Proc. of Concurrent Engineering (CE'2003), Special track Data Integration in Engineering, Madeira, Portugal, 26-30/07/2003, 268-278

# PLIB Ontology For B2B Electronic Commerce

Y. AKLOUF , G. PIERRA & Y. AIT AMEUR

*Laboratory of Applied Computer Science*
*National Engineering School for Mechanics and Aerotechnics*
*LISI/ENSMA - Site du Futuroscope - B.P. 109 - 86960 FUTUROSCOPE Cedex - FRANCE*
*Tel : (+33) (0)5-49-49-80-60 - Fax : (+33) (0)5-49-49-80-64*
*Email : {Aklouf, Pierra,Yamine}@ensma.fr*

H. DRIAS
*Electronics and Computer Science Faculty -*
*Research Laboratory in Artificial Intelligence, LRIA-USTHB*
*LP 32, El ALIA, 16111 Algiers, ALGERIE*
*Tel : (+213) 21247892 - Fax: (+213) 21247892*
*Email: Drias@wissal.dz*

ABSTRACT: Currently computers are changing from single isolated devices to entry points into a world wide network of information exchange and business transactions called the World Wide Web (WWW). Therefore support in the exchange of data, information, and knowledge is becoming the key issue in current computer technology. Ontologies provide a shared and common understanding of a domain that can be communicated between partners (with are two company for example) and application systems. Therefore, they may play a major role in supporting information exchange processes in various areas especially in B2B ecommerce. This paper discusses the role ontologies will play in electronic commerce and the need for having a common ontology for defining products and components involved in business processes and data exchange. In addition, we show that business protocols and product ontologies may be orthogonal.
KEYWORDS: Business To Business, Ecommerce, PLIB, Dictionary, Ontology, Business protocols.

## 1 INTRODUCTION

Electronic Commerce in the Business To Business field (B2B) is not a new phenomena. Initiatives to support electronic data exchange in business processes between different companies existed already in the sixties. In order to exchange business transactions sender and receiver have to agree on a common standard (a protocol for transmitting the content and a language for describing the content) A number of standards arose for this purpose. One of them is the UN initiative *Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT)* (United Nation.1999).

In general, automatization of business transactions has not lived up to the expectations of its propagandists. This can be explained by some serious shortcomings of existing approaches like EDIFACT: It is a rather procedural and cumbersome standard, making the programming of business transactions expensive, error prone and hard to maintain. Finally, the exchange of business data via extranets is not integrated with other document exchange processes, i.e., EDIFACT is an isolated standard. Using the infrastructure of the Internet for business exchange will significantly improve this situation. Standard browsers can be used to render business transactions and these transactions are transparently integrated into other document exchange processes in intranet and Internet environments. The first portals for electronic commerce using Internet facilities are *harbinger.net*, *mysap.com* and *VerticalNet.com*. However, this is currently hampered by the fact that HTML does not provide means for presenting rich syntax and semantics of data. XML, which is designed to close this gap in current Internet technology, should therefore change the situation (Glushko et al. 1999). B2B communication and data exchange can then be modelled with the same means that are available for the other data exchange processes, transaction content can easily be rendered by standard browsers, maintenance should become cheaper (see WebEDI Westarp et al. 1999 and XML/EDI, Peat & Webber. (1997).

XML will provide a standard serialized syntax for defining the structure of data. Still, it does not provide standard data structures and terminologies to describe business processes and exchanged products. Therefore, ontologies will have to play a crucial role in XML-based electronic commerce.

Standard ontologies have to be developed for covering the various business areas and product domains. In addition to official standards, on-line marketplaces (Internet portals) may generate de facto standards. If they can attract significant shares of the on-line transactions in a business field they will factually create a standard ontology for this area. In the scope of this paper we propose to use PLIB-based ontologies (ISO 13584-42, 1998).

Then, ontology-based trading will significantly extend the degree to which data exchange is automated and will create complete new business models in the participating Market segments (McGuinness. 1999).

Comparing electronic commerce in the B2C and B2B one has to admit that B2C is more mature. However, the B2B area will be perspectively more interesting as around 80% of the transaction volume will be in the B2B area.

In the first part of this paper we present the context of our study. In the second part, we show a layered architecture of B2B ecommerce and its various modules, in the third part we specify concept of ontology used in PLIB model, and its integration in the proposed approach. Finally in the last part, we show an application of the use of the PLIB ontology (dictionary) in the RosettaNet business protocol called PIP2A9.

## 2 A CONCEPTUAL MODEL FOR B2B ECOMMERCE

B2B Automation Standardization is about interoperability of business content and message exchange between business systems of different enterprises, as well as the process automation associated with them. It requires many decisions at different levels. Without careful architectural thinking and planning, it is impossible to make right decisions that cover all the bases.

Because of the level of complexity, it is unrealistic to have any single group or standard body to make all the decisions. It requires collaboration of both vertical and horizontal standardisation bodies and industry experts to work together to drive B2B standards definition and convergence (among the most known standards we find RosettaNet and ebXML). However, there is a need for a common view of what the basic components of B2B solutions should be and what are the common definitions of terminologies for these architectural elements.

Therefore, we need to have a conceptual model (Heather. 2001) that states the high level architectural elements, without getting into details of architectural definitions. Such a model will remain relatively stable over time as technologies, standards and implementation details evolve.

## 3 A LAYERED ARCHITECTURE FOR B2B ECOMMERCE

As we stated in the previous sections, B2B solutions are very complex, involving business processes, business contents, and enabling technologies. By using a layered architecture and having the lower layers support and enable the upper layers, it is possible to divide a very complex problem into several less complex, more manageable sub-problems – a classic divide and conquer approach. Another advantage of the layered architecture is to allow different groups (standards bodies) to work on different layers at the same time, while remaining connected, which will shorten the time needed to solve the overall problem.

In the conceptual model (Heather. 2001) (BIC. 2001) described herein, the different elements of the B2B architecture can be represented as layers where one is built on top of the other; each layer supporting all of those above it. It makes sense then that the lower is the layer the bigger is the effect of deviation and duplication, therefore, the bigger the benefit for convergence. It is also interesting to note that since the lower layers are more technical and support more horizontal functionality, it is easier to seek commonalities in terms of used basic technologies. Proceeding this way leads to greater opportunities for convergence. It is conceivable that we should drive standard convergence from the bottom up, i.e. from the Network Transport and Messaging layers up to more sophisticated business content and business process description layers.

We also realized that we may not be able to achieve convergence at all layers, especially the business content, business process, and backend integration layers. However, a broad agreement on convergence at the lower, enabling layers will make diversity on the top layers more effective and manageable. It also offers the potential of reuse and interoperability among the different business content and processes as we drive convergence to higher levels in the conceptual model.

The convergence model of RosettaNet (RosettaNet. 2001a) is a document defining how the multiple initiatives (SOAP, ebXML, RosettaNet…) are complementary with an aim of setting up a solution of B2B integration in a "Supply Chain" problems. It is an interesting initiative because it is the first time that an actor explains the complementarity of XML standards. We will expose below the elements of this model which are valid in horizontal problems, i.e. independent of the trade of the company. For RosettaNet, a solution B2B contains the following layers in addition to the layers specific to the trade association of the company:

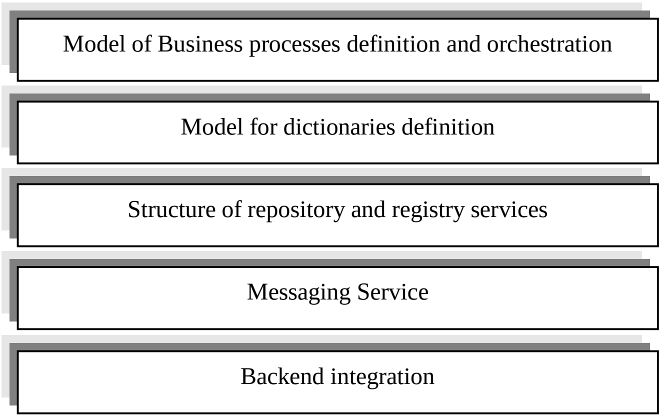



Figure 1: The layered architecture of B2B Ecommerce.

*3.1 Backend Integration*

This layer provides hooks into the backend enterprise systems through API or shared messaging bus. It includes functions like business logic processing and format transformation.

This is the gateway to the backend ERP systems. It is developed using tools from the Service-Oriented Architecture layer and communicates with upper layers through the Messaging Service layers. Due to the proprietary nature of ERP systems, there are fewer opportunities for standardization in this layer. However, XML provides a vehicle to have common adapters for popular ERP systems. As part of the private processes development, B2B system integrators need to work with ERP vendors to build seamless connections. From the B2B standards development point of view, it may be out of the scope. However, this layer is vital for developing end-to-end B2B solutions and is very critical to customers who want to implement their own solutions. One strategy is to work with ERP vendors like SAP, PeopleSoft, etc., so that they will adopt the principle of this conceptual model and develop backend integration solutions that match the conceptual model and are suitable for corresponding ERP customers (Gartner. 2000).

*3.2 Messaging Service*

This layer defines a standardized message and envelope structure and layout definitions, which have specific technical purposes. It addresses the need to record session and communication settings for message transport in order to enable coordination between parties in a business transaction, including parameters that control Reliable Messaging, Secured Messaging, etc.

This layer is the foundation of communications amongst all the layers. It provides the lower-level message exchange support for the Service Description Language, Repository/Registry, Process Description Language layers. It also provides a base for Business Content Format Definitions layer. in this layer the most used specification are RosettaNet. RNIF1.1 (RosettaNet 2001b), SOAP, ebXML TRP(ebXML 2001a).

*3.3 Structure of repository and registry services*

*Repository.*

It represents the standardized repository services that specify the structure and access protocol and schemas for business content storage and retrieval, which includes the term, its constraints, its representations, etc.

This part of this layer provides standard-based services for storage and retrieval of entries at the Registry Services. It will provide a platform independent way to store and retrieve business content format definition schema and business process description. We can use in this case the RosettaNet Dictionary Repository or ebXML Reg/Rep.

Note that there have not been standards defined for this layer. Repository is typically defined on ad hoc based and closely tied to database technologies used for implementation. The advantage of having a platform neutral repository standard for business content and business processes is that the higher layers of business content will have a standard way to store and retrieve business content and business processes once they are defined.

*Registry Services.*

It specifies the structure and access protocol of registries and repositories that trading entities can access to discover each other’s capabilities and services. It covers naming, directory, registry, privacy, authorization and identification services.

The registry in this layer is used to publish and register business processes and services. Business processes that need to dynamically explore and discover available services or that publish services for other businesses to use will make use of the services specified in this layer. The Registry Services could be used to publish and discover both business content and business processes. The Registry Service keeps a list of the entries of entities and stores the objects in the Repository (Randy. 2001).

### *3.4 Dictionary layer*

It is not sufficient to have a standardised method of packing and shipping a message. Participants must also share a common meaning of the message contents within the context of a particular business domain. This is accomplished via dictionaries.

The dictionary is a set of concepts and standard definitions, for a particular domain, shared and used by two partners during an exchange.

For example consider RosettaNet which defines two types of dictionaries :

1- The Business Dictionary which is a repository for describing a semantics for objects (XML elements, attributes, and entities) using a message DTD to tag the individual components of message contents according to its type of information. When a message is regarded as a view of a relational database, then the Business Dictionary defines the column names.

2- The RNTD (RosettaNet Technical Dictionary ) Dictionary which is a repository of semantics for the content of each element or attribute. The Dictionary defines the meaning of the contents of each cell. These definitions standardise a shared understanding for the complete domain vocabulary. Consequently, none of the semantics need to be transmitted in each message. This enables sender and recipient to understand the content in the same way.

### *3.5 Model of Business processes definition and orchestration*

This layer specifies business processes that are applicable to a broad range of businesses, regardless of the vertical industry or locale within which the business operates or of the specific characteristics of the business. These processes cover many domains of activity that businesses engage in, such as collaborative product development, request for quote, supply chain execution, purchasing, and manufacturing.

Business Process is the business rules, the definition of the roles of the parties involved, and the trigger events that provide the context for the exchange of information. Process definitions should cover the complete set of business events required to accomplish a business objective (e.g., placing an order would include steps such as sourcing, issuing a purchase order, receiving acknowledgments and dealing with changes) rather than just discrete steps (e.g., issuing a purchase order).

This layer uses tools provided by the Process Description (Aberdeen 2001) to describe the business process sequencing and choreography amongst processes that are Universal to all businesses or business domains.

Examples: Invoicing process, purchasing process, Base level Purchase Order. For this layer the use of RosettaNet PIPs (Partner Interface Process) and ebXML may be recommended.

This layer also defined the format of the business contents, which is Explained in the following point

#### *Business Content Format Definition*

Business Content includes everything that composes the payload of business transactions, which dictionary entries, composition of dictionary entries, special business documents, and attachments. Business Content Format Definition is the specification of the data structures, data types, constraints and code lists of all the items necessary to compose valid business content.

This layer specifies the structure and semantics for particular business processes. It is built on top of the Core XML Format Standards and with knowledge of particular business processes required for business transactions on the right side (business process side) of the model. This layer also takes into consideration the schema required to store and retrieve content formation definition based on the services provided from the dictionary and repository layer.

## 4 THE PROPOSED APPROACH

The various models and standards of B2B electronic commerce use an architecture based on different layers presented in the model suggested previously, with some specific characteristics that each domain of application has. The two most significant layers in this model are the layer which defines the business processes and the layer which defined the semantic of dictionary contents i.e. products, their characteristic and parameters. For the first point, we propose to use the business process defined by the following existing standards:

**ebXML**(ebXML. 2001a): provides an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties. The different components of ebXML architecture collaborate between them to:

- ✓ provide a view for integration of business processes among ad-hoc or established independent business partners by electronic means.
- ✓ reduce the need for collaborative business partners to have individual and expensive prior agreement on how to integrate business processes.
- ✓ provide a high-level business-centric view of distributed e-business processes.
- ✓ specify the roles, interactions, and interfaces among the various ebXML specification components such as the business process metamodel; core components; registry and repository; and message transport, routing, and packaging.
- ✓ allow for both business processes and enabling technologies to evolve independently while retaining long-term investments in both.
- ✓ integrate with new and legacy systems throughout the enterprise.
- ✓ leverage existing technologies and standards.
- ✓ co-ordinate with BP process specification and core components identification, provide for naming conventions for technical and business content in the technical architecture.
- ✓ provide design guidelines for ebXML compliant messages.

**RosettaNet**(RosettaNet 2001b)**:** RosettaNet is a non-profit consortium working to create, implement and promote open e-business process standards. The RosettaNet standard allow to automate relatively common business processes between trading partners across almost any supply chain or business model. It is generally accepted that an XML business process standard should include both a content (payload) and choreography (dialogue) component. RosettaNet seeks to increase the speed of B2B business process integration by:

- ✓ reducing confusion through a conceptual model that puts individual XML component standards into context.
- ✓ evaluating existing and new XML initiatives against the conceptual model.
- ✓ embracing horizontal XML initiatives as they become proven and universally accepted.
- ✓ maintaining a leadership role in the development of XML business process standards required to meet the specific needs of the high technology industry.

With the various processes defined by these two standards we use the PLIB model in the layer which defines the dictionary metamodel. PLIB, the Parts Library standardisation initiative, was launched at the ISO level in 1990. Its goal is to develop a computer-interpretable representation of parts library data to enable a full digital information exchange between component suppliers and users.

PLIB ontologies allow the description of classes, properties, domains of values, instances of objects and information sources. A class is a collection of objects defined in intention. A property is a binary relation between two classes or a class and a domain of values (the term "attribute" is used for the meta-descriptors of classes and properties). A domain of values is a mathematical set defined in extension or in intention. An instance represents an object pertaining to a class. Lastly, any ontological definition emanates from a certain source which assumes the responsibility for it.

A PLIB ontology is modelled according to the PLIB (meta) dictionary model. This model (Pierra. 1994) (ISO13584-42. 1998) is based on the object oriented paradigm (Meyer. 1998) (Coad & Yourdon. (1992): components are gathered in parts families that are represented by classes. This set of classes is organised according to a simple hierarchy (on which factorisation/inheritance applies) of classes. Such classes are then precisely described (textually, with technical drawings, and formally by class relationships). Finally each class is associated with a set of technical properties, also described both textually and formally (domain of values, possible measurement unit,...). A basic idea of the definition of a PLIB dictionary is that properties and classes shall be defined simultaneously: applicable properties allow to precisely define the meaning of a class, and conversely, a class determines the context in which a property is unambiguously defined.

The modelling formalism used in PLIB is the EXPRESS language. Developed within the framework of STEP project (Pierra. 2000) the objective of EXPRESS (ISO10303-11. 1994) is the description of models of information for data exchange representing this information in a reliable and non-ambiguous way (Schenck & Wilson. (1994)). With the growth of the Internet, of e-commerce and of B2B applications, several levels of granularity of information exchange between components users and components manufacturers may be identified (one instance, a classs, a whole catalogue). All these levels may be modeled using

PLIB. Moreover, XML emerged as the most popular format for exchanging over the Internet. Thus, it became desirable to allow PLIB information to be exchanged using XML.

The PLIB standard defines classes and properties by means of a meta-model that consists of entities expressed in EXPRESS. We can access and retrieve this information in a portable way through the standardised interface SDAI (Standard Data Access Interface) (ISO 10303-22. 1997). SDAI is based on the definition of a library of standardised access functions (an API).

This interface allows in particular:

- ✓ accessing to and handling automatically entities instances defined in EXPRESS;
- ✓ simultaneous access to several data bases by several applications;
- ✓ access to EXPRESS definitions of the elements of data which can be handled by an application;
- ✓ checking the constraints defined in the EXPRESS data model.

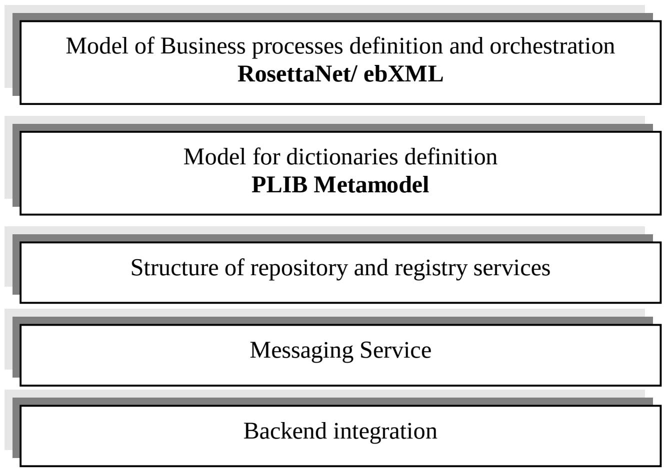



Figure 2: The layered architecture with PLIB metamodel.

If we take as example, an exchange between two partners, the diagram corresponding to the above layered structure would have the following exchange form.

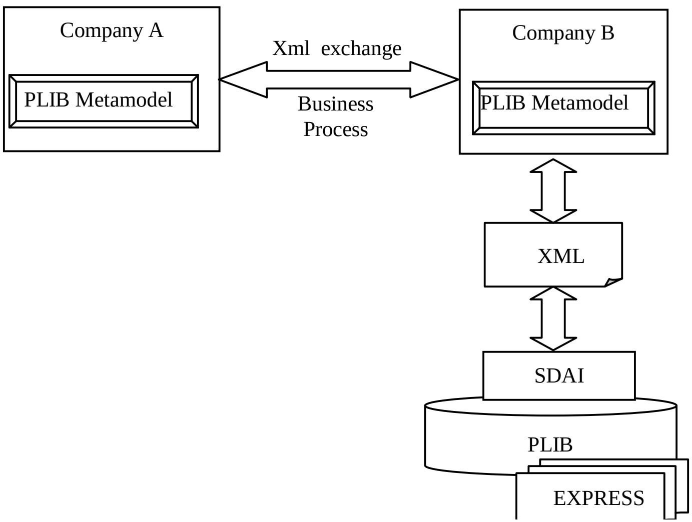



Figure 3: Example of PLIB exchange between two partners.

## 5 AN ONTOLOGY FOR B2B ECOMMERCE

The different sections described above showed the potential use in a separate way of data dictionaries and business models. What follows describe the need for having a common ontology to make exchanges in electronic commerce.

Normally, the design of an electronic commerce system starts with the development of a business model. In most cases, such a business model is written down in natural language, perhaps with some informal sketches. The concepts and their interpretations used to describe a business model vary across different stakeholders, and this leads to important obstacles to achieve business-IT alignment in e-commerce applications. Given the enabling role of IT in electronic commerce, this alignment problem is no longer just an engineering issue: it has a strategic significance.

During the design of a business model, ontology is therefore useful to *prescribe* which concepts and relations have to be present in a business model. An ontology should provide a reusable conceptualisation, in this case of the concept of e-business model, on which people can agree. By specializing and instantiating concepts and relations of the ontology for a particular case, the ontology can also be used to describe a particular business model in a precise and structured way. In the present context, we are mainly interested in ways to enhance communication between various stakeholders, that is, in shared meaning rather than automated reasoning. Thus, our current goal is to construct a so-called "lightweight" ontology ( Coad & Yourdon. 1992).

Furthermore, a business model ontology shows to the designers what kind of decisions should be taken during business model development. If stakeholders agree on a particular business model, a number of business decisions have been taken, so that the model serves as a precise set of business requirements for the electronic commerce information system. These requirements are useful for software architects who design the electronic commerce system from a technical point of view.

An ontology for e-business models (Jasper & Uschold. 1999) must be capable of representing a range of business issues. These issues center, around the generic concept of *value*, and how to create and exchange it in a network setting. This is our key proposal. Informally, a business model highlights a network of actors and how they create or consume objects of value. These actors can be private persons, companies or enterprise alliances. Furthermore, a business model represents the services offered by and requested from actors. It should be capable to represent if an actor is willing to exchange an object of value (e.g., the right to listen to a music track) for another object of value.

## 6 EXAMPLE

We finish our study by an example in which we will describe the use of a business protocols of RosettaNet called PIP2A9 that we will describe after and how to use a PLIB database with this last. We describe first the PIP and then the PIP2A9.

### *6.1. PIP*

The Partner Interface Process (ebXML. 2001b) is a specialized sub-process activity. The contents of a PIP are a set of transactions that are sent back and forth between two companies. These transactions occur between two computers within the companies in support of transferring data required for specific tasks that occur in both companies. Thus, a PIP will be displayed in a Process as being connected between two standard sub-processes. The first sub-process is performed by one of the Partner companies and the second sub-process is performed by another Partner company. The scope of a PIP is far from arbitrary. The RosettaNet development strategy begins by partitioning distribution business processes into categories termed clusters. These clusters are in turn partitioned into finer grained categories termed segments. Analyses of the business processes in a segment yield a number of RosettaNet opportunity areas whose definition provides the scope of a PIP.

### *6.2. PIP2A9*

In our case, we study the PIP2A9 (RosettaNet. 2001c) which is named "Query Technical Product Information". Technical product information is that category of information that describes the behavioural, electrical, physical, and other characteristics of products. There are numerous classes of customers within the supply chain that need to be able to access product technical information. These include distributors, information providers (such as web portal companies, other commercial information aggregators, and end-customer information system owners), engineering, design engineering, and manufacturing and test engineering.

Depending on the phase of the design cycle, different subsets of the overall set of characteristic information are required, as may be the manner in which that information is requested. Requirements for this PIP span across many phases of the information life cycle. For example, the design of Printed Circuit Assemblies (PCA) by Original Equipment Manufacturers (OEM) includes the following lifecycle phases:

- **Find** – OEM companies identify product(s) that match essential needs without knowledge of product identifier, and possibly by search across many potential ProductInformationDistributors (Distributors).
- **Try** – OEM companies evaluate potential selections of information (such as datasheets, models, test benches, packaging options, etc.) for fit to design problem.
- **Buy** – Pricing, availability, risk assessment information, lifecycle data, and sample parts availability are important in this phase.
- **Design** – OEM companies perform design using the selected product. For this phase, EDA libraries and various models are of particular importance.
- **Build** – OEM companies actually build and test the printed circuit assembly and thus, require test patterns and setup conditions, physical information to pick and place the product, and alternate source information, etc.

This diagram presents the different steps of this PIP:

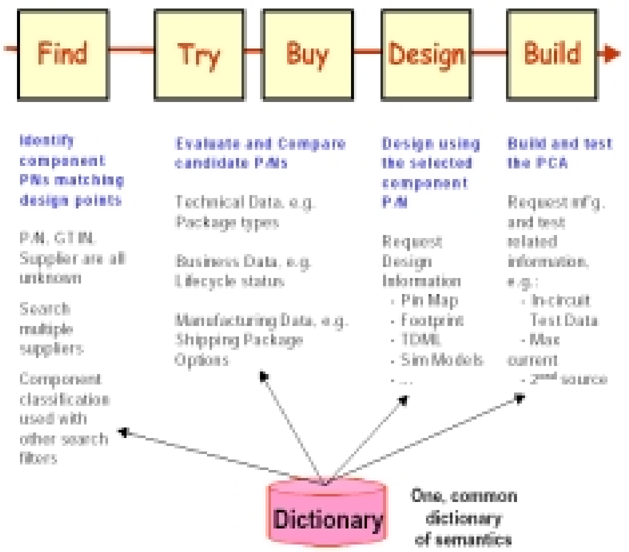



Figure 4: Various phases of PIP2A9.

*6.3 The PLIB dictionary*

In our case, the dictionary will be PLIB-based. PLIB (ISO13584-42. 1998) comprises a universal method of identification (GUI: "Global Unit identifier"). Each potential source is associated with one unique identifier (in general pre-existent for any organization or company, for example in France it is built on SIRET or SIREN codes). Each source must then assign a single code to each class which it defines. Finally, the code of a property must be unique for a class and all its subclasses. The concatenation of these codes then makes it possible to identify in a single and universal way, each concept above. It is this simple code, called a BSU (Basic Semantic Unit), that will be sufficient to refer unambiguously to a class or a property.

The basic idea that we want to show in this paper, is that the business process are independent of the definitions of the dictionaries. Thus, in our example we will show that the dictionary used in this business process (PIP2A9) is not necessarily the one defined by RosettaNet. We claim that multiple different dictionaries (or ontologies) can be used within any business protocol, and for example, within the protocols defined by RosettaNet. The result of such an approach is a modularisation of the use of business protocols and of dictionaries. It shows that both business protocols and dictionaries may be considered as orthogonal. In order to validate our claim, we show that it is possible to use the PLIB dictionaries.

*6.4. Application of the described approach to the example*

There is a broad set of product information this PIP must support. This information can be organized into the following categories:

1. Characteristics – physical, electrical, behavioral, or other characteristics of the product that are described as simple name/value pairs (such as supply voltage, rise delay, package type, etc.) Each of these Characteristics is specified in the RosettaNet Technical Dictionary (Dictionary)
2. CharacteristicSets – multiple characteristics that can be requested with a single name (e.g. RiskAssessment) with semantics defined in the Dictionary. Standardized sets of characteristics are defined by PropertyDefinitionSets in the Dictionary, however, a Distributor may include definitions of its own Sets referenced in a particular message.
3. ProductInformationObjects – standalone files that describe product information as complex structures. Metadata for PIOs are also defined in the Dictionary. A PIO may have any content (defined by its specification), including binary. Some PIOs may also be XML with their own DTDs. Typical product information objects include e. g., data sheets, instruction booklets.

Technical product information requests are made by queries in which the requestor may specify a set of characteristics, each with a value, or value range, that must be matched on products for which a response of the requested information will be returned. All query characteristics containing a value should be used in the product search. Responses for the matching product(s) will include Characteristics and Product InformationObjects of all types requested in the query (whether or not they contained values in the query or not) (RosettaNet. 2001d).

For each phase of this process, partner A sends a request to B. And B answers with the data that it has. The request has the following form:

Query
    RosettaNet Class="**XJA644 OR XJA645**"
    Voltage="**5**"
    Rated Maximum Power="**GT 4 AND LT 6**"
    Pin Count="**24**"
    Operating Temperature
    Technology

And the answer will be in form

    Distributor 1:

    Response

PartInfo
    RosettaNet Class="**XJA644** "
    Part Number="**SN74LV652**"
    Voltage[**nom**]="**5**"
    Operating Temperature="**50**"
    Technology[**notAvailable**]=""
    Is Generic="**1**"
    Rated Maximum Power="**5**"
    Pin Count="**24**"
PartInfo
    RosettaNet Class="**XJA645**"
    Part Number="**SN74LV654**"
    Voltage[**nom**]="**5**"
    Operating Temperature="**45**"
    Technology[**notAvailable**]=""
    Rated Maximum Power="**5.5**"
    Is Generic="**1**"
    Pin Count="**24**"

We have given an example of use of PIP2A9 process with RosettaNet dictionary. Now we present another example of use of this process, but with a PLIB dictionary for the fasteners (ISO CD 13584-511) defined by the Chinese National Institute for Standardisation. All the definitions used in the following example come from part this standard PLIB dictionary. .

Query
PLIB Class="**P511CAA168**" // machine screw
shank shape = "Self- drilling screw shank"
nominal diameter of thread ="5"
nominal length="18"
mechanical property class
shape of end

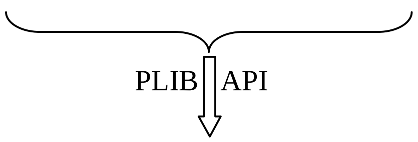
The answer will be as follows
Response
PartInfo
PLIB Class="**P511CAA168**"
shank shape = "Self- drilling screw shank"
nominal diameter of thread ="5"
nominal length="18"
mechanical proprety class="A"
shape of end="Truncated cone point"

Such an automatic answer requires the possibility to query PLIB supported dictionaries and catalogues. Such a possibility exists (supported by an API) and is discussed later in this paper.

The dictionary information used in these exchanges is defined as a set of PLIB data. Thus it may be exchanged by means of EXPRESS files (an XML version of which also exists). Such EXPRESS files may for instance be send to each partner once and for all. To read these EXPRESS files in a PLIB database, we used a set of method written in Java arranged in API called PLIB API. This API handles instances in a PLIB base .Then transactions like the one above may take place.

PLIB API has a wide range of functions and mechanisms that allows to select components in multiple ways. Thus each request sent by a partner who plays a role in the PIP2A9 will be transformed in the local database of the other partner in a set of methods which recovers corresponding information as requested initially.

PLIB Class="**P511CAA168**"
shank shape = "Self- drilling screw shank"
nominal diameter of thread ="5"
nominal length="18"
mechanical proprety class="A"
shape of end="Truncated cone point"




PLIB API is a set of java classes containing PLIB API methods. Each class defines all necessary methods for entity handling (retriving, searching, comparing, etc.).

Several classes are used in the previous example, among them:

PLib_Api_Class_BSU
PLib_Api_Basic_Semantic_Unit
PLib_Api_Class_And_Property_Elements
PLib_Api_Class_BSU_Related_Content
Plib_Api_Database
PLib_Api_Dictionary
....…….

As example *PLib_Api_Class_BSU* contains the different methods for handling the BSU code which is a unique identifier of a class or property. *PLib_Api_Class_BSU_Related_Content* contains all the methods for handling part of class content identified by a BSU code.

Each class used comprises set of methods for handling instances, We have for example:
*get_value_dic_identifier*
*get_value_associated_item*
*get_value_content*
*get_value_attribute*
*get_value_properties*
............
............
As example, *get_value_attribute*, takes an attribute code, its belonging class code and returns the value of this attribute from the database.
*get_value_dic_identifier* read the value of *Class_BSU attribut* from the PLIB dictionary.
*get_value_content* read the value content referred by the *Class_BSU_Related_Content* attribute from the database.

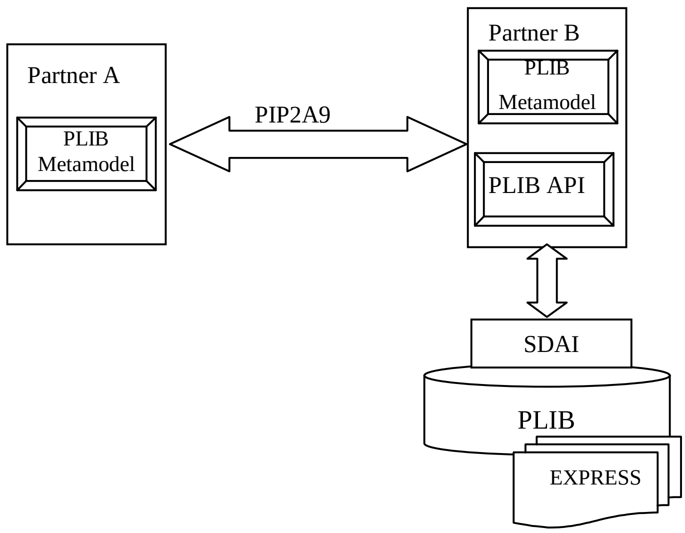



Figure 5: Integration of a PLIB database with PIP2A9.

## 7 CONCLUSION

Ebusiness applications are adopting standards and initiatives for allowing interoperation and interchange of information between information systems. Ontologies aim to provide a shared machine-readable view of domain knowledge, allowing information sharing for heterogeneous systems. In this paper, we have shown how different standard for B2B ecommerce (RosettaNet and ebXML) with different dictionaries (due to a layered architecture for the ecommerce) based on the PLIB dictionary model can be used jointly. This approach allows an integrated and multi-layered architecture. The use of the PLIB dictionary model provides us with a uniform representation of ontologies that that may be exchanged and integrated to constitute a common shared ontology.

We have shown the need for having such a common ontology for exchange between partners, as well as a set of standard business protocols which will be integrated in a layered architecture in order to allow independence and orthogonality between the various layers. Finally, we have finished our study by an example of use of a RosettaNet business protocol, called PIP2A9, that is used for querying technical product information and shown is implementation on a PLIB database.

Currently, RosettaNet defines at the same time, several business protocols and its own dictionary. In this approach, these two kinds of elements are considered as necessarily dependent. In this paper we have shown that the business protocols and the dictionaries are in fact orthogonal.

Moreover, since we do not have a standard which defines a dictionary model in e-business standards, we propose the use of the standardized PLIB model which is completely independent of any business protocol. This approach was illustrated through the developed example.

Another additional and significant advantage, is that if the data are stored in an EXPRESS "repository", we have a standardised interface to build the layer "backend integration", the SDAI, which makes it possible to carry out a portable implementation.

In the future, we plan to continue the study of the integration of the PLIB model in the RosettaNet and ebXML business protocols, and to define a standardised layer of integration between the business protocols for RosettaNet and ebXML on the one hand and PLIB model on the other hand.

## REFERENCES


Aberdeen Group, 2001. The Fast Track to Positive e-Business ROI", *Trading Community Management, June.*

BIC. 2001. XML Convergence Workgroup: High-Level Conceptual Model for B2B Integration, *Business Internet Consortium ,Version: 1.0, 5 October.*

Coad, P. & Yourdon, E. 1992. Object-Oriented Analysis, *Prentice Hall, Englewood Cliffs, N.J.*

ebXML. 2001a. Ebxml technical architecture specification, *11May*, [www.ebxml.org](www.ebxml.org)

ebXML. 2001b. Business Process Team 10 May 2001 Business Process Analysis Worksheets and Guidelines. *Procedures for Developing Business Processes in ebXML v1.0, July.*

Gartner Group. 2000. OAGI: Fostering Standards Evolution or Revolution?. *A Gartner Advisory Research Note 14 December.*

Glushko, R. J. Tenenbaum, J. M. & Meltzer, B. 1999: An XML Framework for Agent-based E-commerce, *Communications of the ACM*, 42(3).

Heather, K. 2001. Web Service Conceptual Architecture 1.0, May 2001, *IBM Software Group.*

ISO 10303-11. 1994. Industrial automation systems: and integration — Product data representation and exchange – *Part 11: Description Methods: The EXPRESS language reference manual.*

ISO 10303-22. 1997. Industrial automation systems and integration. – *Product data representation and exchange– Part 22: Implementation methods: Standard Data Acess Interface.*

ISO 13584-42. 1998. Industrial automation systems and integration. *Parts library– Methodology for structuring Parts Families, ISO, Genev.*

Jasper, R. & Uschold, M. 1999. A framework for understanding and classifying ontology applications. *In B. Gaines, R. Cremer, and M. Musen, editors, Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW'99 (16-21 October 1999, Banff, Alberta, Canada), volume I, pages 4–9–1— 4–9–20, Calgary. University of Calgary, SRDG Publications.*

McGuinness, D. L.1999. Ontologies for Electronic Commerce. In *Proceedings of the AAAI '99 Artificial Intelligence for Electronic Commerce Workshop* , Orlando, Florida, July.

Meyer, B. 1998. Object-Oriented Software Construction, *Prentice hall, Hemel Hempstead, UK.*

Peat, B. & Webber, D. 1997. XML/EDI - the E-business framework, August 1997, [http://www.geocities.com/WallStreet/Floor/5815/startde.htm](http://www.geocities.com/WallStreet/Floor/5815/startde.htm). [Perkowitz & Etzioni, 1997] M. Perkowitz and O. Etzioni: Adaptive Web Sites: an AI Challenge. In *Proceedings of the 15th International Joint Conference on AI(IJCAI-97)*, Nagoya, Japan, August 23-29.

Pierra, G. 1994. Modelling classes of pre-existing components in a CIM perspective*: the ISO13584/ENV 400014 Approach, Revue internationale de CFAO et d'Infographie, vol. 9, n°3, 1994 , pp. 435-454.*

Pierra, G. 2000. Représentation et échange de données techniques, *Mec. Ind., 1, pp.397-400.*

Randy, E. Hall. 2001. "W3C Web Service Position Paper from Intel", *W3C Workshop, April 12.*

RosettaNet. 2001a. RosettaNet Architecture Conceptual Model, *July.*

RosettaNet. 2001b, RosettaNet Implementation Framework: Core Specification *Version: Validated 02.00.00 13 July.*

RosettaNet. 2001c. Specification: PIP Specification Cluster 2: Product Information, Segment A: Preparation for Distribution, PIP2A9: Query Technical Product Information, *Validated 01.01.01, 01 November.*

RosettaNet. 2001d. Specification: Protocol PIP Specification Cluster 2: Product Information, Segment A: Preparation for Distribution, PIP2A9: Technical Product Information Exchange Protocol. *Validated 01.01.01, 01 November.*

Schenck, D. & Wilson, P. 1994. Information Modelling The EXPRESS Way, *Oxford University Press.*

United Nation, UN/EDIFACT-*Directory* 1999. [http://www.unece.org/trade/](http://www.unece.org/trade/) untdid.

Westarp, F. V. Weitzel, T. Buxmann, P. & König, W.1999. The Status Quo and the Future of EDI - Results of an Empirical Study. In *Proceedings of the European Conference on Information Systems (ECIS'99).*