and country codes) represented in the form of ontologies.

Another work that is done by the DAML-S consortium is closed to SWWS. Bussler’s framework as well as DAML-S uses the Semantic Web's key enabling technology of ontologies as their basis. In contrast to DAML-S, the underlying principles of this framework are the ideas of de-coupling and mediation. Furthermore, DAML-S misses important modeling constructs like the distinction between business logic and message exchange protocols. Thus, SWWS allows for a more flexible framework including greater abstraction and encapsulation of proprietary business information.

2.3.4 Domain Ontology Management Environment (DOME)

Solving queries to support e-Commerce transactions can involve retrieving and integration information from multiple information resources. One example is providing seamless access to a set of heterogeneous electronic product catalogues. Cui, Jones and O’Brien (2002) focus on the issue of semantic interoperability from the heterogeneous resources. These issues include developing ontology, mapping between ontology, ontologies and resource information, ontologies and database schemas and entity correspondence. They implement the prototype, called DOME. The DOME project has been researching and developing ontology-based techniques to support the building of a "one stop knowledge shop" for corporate information. They have developed a methodology, a set of tools and an architecture to enable enterprise-wide information management for data re-use and knowledge sharing. The system retrieves information from multiple resources to answer user queries and presents the results in a consistent way that is meaningful to the user.

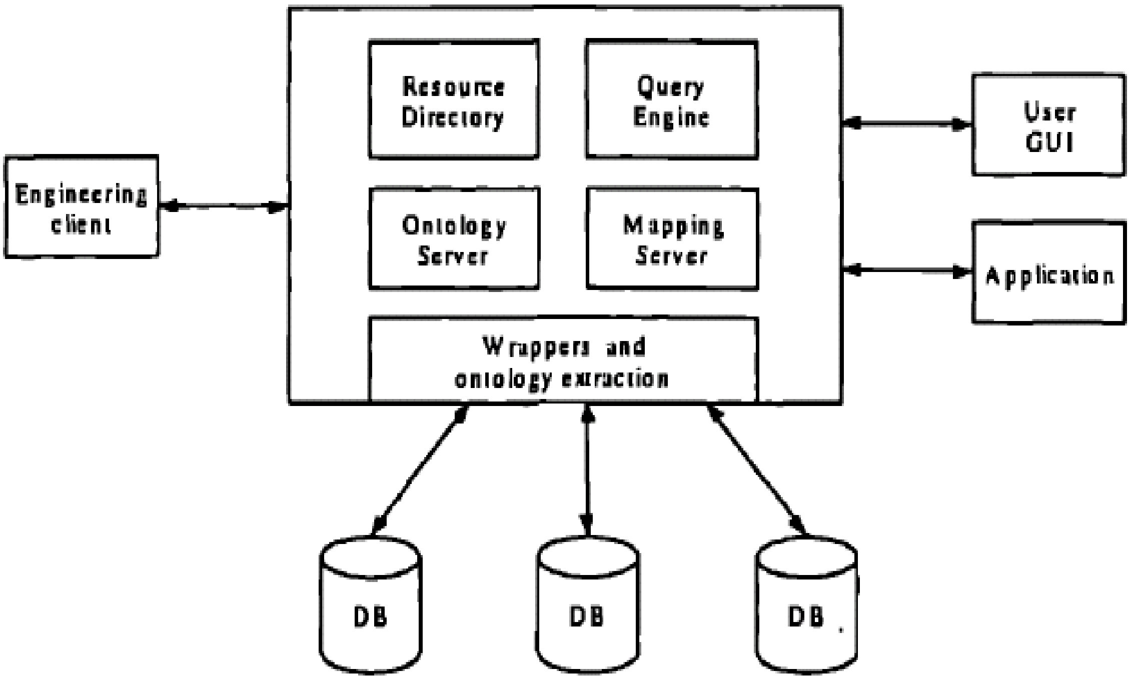

Figure 2-10: The DOME Architecture (Cui, Jones and O'Brien 2002)

The ontology server in this architecture (shown in Figure 2-10) stores the ontologies that are defined using the engineering client and allows access to the three kinds of ontologies in a DOME network: shared, resource and application ontologies. Shared ontologies contain definitions of general terms that are common across and between enterprises. Resource ontology contains definitions of terms used by a particular resource. Application ontology defines the terminology of a user-group or client application.

2.3.5 Ontology with ebXML and RosettaNet

2.3.5.1 Aligning ebXML and Ontology

Hofreiter and Hurmer (2002) propose this approach to align ebXML and Ontology. There are six levels we have to consider when mentioning interoperability in B2B. These levels, ranging from low to high, are Transport Protocol, Message Envelop, Transfer Syntax, E-Business Vocabulary, Document Semantics and Business Process Semantics (shown in Figure 2-11). This research interlinks and coordinates existing technologies that are E-Business vocabularies, Ontologies, Open-EDI, UMM and ebXML to ensure B2B interoperability.

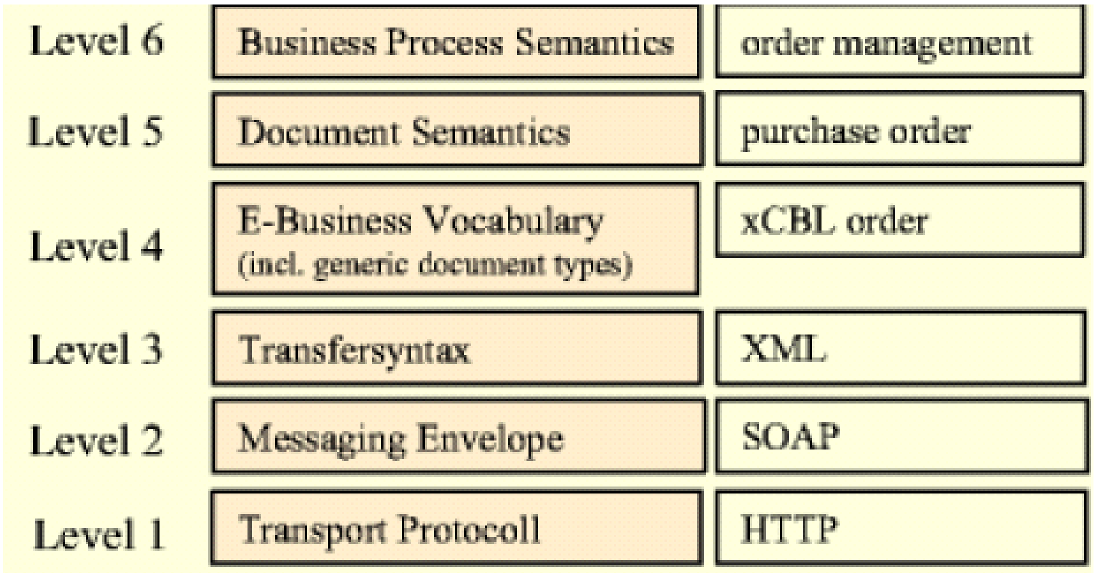

Figure 2-11: Levels of Interoperability in B2B (Hofreiter and Hurmer, 2002)

In this research, the authors present an ebXML core components-based ontology framework. This framework uses an ontology based on ebXML core components expressed in RDF to allow for bridging between different e-business vocabularies.

This framework is based on four major steps. The first step is to develop an ontology derived from the latest version of the ebXML core components specification. The second step (shown in Figure 2-12) is to provide language bindings between the document ontology and the corresponding document type of an e-business vocabulary. The third step requires the definition of a context-specific view into the ontology. The final step represents the language binding and the views and derives an implementation guideline in a certain e-business vocabularies.

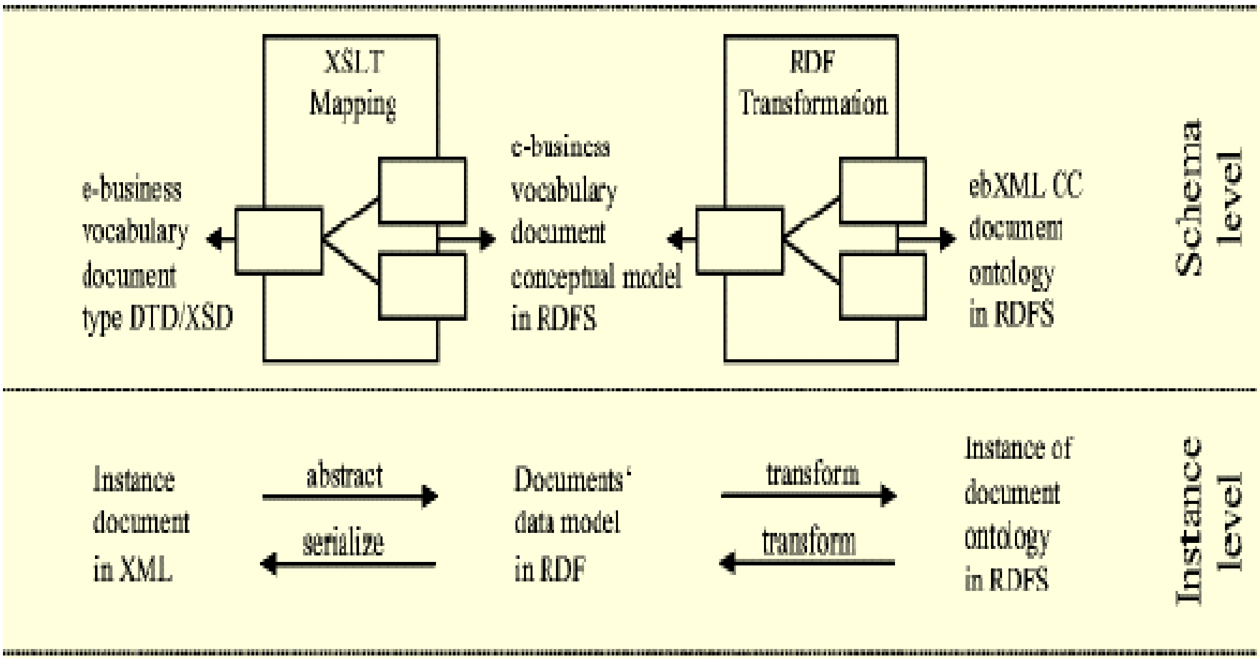

Figure 2-12: Language Binding Between Ontology and E-Business Vocabulary

Most B2B standards specify the syntax of specification. The major feature of ebXML is bridging between different e-business vocabularies or different B2B standards. However the detail mechanism is not specified in ebXML. This research provides an ontology approach in B2B integration based on ebXML standard. Indeed, this approach complements ebXML.

Another standard, RosettaNet, also specifies the syntax of messages, vocabulary and process. But RosettaNet is different from ebXML. RosettaNet does not bridge other B2B standards. RosettaNet only can communicate with itself. Therefore, RosettaNet cannot adopt the four steps, which are proposed here, completely. This approach is more suitable for horizontal standards.

2.3.5.1 Aligning RosettaNet and Ontology

It is not easy to define the business constraints when two companies set up the business processes, and it is not easy to reuse and share the business constraints either. The B2B standards such as RosettaNet are more syntactic rather than semantic. RosettaNet uses XML and XML Schema to define standardized syntax for messages exchanges used in B2B interactions. Semantic constraints on interaction are currently represented in an informal way. RosettaNet uses the DTD describing its syntactic structure of message currently. Even RosettaNet uses the UML class diagram and XML Schema for Next Generation PIP. There are still some problems in the representation of the semantic constraints. Trastour, Preist and Coleman (2003) propose the Nile system to help with the semantic constraints management. This proposed system is able to transform XML Schema into DAML+OIL and transform XML into RDF. In this research, DAML+OIL is used to model:

- The business object class hierarchies and their attributes
- The semantic constraints on business objects coming from the PIP definitions
- The notion of deployment context
- The additional semantic constraints imposed by a business with respect to a deployment context

The Nile system consists of three key technology components. The first component is the XML to schema to DAML+OIL translation tool because DAML+OIL helps to define the semantic constraints. The second component is the Constraint Knowledge Base that contains and defines the constraints in DAML+OIL. The Nile system also provides the Nile Constraint Editor to manipulate the knowledge base. The third component is the XML document Validator. It validates the semantic and syntactic constraints of an XML document.

The Nile tool can be used to commission a new B2B partnership, and to manage an existing one. The Figure 2-13 depicts how the Nile works. The XML Schemas are loaded into the Nile Constraint Editor and automatically translated into DAML+OIL and loaded into the Knowledge Base Editor (step 1). If the PIP specifies additional constraints, these can also be entered in the Knowledge Base (step 2). If a business has already used this PIP with another partner, appropriate information will already appear in their knowledge base, so they can skip this stage. The partners augment the set of constraints with personal constraints, which are imposed by their internal business processes and the specifics of the relationship they are trying to set up (step 3 and 4). The partners can use constraint editor to edit.

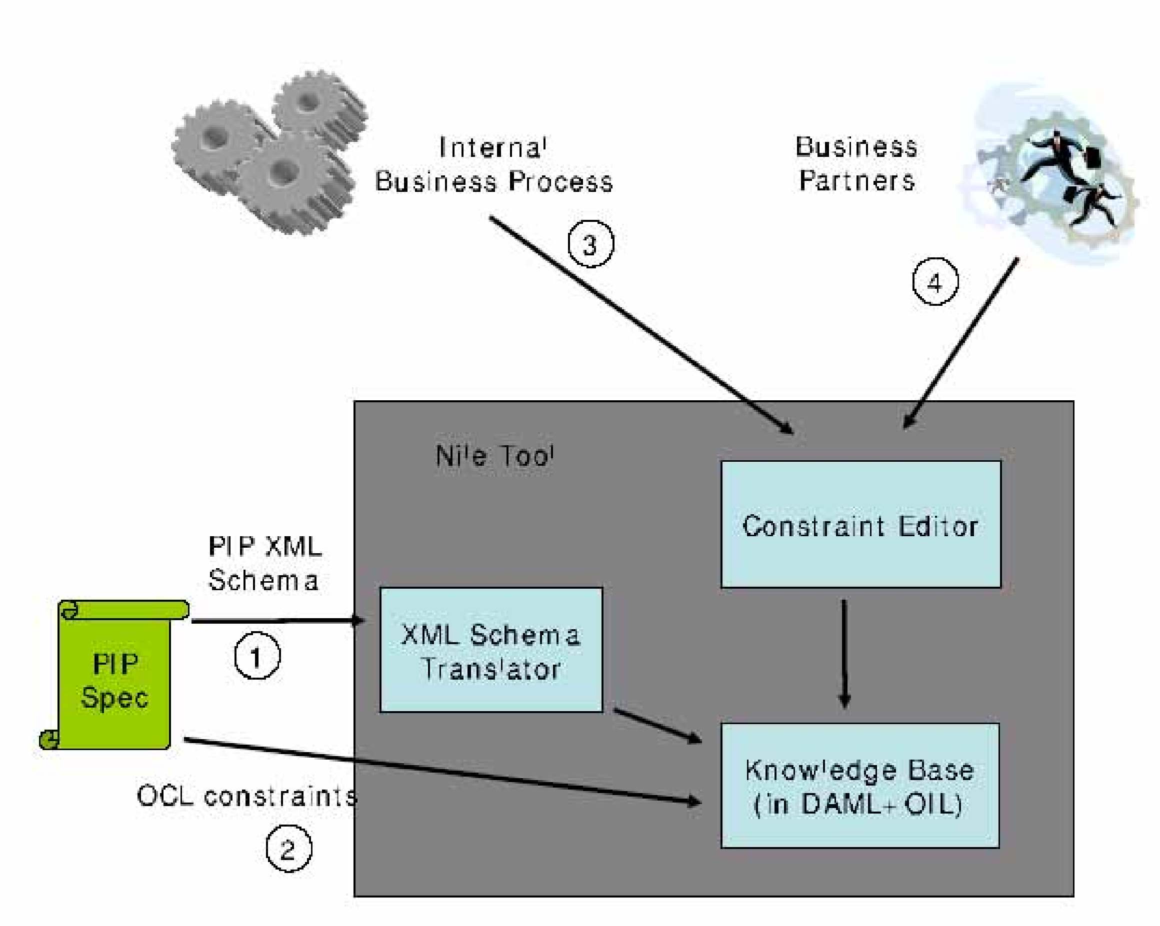

Figure 2-13: Deploying a PIP In a Given Context (Trastour, Preist, and Coleman, 2003)

This research points out another issue in B2B integration. It needs methods and tools to manage the semantic constraints in B2B integration. The authors develop the

Nile system so that it can manage semantic constraints. However, B2B standards do not all use XML Schema and OCL constraints, which are the input of Nile, to specify their messages.

2.3.6 Discussion

We have summarized the dimensions of B2B integration in section 2.1. They are the B2B protocols, the trading partners, the business processes, the enterprise applications, and the business data. We split the business data in to document, catalog and content level from section 2.3.1. We also choose the service to be another dimension because of the linkage between B2B and Web Service is the trend. The constraint is one of dimension, too. Since the business constraints and process constraints are required. We select these dimensions to compare and discuss with the approaches that we mention in section 2.3.

In Table 2-2, we present the components of B2B integration and what researches support ontology to the components.

Table 2-2: The Comparison Between Different Works in B2B and Ontology (This Research).

Dimensions and components of B2Bi	This Research	RDFT (Omelayenko ,2001)	Semantic B2Bi (Bussler, 2001)	SWWS (Bussler et al., 2002)	DAML-S (DAML, 2004)	DOME (Cui, et al., 2002)	ebXML (Hofreiter and Hurmer, 2002)	RosettaNet (Trastour, et al., 2003)
Process	◎	◎	◎	◎	◎			
Partner	◎							
Application						◎		
Document	◎	◎		◎			◎	◎
Catalog		◎		◎				
Content		◎		◎		◎		
Vocabulary	◎					◎	◎	◎
Protocol	◎			◎				
Service				◎	◎			
Constraint	◎							◎
Architecture			◎	◎				
Ontology Language	OWL	RDF RDFS	No specify	RDF OWL	DAML-S	RDF DAML+OIL	RDF RDFS	DAML+OIL

◎ : Support

The RDFT approach provides the ability of mapping between different data models that are modeled by RDF. However, RDFT is applied to the process and the data level currently. If we can find other methods to extract the data models from other dimensions, RDFT still can provide its utility.

Bussler's Semantic B2B Integration tries to introduce the field of B2B integration from a technical viewpoint with the focus on semantic integration aspects. Bussler describes the semantic B2B architecture, but he does not provide details about how this architecture functions.

SWWS is similar to Bussler's Semantic B2B Integration, but SWWS adds on the concept of Web Service. SWWS will support the Semantic transformation on different levels: Data, Business Logics, Message Exchange Protocols and Dynamic Service Invocation. Again, SWWS is still a conceptual architecture, too. However, the concept of semantic integration is clearer than the Semantic B2B Integration architecture.

DAML-S is a service description language and an ontology. DAML-S can describe the semantics of the web services and the processes. DAML-S will be the key technology enabled semantic web service especially in e-Commerce.

DOME focuses on application level to contrast with other research. DOME is helpful for integrating applications and managing the different data source and adapters. Other research that we mention here does not involve the application level.

Research of alignment with ebXML and Ontology provides an approach to align different vocabulary. This approach is helpful to the ebXML standard. ebXML provides the syntactic exchange and platform between different protocols, but it does not provide the semantic exchange.

Research of alignment with RosettaNet and Ontology gives us another aspect of B2B integration. This research uses DAML+OIL to model the document ontology and define the business constraints that is also important to B2B integration.

Table 2-3: The Discussion Between Different B2B Research (This Research).

Research	Feature	Defect
This Research	● It provides a common way to model different B2B standards. ● B2B standards will have the similar content model and process model. ● This research proposes the process and document should consider together.	● It only considers DTD, not XML Schema ● It does not have inference and query model.
RDFT (Omelayenko ,2001)	● It proposed the multi-layered model for content integration. ● It provides the ability of mapping between different data models for e-Marketplace.	● It does not provide the method to model different data models.
Semantic B2Bi (Bussler, 2001)	● It proposes the architecture of semantic B2B integration.	● It does not mention how each component work.
SWWS (Bussler et al., 2002)	● It adds on the concept of Web Service. ● It provides a platform for the semantic B2B integration. ● SWWS will support the Semantic transformation on different levels.	● The language for describing semantic of web service is not a standard language.
OWL-S (DAML, 2004)	● It describes the semantics of the web services and the processes.	● It does not provide the ability to describe the semantic of content.
DOME (Cui, et al., 2002)	● It integrates applications and manages the different data source and adapters. ● It builds a "one stop knowledge shop" for corporate information.	● It does not provide the ability to describe the business processes.
ebXML and Ontology (Hofreiter and Hurmer, 2002)	● It is an approach for ebXML to align different vocabulary.	● The method to build the vocabulary ontology is complex and not automatically.
RosettaNet and Ontology (Trastour et al.,2003)	● It provides the Constraint Knowledge Base. ● It models the document ontology and defines the business constraints.	● It only supports the RosettaNet.

We summarize this research and present a conclusion. We need a powerful and a semantic supported B2B integration platform to support the dimensions more fully. SWWS is an ideal architecture for this purpose. However, SWWS needs to improve by supporting more dimensions, which the SWWS ontology server needs to support more functions. The required technologies include the ontology development environment, ontology development methods, ontology query, inference engines, ontology management, and ontology mapping methods.

To summarize the open issues of the semantic B2B integration:

1. We need a flexible and scalable B2B platform to support the semantic requirements.
2. The selection of ontology languages is also important. RDF/RDFS and OWL have their own advantages. Selecting a suitable ontology language is an issue.
3. We need a common approach to build up our ontologies for different B2B standards.
4. We need mapping methods between the ontologies at each B2B level.
5. The required technologies of ontology server are researched individually. We need to integrate and standardize the required technologies.

We have discussed the ontology approaches to align with ebXML and RosettaNet. However, the focuses of each alignment are different. Our research will focus on the third issue above to develop a common approach. Such the ontologies of different B2B standards can share to each other easier, because they are built in a common way and they have the similar data model and process model.