Workshop on Databases and Software Engineering

A MetaModel-based XML Document Warehouse Architecture

Huei-Huang Chen Chun-Feng Hung

hhchen@ttu.edu.tw hcf08@pchome.com.tw

Department of Computer Science and Engineering, Tatung University
No. 40, Chung-Shan N. Rd., Sec. 3, Taipei, 104, Taiwan, R.O.C.
Tel: 886-2-25925252 ext.3295 Fax: 886-2-25925252 ext.2288

Abstract

In the last decade of the 20th century, because of the popularity of Internet, the trend is towards e-solutions for businesses. Not only apply on the electronic commerce but also on the information exchange to decrease time from material in the manufacturer to products brought by customers. However, the problem we confront today is that there are full of e-documents in businesses. This paper provides an overview of the technologies and design issues that we have explored to meet the needs of enterprise information integration infrastructure. We propose a modeling metadata to highlight the intelligent document warehousing management to enable the enterprises to have overall document management.

Keywords: Metadata, XML, Document Warehouse

1. Introduction

In recent years, the number of digital information storage and retrieval systems has increased immensely. Text plays more important role in providing decision makers with a more expansive model of business intelligence (BI). The problem we confront today is that we have too many e-documents at business, so businesses are at once drowning in and starved for information. Besides, for most of us, thinking about BI brings to mind data warehouses, multidimensional models, and ad hoc reports. But it isn't limited to numeric reports and graphs. Imagine if we had databases filled with transactional data but no way to distill the information down into key pieces of information. So that business not only need data warehouse to support decision making but also need document warehouse to manage every e-documents. Most managers cannot get an overview of the contents of a large document collection without enlisting the help of a researcher, analyst, or staff member to review a group of documents, identify the relevant ones, read the text, and summarize the findings.

Document warehousing is one approach to dealing with a glut of textual information [1]. They are designed to answer the "WHY" question. The answers are focused on the whole documents' presentation. In order to overcome the lack of data warehouse, we propose the document warehouse architecture. By supporting between data warehouse and document warehouse each other (as show in Figure 1.1), bringing business a complete BI.

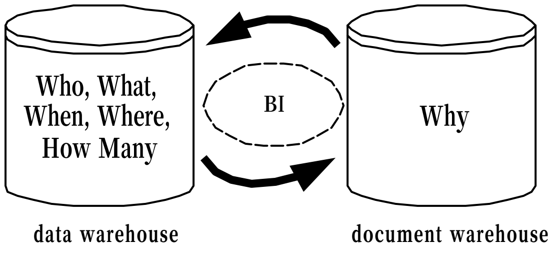


Figure 1.1：Supporting Each Other Bring Business A Complete BI

Table 1.1：Characteristic of Document Warehouse

Document Warehouse
✍✍ Maintain-oriented
✍✍ Related
✍✍ Version-varying
✍✍ Nonvolatile

We will propose a method for maintaining a warehouse of XML-based documents that can be retrieval and analysis from multiple perspectives based upon the semantic content of the document, not only just on external attributes such as file name. Such document warehouses increase the precision and efficiency of document exploration. As Table 1.1, the architecture' s features we propose are maintain oriented, related, version-varying, and nonvolatile. Maintain oriented means we will remain the whole complete document content instead of extracting some data that is organized around a particular topic because the every information in the document are all important for us. We will have the version control. Furthermore, by document-based unit, we will have some relation between documents. This means that some documents are reference. Nonvolatile means that original documents can't be modified. If some data need to change, we have version to control it.

2. Related Works

2.1. Defining the Document Warehouse

A document warehouse makes up a deficiency of data warehouse, so we will first review two definitions of a data warehouse. According to Bill Inmon (1995), the four defining characteristics of a data warehouse are: subject-oriented, integrated, time-variant, nonvolatile. A common practice in such situations is to archive data that is more than three years old.

Ralph Kimball' s describes a data warehouse as "a copy of transaction data specifically structured for query and analysis" (1996). Here, the emphasis is on restructuring transaction data to make the information entailed in the raw data easily accessible. This is also the key driver in document warehousing. To sum up, document warehousing is a metamodel-based environment that enables to capture, link and retrieve various types of documents in a form that can be easily assessed and immediately distributed [2].

2.2. Document Warehouse Architecture

The document warehouse can be divided by some components [2]:

✍✍ Document sources✍ In general, there are three distinct locations of sources:

internal sources, the Internet, and subscription services.

✍✍ Processing servers✍ This job is to identify document formats and, if necessary, convert the document to a format acceptable to other text processing tools. The goal is to distill text into a form suitable for efficient retrieval and text mining.

✍✍ Textbases or other storage repositories✍ Textbases, such as Lotus Notes, have been widely adopted in large organizations. The key benefits of these applications are easily configured interfaces, a rich set of text-searching features, flexible programming tools, and scalable architectures.

✍✍ Metadata repositories✍ Metadata is information describing documents and contents and is a critical piece of the document warehousing environment. Metadata serves several purposes, for instance, improves search precision and recall; allows for extended searching options; categorizes documents; indicates relationship between documents.

✍✍ User profiling✍ End-user operation, in many respects, is the most important operation of the document warehouse. The documents have been gathered, converted, and translated as necessary, then thematically indexed, grouped into similar clusters, summarized, routed to interested readers, and finally recorded in the document warehouses. Now end users are relying upon the document warehouse to support their tasks.

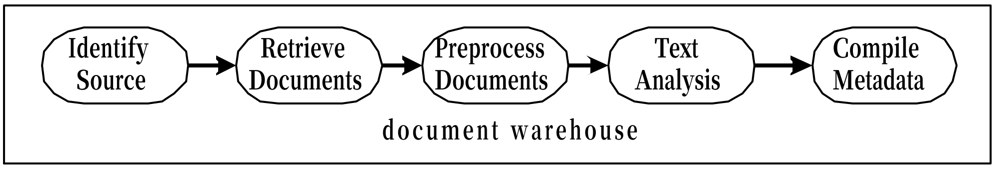


Figure 2.1：The Basic Steps in Document Warehouse Construction

As Figure 2.1 shows, the basic steps to building a document warehouse.

1. Documents of document warehouse are not fixed. Unlike data warehouses, identifying data sources is initially done during the requirements and design

phases of developing the repository.

2. Document retrieval is tightly linked to document identification.
3. The objective of the preprocessing stage, like the transaction phase in data warehousing, is to format the information in a consistent manner to support later operations in the document warehouse process.
4. Text analysis linguistic analysis is preformed, key features and facts are extracted, documents are indexed by topic, and summaries are generated.
5. Compiling metadata is a critical operation because metadata makes explicit, in easily queries form, the essential attributes of a document, such as the key topics, author, source, date of publication, and other factors that can be used to determine its relevancy to a particular issue.

2.3 MetaModle Standards

Metamodel was designed to model the metadata needs of warehouses. There are two bodies working on the standards for metadata: the Object Management Group (OMG) and the Meta Data Coalition (MDC). OMG is a larger established forum dealing with wider array of standards in object technology. In June 2000, the OMG unveiled the Common Warehouse Metamodel (CWM) as the standard for metadata interchange. MDC formed as a consortium of vendors and interested parties in October 1995 to launch a metadata standards initiative. The coalition has been working on a standard known as the Open Information Model (OIM). Unlike the CWM, it extended its coverage beyond the warehouse to include a wide range of information systems, such as analysis and design, components and objects, database and warehousing, and knowledge management.

- Common Warehouse MetaModel (CWM)--CWM includes support for transformations and business nomenclature, is an OMG standard developed by

IBM, Oracle, Hyperion, Unysis, NCR, and other data warehouse vendors [3]. It describes 11 distinct types of metadata along with foundational elements of the model.

Relational data sources Record data sources XML data sources
Business nomenclature Warehouse process Data mining
Information visualization Warehouse operations Data transformations
Multidimensional data sources On-Line analytical processing (OLAP)

✍✍ Dublin Core Metadata Initiative--The Dublin Core is a metadata standard designed to support resource discovery and was initiated in 1995. It does not presuppose a particular subject matter area or specific type of data like some metadata standards. Now it is becoming a de facto standard on the Internet resource metadata across disciplines. The Simple Dublin Core is a set of fifteen elements that use attributes value pairs to describe a document [4]. The fifteen elements show as follow.

DC.Title DC.Creator DC.Subject DC.Description
DC.Publisher DC.Contributor DC.Date DC.Type
DC.Format DC.Identifier DC.Source DC.Language

2.4 Understanding XML

XML is the new generation of eXtensible Markup Language. It is brought up by W3C in 1996. In fact, XML is a simple version of SGML for network. XML have the extensibility, self-description, well-construction of SGME but doesn’t have the disadvantage of SGML such as large and can’t be popular. On the other hand, it is established to enable system-independent, platform-independent and complete electronic text interchange environment for authoring and delivery of information resources across the Internet. Since then, many standards of data interchange have been built on top of XML, because XML enables you to specify data in a very precise way.

✍✍ XML Family Technique

XML allows authors to define their own sets of markup elements that are most appropriate to the specific class of documents they are dealing with. Related technologies like XSL, XPointer, and XLink are used by authors to associate these elements to some rendering or linking semantics for their display on paper or screen. Figure 2.2 shows the relation between XML family techniques.

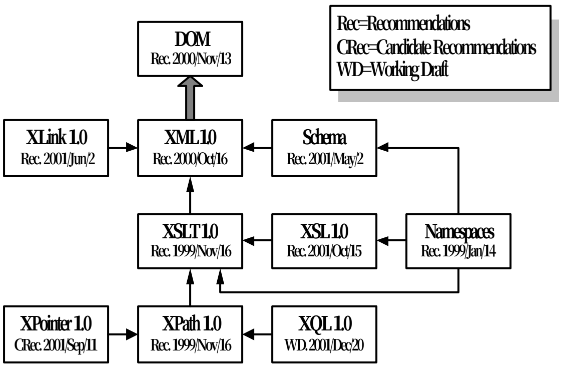


Figure 2.2：XML Family Technique' s Relation

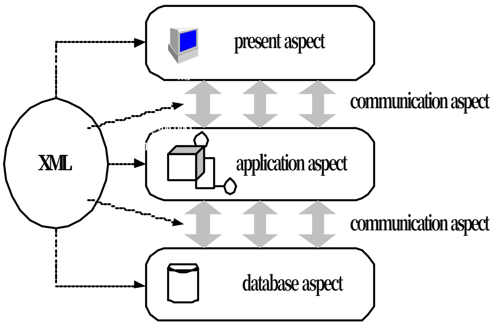


Figure 2.3：XML' s Impact Between Information Exchanging Modeling

✍✍ *XML's Improve in IEM (Information Exchanging Modeling)*

XML will go a long way toward simplifying the integration of utility systems. After some additional standardization efforts XML will be effective for implementing and providing interoperability between disparate systems (as Figure 2.3).

1. **Databases of Aspect**--XML and database technology complement rather than complete with each other. Because XML makes the structure in nontabular data explicit, database technologies can provide some of the amenities found in relational databases. Conversely, database techniques can improve the integrity and semantic integration of XML resource sets.

2. **Applications of Aspect**--XML supports validation in two ways [5]. Application developers can associate an XML document with a document type description (DTD or Schema) that describes the structure to which the document should

conform. Applications can use off-the-shelf XML parsers to validate imported data for conformation to a DTD or schema.

3. **Presentation of Aspect**--We can use XSL expresses rules that indicate how to transform an XML document to a presentation format such as HTML or PDF, or to an alternate representation of the content such as XML document with a different DTD. However they can manage content independently of its presentation, and they can use different XSL style sheets to product alternate views of that content [6].

4. **Data Communication of Aspect**--An application or Web portal uses each sources native interface to communicate directly with source database and reconciles the data it receives. Besides, one application or database uses structures messages to pass data to others. It invokes another and structured messages to as parameters. For example, we can improve this solution in EAI (enterprise application integration) products [7].

3. MetaModel-based XML Document Warehouse Architecture

The primary goal of our framework is to design document warehouse for managing documents inside the business. We have tried to find out the most general pattern of document warehouse.

3.1. Our Architecture

The whole architecture of our approach depicts in Figure 3.1. The document sources which are useful for business maybe heterogeneous and come from difference platform. These data sources are processed by ETL (extraction, transformation, and loading) and recorded. We will store the original document in repository and extract some futures in warehouse. Sometimes they need to analysis by text mining. Those documents are monitored by management server, processed in a multidimensional way and supplied to the users to be searched. Storage server provides the space to store physical and lo gical data. Presentation Server provides a browser interface for users to access documents.

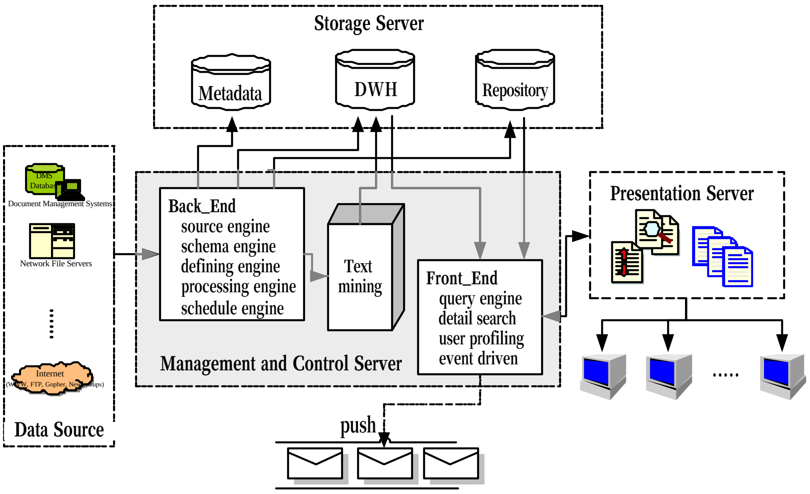


Figure 3.1 ： The Architecture of Our Approach

3.2. Overview the Components

3.2.1. Storage Server

The storage section contains metadata clearinghouse, warehouse storage, and repository storage.

✍✍ Metadata clearinghouse--Metadata is an emerging approach to build document warehouse in a structured manner and support precise retrieval. We keep the information about the logical data structures, the information about the extract attributes. We fall metadata into three major categories that encompasses both physical documents and logical techniques: Content metadata、 Business metadata、 Technique metadata.

✍✍ Warehouse Storage--We store some additional attributes about documents and most important elements inside this document for index search in warehouse storage, such as the attributes of document, keywords, and future about source documents. By this method that to have efficiently search.

✍✍ Repository Storage--Searching out the source of document warehousing texts can be as slow and arduous a task. So we will store the original documents in repository to have precise management.

3.2.2. Management and Control Server

This section contains the data in and data out strategies, rules for the driven engine, mapping of document in warehouse and local database in data sources, and templates for query translation. The management and control component coordinates the services and activities within the warehouse. It is composed of back end components, text mining, and front end components (see Figure 3.1). At back end section, it has component control the data transform and the data store into the warehouse. At front end section, it manages the information delivery to the users and monitors the movement of data in the warehouse. Text mining can help us understand what the text means, extract summary, etc.

3.2.3. Presentation Server

"Build it and they shall come" is an anathema in document warehousing. Often the value of business intelligence information is lost if it is not distributed to the right people or not published in a timely manner. Recently web environments have been changed from hypertext structure to database-related web application type. This server involves using XML-based technologies reduce development time in web environment [8].

Another aspect is information visualization which the process of representing graphically or geometrically complex relationships between data elements. In document Warehousing visualization is especially important for complex and intricately connected by pretext documents. The primary purpose is to allow users to render the logical structure of a section of the document Warehouse in a multidimensional form.

3.3. Metadata Clearinghouse

The data flow in warehousing system is multiple types and flow in multiple directions. We try to discuss three regions: Content Metadata (Figure 3.2), Business Metadata (Figure 3.3), and Technical Metadata (Figure 3.4). In order to help users to be aware of useful content buried in computers. There are some methods of adding metadata to content. Like Durbin Core, it tried to provide users with elements for describing meta-tags that show the characteristics of content. This is suitable for representing static information embedded in documents.

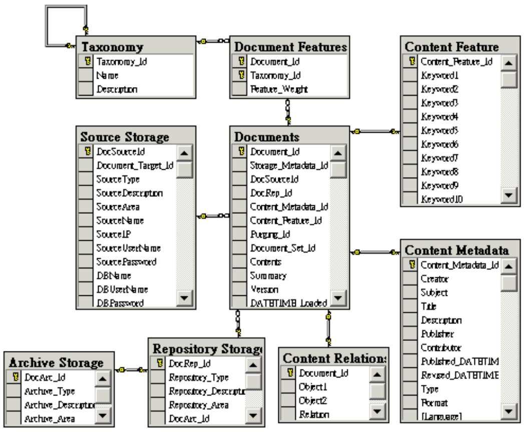


Figure 3.2：Document Content Metadata

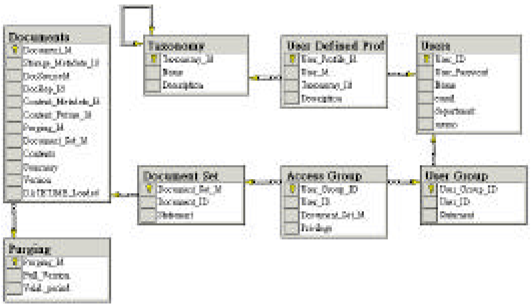


Figure 3.3：Business Rule Metadata

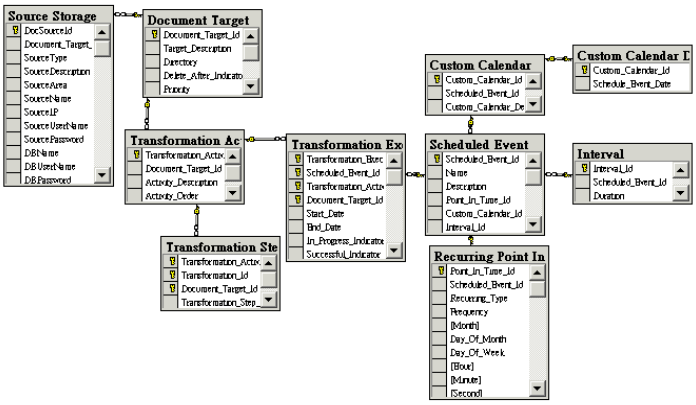


Figure 3.4：Transaction Metadata

✍✍ Content Metadata

The document data model is at the heart of the logical model of the warehouse. The document warehouse logical model is similar the star schema architecture used so often in data warehouses. The entity descriptions show in Table 3.1. Documents like the fact table. Content, source, and feature correspond to dimensions in a dimensional model. Documents have a many-to-one relationship with content relations and document features. Features ideally correspond to the preferred terms used in a thesaurus or taxonomy.

Table 3.1：Content Metadata Entity Description

Entity	Description
Documents	like the fact table in data warehouse
Content Metadata	record the documents’ extract attributes (We use the Dublin Core’ s standard)
Content Future	record the documents’ internal attributes
Source Storage	describe the local data source information
Repository Storage	describe the original data store information
Archive Storage	Describe the older data store information
Document Feature	record document’ s classification and weight
Document Relations	documents’ relation such as reference and be referred
Taxonomy	like dictionary used for document feature

✍✍ Business Metadata

Business metadata helps both end users and programmers to understand what is in the warehouse. At content metadata section, it doesn’ t provide enough information such as access control, versioning, and purging. Business rule define how long documents are kept in the warehouse repository, users’ authority, etc. User profiling implies the need for an added entity in the document data model to represent document groups. The logical representation, documents can belong to multiple groups, and user may have access to one or more groups. By this way users can be granted privileges to view that limit the scope of access. Table 3.2 is the entity descriptions.

Table 3.2：Business Rule Metadata Entity Description

Entity	Description
Documents	like the fact table in data warehouse
Purging	record the document valid period stored in repository
Document Set	clustering the document by privilege
Access Grants	access authority
User Access Group	clustering the users
Users	business user' s description
User Defined Frofile	set user's interest profiles
Taxonomy	like dictionary used for document feature

✍✍ Technical Metadata

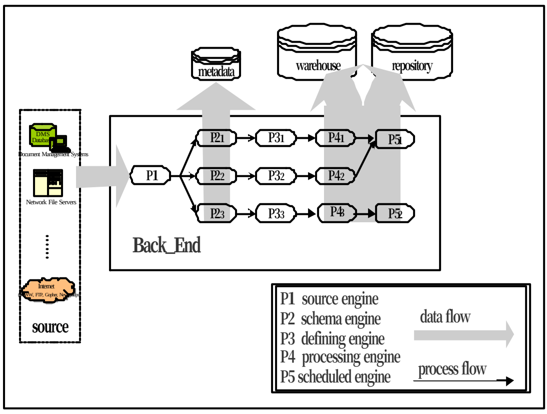
Technique metadata deals with documents warehouse processes. As Table 3.3 shows, multiple document sources may be associated with a single document target. Multiple transformation activities can be associated with a document target. A transformation step may have multiple parameters associated with it. Schedule event control how frequently the transformation activities execute.

Table 3.3：Transaction Metadata Entity Description

Entity	Description
Source Storage	describe the local data source information
Document Target	specify the directory in the staging area that will hose the documents retrieved
Transformation Activities	define a transformation on a document target. On the other hand, a logical set of steps that is curried out to perform a logical task
Transformation Steps	define the all of steps of a activity
Translation Execution	record the results of execution of a particular transformation
Scheduled Event	define when the execution should occur (supports three different types of time specifications)
✍✍**Point in Time**	specify a frequency or a recurring period
✍✍**Custom Select**	provide a list of days
✍✍**Interval**	define the period of time between executions

3.4. Processing Steps

In this session, we have made a transition from the physical architecture of the document warehouse to the logical design considerations. Figure 3.5 is the back end processes. Figure 3.6 is the front end processes. They are all on the management server area of Figure 3.1.



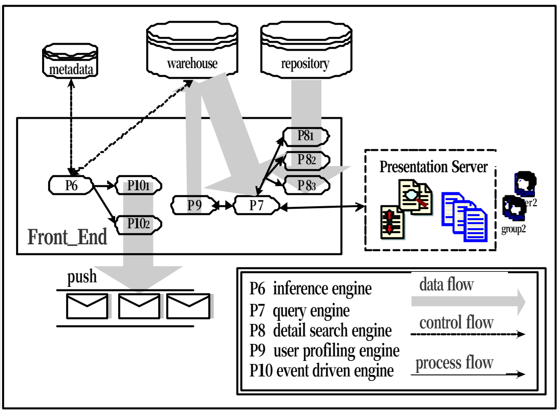


Figure 3.5：Back_End Scenario

Figure 3.6：Front_End Scenario

Step1. (Source engine)—The functions of the source engines are to manage data from data sources. **Step2.** (Schema engine)—Schema engine means to get the data structure of source data and sends it to the metadata clearinghouse to be stored. **Step3.** (Defining engine)—The data before stored in the warehouse may also be sent to text mining server or to any analysis tools to filter or extract summary, or to metadata if it belongs to schema information. **Step4.** (Process engine)—This means to do ETL translation operations. **Step5.** (Schedule engine)—Schedule engine is to define what time to load data into warehouse and repository. **Step6.** (Inference engine)—Inference engine is a set of rules to monitor the data's trend, user profiles and support the push activities of event driven. **Step7.** (Query engine)—This receives the query from user and search the result return back. **Step8.** (Detail search engine)—This detail search engine coordinates the repository storage and drive the results. It maintains a database of XML documents that can be queried for retrieval based on structure and content. **Step9.** (User profiling engine)—By interacting with users to understand what their behavior is. **Step10.** (Event driven engine)—Event driven uses user profiling and a group of rules to actively notify users information may is interesting for them. The information generated is sent by mail.

4. Case Study and Implementation

4.1. RosettaNet Introduction

RosettaNet consortium was created to define standard processes and interfaces to manage supply chains. It was officially formed in 1998 to standardize e-Business activities in the high-technology industry. It defined open interfaces between manufacturers, distributors, resellers, and other participants in the supply chain [9]. Catalyzing RosettaNet's development was the perception that EDI had largely failed because of its expense, its proprietary nature, and its inability to adapt to changes in B2B processes. In addition, EDI was weak at real-time information exchange and was unable to keep up with high-volume information flows. XML was added to take the place of EDI as a mere data interchange standard.

RosettaNet's mission is to facilitate electronic exchange of standard business documents between trading partners, adhering to the Partner Interface Processes (PIPs) specified and standardized by RosettaNet. Fundamental to this are the RosettaNet Implement Framework (RNIF), the PIP specifications, and the business and technical dictionaries.

The most important aspect of RosettaNet is the development of common partner interface processes (PIPs) and common dictionaries. They can be used to support the B2B application integration dialog, provide alignment within the overall supply chain process, allowing businesses to interact at a number of levels to support the processes of common trading communities (see Figure 4.1). PIPs like the dialog of humans.

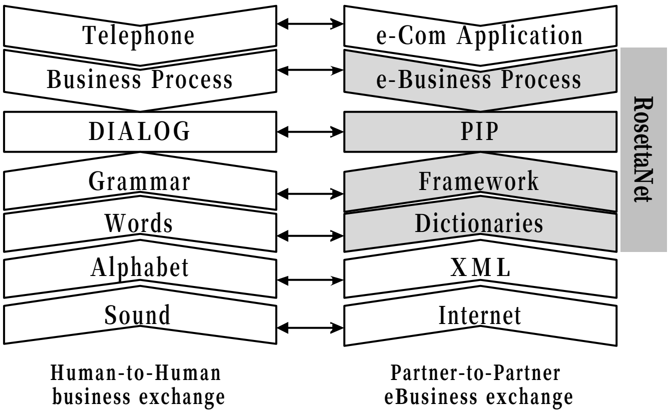


Figure 4.1：RosettaNet Position in Information Exchange Model

A PIP specification includes three major views of the e-Business PIP model. Now we will by the PIP3A4 , “ Request Purchase Order”, to illustrate the example. Business of View (BOV) provides the semantics of the business data entries and their exchange flow between roles during normal operations. The content of the BOV section uses the PIP Blueprint document created for the RosettaNet Business community. Figure 4.2 is an example of BOV flow diagram using PIP3A4. Figure 4.3 is the message guideline which defines the interchange information.

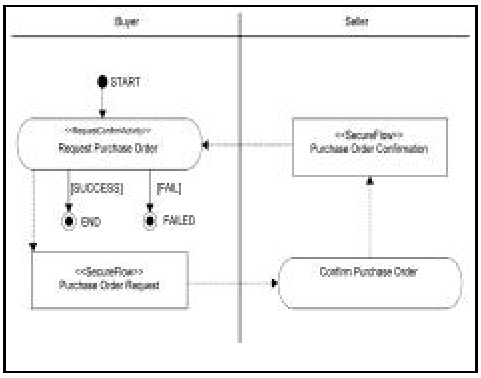


Figure 4.2：PIP3A4 BOV Flow Diagram

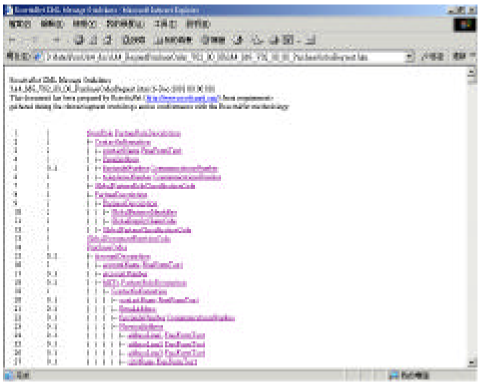


Figure 4.3：PIP3A4 Message Guidelines

4.2. Implement Architecture

Now, we will illustrate how to implement this architecture that we have proposed step by step. As Figure 4.4, we will by XML-based technology to build our document warehouse.

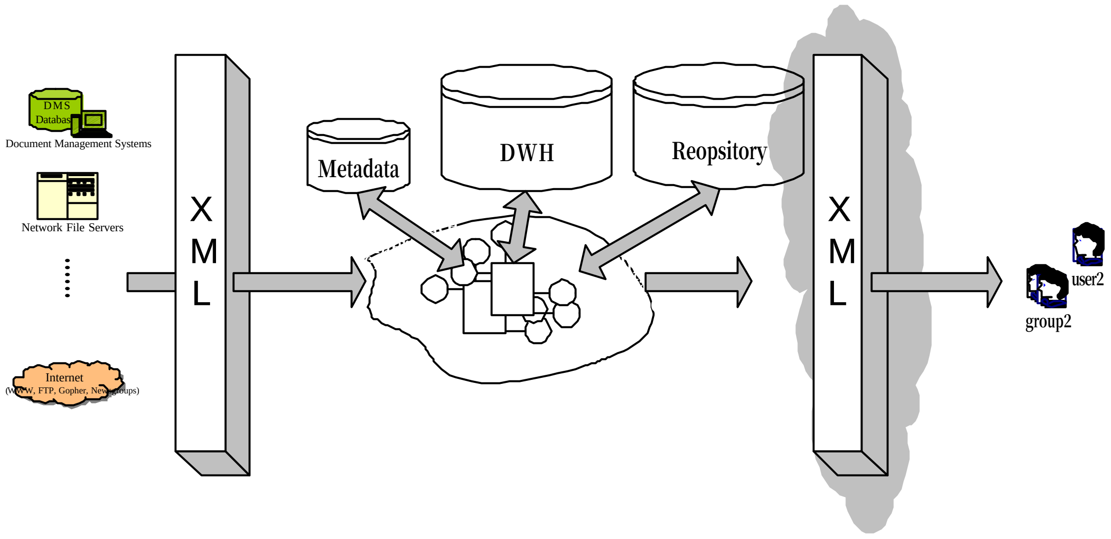


Figure 4.4：Implement Architecture

Now we have to establish a Web interface for users to search or manage. At first, we can by login limit users' arrange by different authorities to have different interface. In Figure 4.5 is the advanced search for users. They can have internal or external attributes search about the document they want to see. Figure 4.6 is the search result and Figure 4.7 shows the all attributes for users that they have selected documents. Moreover, when users click the original icon search engine will sent the comment to the detail search engine for local server to get the results. As Figure 4.8 it will map XML file and XSL file to show in Browser.

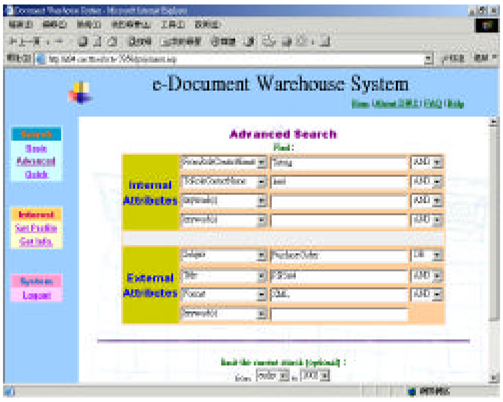


Figure 4.5：Advanced Search Interface

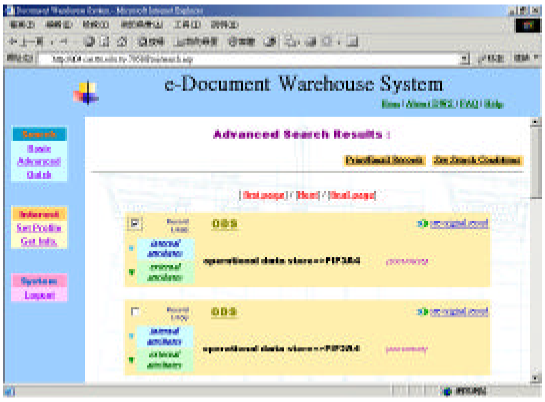
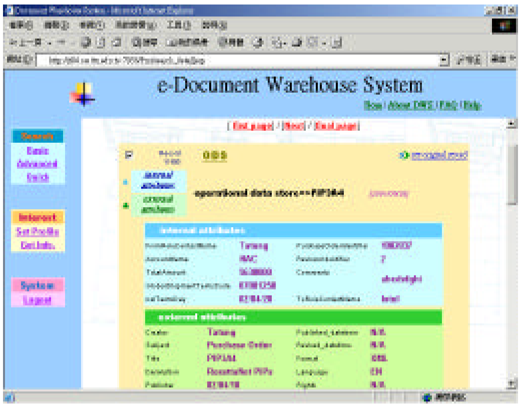


Figure 4.6：Advanced Search Results



e-Document Warehouse System

Home | About DWS | FAQ | Help

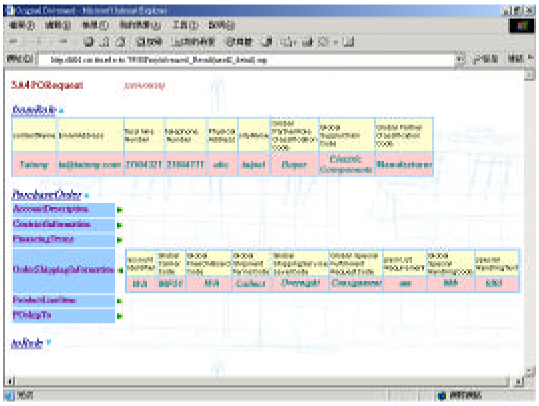
[First page] / [Next] / [Final page]

operational data store==PIP3A4

internal attributes

external attributes

Creator	Tatung	Published_address	N/A
Subject	Purchase Order	Revised_address	N/A
Title	PIP3A4	Format	XML
Description	RosettaNet PIPs	Language	EN



3A4 PORequest

PurchaseOrder

AccountDescription

ContactInformation

FinancingTerms

OrderShippingInformation

ProductLineItem

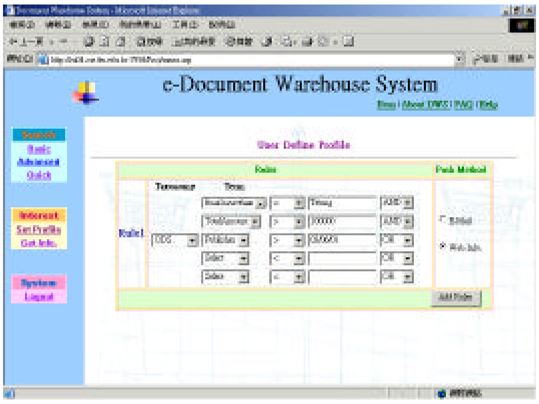
POShipTo

toRole

Figure 4.7：Advanced Search Detail Result

Figure 4.8：Detail See Original Document

Figure 4.9 is the interface for users to set their interest profile. This can by mail or web push information actively to users. Figure 4.10 is for manager to manage the running eve nts. System manager can monitor the transformations by Internet.



e-Document Warehouse System

Home | About DWS | FAQ | Help

User Define Profile

Rules | Push Method

Taxonomy | Term

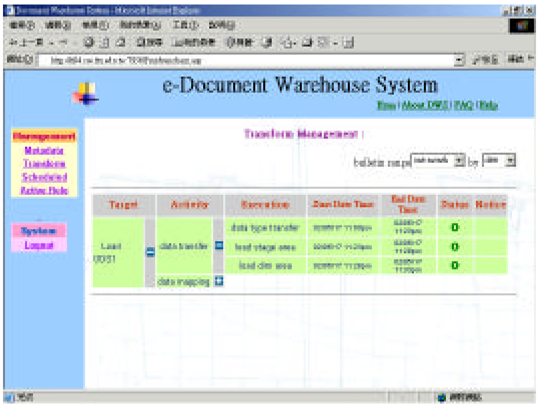
Rule1

ODS

○ Email

● Web Info.

Add Rule



e-Document Warehouse System

Home | About DWS | FAQ | Help

Transform Management

Target	Activity	Execution	Start Date Time	End Date Time	Status	Notice
Load ODS1	data transfer	data type transfer	[illegible]	[illegible]	0	
		load stage area	[illegible]	[illegible]	0	
		load dim area	[illegible]	[illegible]	0	
	data mapping					

Figure 4.9：User Defined Profile

Figure 4.10：Manage the Document Warehouse Running Event

5. Summary

We provided an overview of the technologies and design issues that we have explored to meet the needs of enterprise info rmation integration infrastructure. Moreover, we build an index to have efficient search documents stored on one or more servers in a business. By XML-based the databases may be any combination of Oracle, Sybase or MS-SQL or any other ODBC. The exact information actually maintained in a document warehouse we proposed can include:

✍✍ Automatically monitor versions of documents

✍✍ Presentations of document in several style

✍✍ Rich metadata about document, such as authors' names, publication data, keyword, support search

✍✍ Clustering information about related documents

We focused on the modeling metadata to highlight intelligent management of document warehousing where all the components and connectors are explicitly defined inside a metadata clearinghouse. The proposed architecture was extensible by plugging in new components in management section without affecting other's performances. The inference engine increased the intelligence of the system by using business rules and event driven replace manpower's consume on routine procedure. The contribution of our work was bringing business have the overall document management.

References

1. Bleyberg M. Z., Ganesh K., "Dynamic multi-dimensional models for text warehouses", IEEE International Conference on, Vol:3 2000, pp.2045-2050.
2. Dan Sullivan, "Document Warehousing and Text Mining", 2001, ISBN 0-471-39959-0.
3. Object Management Group, "The Data Warehousing, CWM and MOF Resource Page", http://www.omg.org/cwm.
4. Dublin Core Metadata Initiative, "Dublin Core Metadata Element Set, Version 1.1: Reference Description", http://dublincore.org/documents/1999/07/02/dces.
5. Roy J., Ramanujan A., "XML: Data's Universal Language", IT Professional, Vol:2 Issue:3, May-June, 2000, pp.32-36.
6. Royappa, Andrew V., "Implementing Catalog Clearinghouses with XML and XSL", Proc. The 1999 ACM Symposium on Applied Computing, Sept, 1997, pp.616-621.
7. Len Seligman Arnon Rosentbal, "XML's Impact on Databases and Data Sharing",

IEEE, 2001, pp.59-67.

8. Hwi woong Jeong, Aesun Yoon, “Linguistic Database Handling Using XML in Web Environment”, IEEE International Symposium on, Vol:2 2001, pp.833-838.
9. RosettaNet, “RosettaNet Standard Information”, http://www.rosettaNet.org.