Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

NJIT New Jersey's Science & Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

e-DOCPROS: EXPLORING TEXPROS INTO e-BUSINESS ERA

by

Zhenfu Cheng

Document processing is a critical element of office automation. TEXPROS (TEXt PROcessing System) is a knowledge-based system designed to manage personal documents. However, as the Internet and e-Business changed the way offices operate, there is a need to re-envision document processing, storage, retrieval, and sharing. In the current environment, people must be able to access documents remotely and to share those documents with others. e-DOCPROS (e-DOCument PROcessing System) is a new document processing system that takes advantage of many of TEXPROS's structures but adapts the system to this new environment. The new system is built to serve e-businesses, takes advantage of Internet protocols, and to give remote access and document sharing. e-DOCPROS meets the challenge to provide wider usage, and eventually will improve the efficiency and effectiveness of office automation. It allows end users to access their data through any Web browser with Internet access, even a wireless network, which will evolutionarily change the way we manage information. The application of e-DOCPROS to e-Business is considered. Four types of business models are considered here. The first is the Business-to-Business (B2B) model, which performs business-to-business transactions through an Extranet. The Extranet consists of multiple Intranets connected via the Internet. The second is the Business-to-Consumer (B2C) model, which performs business-to-consumer transactions through the Internet. The third is the Intranet model, which performs transactions within an organization through the organization's network.

The fourth is the Consumer-to-Consumer (C2C) model, which performs consumer-to-consumer transactions through the Internet.

A triple model is proposed in this dissertation to integrate organization type hierarchy and document type hierarchy together into folder organization. e-DOCPROS introduces new features into TEXPROS to support those four business models and to accommodate the system requirements.

Extensible Markup Language (XML), an industrial standard protocol for data exchange, is employed to achieve the goal of information exchange between e-DOCPROS and the other systems, and also among the subsystems within e-DOCPROS. Document Object Model (DOM) specification is followed throughout the implementation of e-DOCPROS to achieve portability.

Agent-based Application Service Provider (ASP) implementation is employed in e-DOCPROS system to achieve cost-effectiveness and accessibility.

e-DOCPROS: EXPLORING TEXPROS INTO e-BUSINESS ERA

by

Zhenfu Cheng

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer and Information Science

Department of Computer and Information Science

May 2001

Copyright © 2001 by Zhenfu Cheng
ALL RIGHTS RESERVED

APPROVAL PAGE

e-DOCPROS: EXPLORING TEXPROS INTO e-BUSINESS ERA

Zhenfu Cheng

12/27/00

Dr. Gary Thomas, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT, Newark, NJ

12/27/00

Dr. Peter A. Ng, Dissertation Co-Advisor — Date
Professor of Computer Science, University of Nebraska at Omaha, Omaha, NE

12-27-00

Dr. D.C. Douglas Hung, Committee Member — Date
Associate Professor of Computer and Information Science, NJIT, Newark, NJ

Dec. 27, 2000

Dr. Ajaz Rana, Committee Member — Date
Assistant Professor of Computer and Information Science, NJIT, Newark, NJ

12/27/2000

Dr. Ronald S. Curtis, Committee Member — Date
Associate Professor of Computer Science, William Paterson University, Wayne, NJ

BIOGRAPHICAL SKETCH

Author: Zhenfu Cheng

Degree: Doctor of Philosophy

Date: May 2001

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer and Information Science,
  New Jersey Institute of Technology, Newark, New Jersey, 2001

- Master of Computer Science,
  New Jersey Institute of Technology, Newark, New Jersey, 1998

- Master of Manufacturing Engineering,
  Beijing University of Aeronautics and Astronautics, Beijing, P. R. China, 1996

- Bachelor of Mechanical Engineering,
  Tianjin University, Tianjin, P. R. China, 1989

Major: Computer Science

Publications:

Xuhong Li, Zhenfu Cheng, Fang Sheng, Xien Fan and Peter A. Ng, "A Document Classification System with Learning Ability," Accepted to *Integrated Design and Process Technology (IDPT 2000)*, 2000.

Xien Fan, Xuhong Li, Fang Sheng, Zhenfu Cheng and Peter A. Ng, "A Scalable Automated System for Document Management," Accepted to *Integrated Design and Process Technology (IDPT 2000)*, 2000.

Xuhong Li, Jianshun Hu, Zhenfu Cheng, Simon Doong, D.C.Hung and Peter A. Ng,"An Integrated Document Processing System: Document Classification and Information Extraction," *Proceedings of the 4th World Conference on Integrated Design and Process Technology*, June 1999.

Xuhong Li, Jianshun Hu, Zhenfu Cheng, D.C. Hung and Peter A. Ng, "Automatic Document Analysis and Understanding System," *The First International Conference on Enterprise Information Systems*, Setubal, Portugal, March 27-30, 1999.

This dissertation is dedicated to my beloved family

ACKNOWLEDGMENT

I would like to express my deep and sincere gratefulness to my advisor, Professor Gary Thomas, for his guidance, valuable input and advise to this dissertation. Thanks to Dr. Peter Ng, Dr. D. C. Hung, Dr. Ajaz Rana and Dr. Ronald S. Curtis for actively participating in my committee and giving me valuable comments for this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

GLOSSARY

e-DOCPROS.....e-Document Processing System

TEXPROS.....Text Processing System

ADoCES.....Automatic Document Classification and Extraction System

WWW.....World Wide Web

B2B.....Business-to-Business

B2C.....Business-to-Consumer

C2C.....Consumer-to-Consumer

HTML.....Hyper Text Markup Language

XML.....Extensible Markup Language

DOM.....Document Object Model

OTH.....Organization Type Hierarchy

DTH.....Document Type Hierarchy

FO.....Folder Organization

CHAPTER 1

INTRODUCTION

1.1 Introduction to e-DOCPROS System

In less than a decade, Internet-based e-Business has emerged as the fastest growing factor in the world for the creation of new wealth and overall economic activity. Internet is so successful and has great potentials that no one would ignore it. Figures for such a fast changing medium look outdated immediately, even if they were correct at the time they were reported. Hence, in this dissertation, they are used sparingly. However, some statistics are so remarkable that we have to mention here. e-Business is growing at an exponent rate [1]. Based on the study funded by CISCO System Inc., Internet-related businesses generates more than $507 billion in revenues in 1999, surpassing for the first time traditional industries such as airlines. The Internet businesses, including Internet-focused software and hardware companies as well as those selling products over the Web, are not expected to subside. If the America's Internet economy were to grow at even half its current rate over the next three years, it would generate $1.2 trillion in revenues in 2002.

TEXPROS (TEXt PROcessing System) is a knowledge-based text processing system to manage personal documents [16-33, 49, 86-91]. A dual model is employed to describe, classify, categorize, file and retrieve documents. This model consists of two levels: a *document type hierarchy*, which depicts the structural organization of the documents, and a *folder organization*, which represents the user's real-world document filing system. In a user's office environment, by identifying common properties for each document type, documents are partitioned into different types. Each document type is

represented by a *frame template*, which describes the common properties in terms of attributes of the documents of the type [35]. After classification, a particular office document can be summarized from the viewpoint of its frame templates to yield a synopsis of the document, which we called a *frame instance*. The frame instances of various document types are deposited into folders over time. A frame instance will generally be in multiple folders along one or more paths from the root. Hence, we consider folders to be heterogeneous repositories that are related by an inclusion relationship to form a folder organization. This folder organization is defined by dividing documents for particular areas of discourse into groups until well-defined groups are reached. The retrieval subsystem of TEXPROS provides functional capabilities for processing incomplete, imprecise and vague queries and provides users with semantically meaningful responses [92-94]. The design of retrieval subsystem is highly integrated with various mechanisms for achieving these goals. Firstly, a system catalog including a thesaurus is used to store the knowledge about the database. Secondly, there is a query transformation mechanism composed of context construction and algebraic query formulation modules. Given an incomplete or imprecise query, the context construction module searches through the system catalog for the required terms and constructs a query that has a complete and precise representation. The resulting query is then formulated into an algebraic expression. Thirdly, in the retrieval process, vague queries can be entered into the system until sufficient information is obtained, through the use of the browser, to the extent that the user is able to construct a query for his request. Finally, when processing of queries fails by responding with a null answer to the user, a generalization mechanism is used to give the user a cooperative explanation for the null

answer. TEXPROS allows users, firstly to retrieve documents and information through frame templates and to match the given values against slot values on the frame instances of a single class or several classes. Secondly, it allows users to manipulate and query folders, and to perform folder-at-a-time operations. Thirdly, TEXPROS allows users to browse through the folders containing frame instances of a specified document type, and the contents of frame instances of a particular type contained in a specific folder. Sometimes, the user may start with a vague idea, and as the search progresses, the notation of what he/she wants becomes clear to the user, and there may be a shift in emphasis. Browsing the folders and the contents of frame instances helps users reformulate queries dynamically. Fourthly, if the user only has a rough idea or only can describe partially the requested documents, he may perform the concept-based retrieval. Documents whose keywords partially match the query are also returned. At times, a user may be impressed with a picture in the document, and may type in words to describe the picture. It is also likely that the input query differs from the description matched partially or conceptually with the query. In the concept-based retrieval, there is no clear distinction between documents that qualify the specified condition and those that do not; some documents are more relevant, while others are less so. For such fuzzy types of queries, TEXPROS always returns a list of documents, ranked according to the degree of their relevance to the query.

TEXPROS is an office-oriented personal system. Its architecture is appropriate for single user but not a multiple user system. However, as the Internet and e-Business change the way offices operate, there is a need to re-envision document processing, storage, retrieval, and information sharing. In the current environment, people must be

able to access documents remotely and to share those documents with others. e-DOCPROS (e-DOCument PROcessing System) is a new document processing system that takes advantage of many of TEXPROS's structures but adapts the system to this new environment. The new system is built to serve e-businesses, takes advantage of Internet protocols, and to give remote access and document sharing. e-DOCPROS meets the challenge to provide wider usage, and will eventually improve the efficiency and effectiveness of office automation. It allows end users to access their data through any Web browser with Internet access, even a wireless network, which will evolutionarily change the way we manage information. The application of e-DOCPROS to e-business is considered. Four types of business models are considered. The first is the Business-to-Business (B2B) model, which performs business-to-business transactions through an Extranet. The Extranet consists of multiple Intranets connected via the Internet. The second is the Business-to-Consumer (B2C) model, which performs business-to-consumer transactions through the Internet. The third is the Intranet model, which performs transactions within an organization through the organization's network. The fourth is the Consumer-to-Consumer (C2C) model, which performs consumer-to-consumer transactions through the Internet.

A triple model is proposed in this dissertation to integrate organization type hierarchy and document type hierarchy together into folder organization. e-DOCPROS introduces new features into TEXPROS to support those four business models and to accommodate the system requirements. These features consist of a multiple user system, an information exchange standard, a document processing data flow system and a dynamic processing, accessible, scalable and security system.

XML (Extensible Markup Language), an industrial standard protocol for data exchange, is employed to achieve the goal of information exchange between e-DOCPROS and the other systems, and also among the subsystems within e-DOCPROS. DOM (Document Object Model) specification is followed throughout the implementation of e-DOCPROS to achieve portability.

Agent-based ASP (Application Service Provider) implementation is employed in the e-DOCPROS system to achieve cost-effectiveness and accessibility.

Within this application domain, many digital libraries already exist and provide services to end users. Also there are several Web sites that provide services to allow users to manage their documents. However, the limitation still exists for those systems. Most of the existing online document management systems are restricted to relatively small application domains and need user interactivity all times, such as StarOffice, Microsoft Online Office, Jungle.com, Damango.com [36]. There are no knowledge base systems to support the whole functionalities of office automation, such as automatic classification, extraction, filing, retrieval and browsing. Therefore, it is challenging to develop an Internet-based system that can be easily adapted into any application domain that provides a high degree of automation by applying artificial intelligence in document understanding and analysis.

In practice, however, user interaction is still needed. It is through user interaction that e-DOCPROS learns the habits or behavior of the users. This information is stored and used by the background knowledge base system that eventually supports an automated dynamic processing system that represents a personalized user interface and user information manager. The system has adaptive capability and is trained by learning

from the user's interaction, which allows the document analysis and understanding to be done accurately, effectively and efficiently.

Most existing document analysis systems [2-15] are restricted to relatively small application domains, such as newspaper reading [37], form content extraction [35, 38], email reading [39, 40, 41, 42] and text categorization [43, 44]. Even if some of the systems can be adapted to a new domain, this adaptation is often as time consuming as developing a new system from scratch. Therefore, it is challenging to develop a system that can be easily adapted into any application domain and provides high degree automation by applying artificial intelligence in document analysis and understanding.

However, e-DOCPROS can process most types of the documents in an office. We consider these documents as semi-structured documents that can be classified into different document types according to the layout and logical structures of the documents.

1.1.1 e-DOCPROS Business Models

Currently, there are four types of e-Business models:

- B2B (Business-to-Business), which performs business-to-business transactions through an Extranet. An Extranet consists of multiple Intranets connected via the Internet.
- B2C (Business-to-Consumer), which performs business-to-consumer transactions through the Internet.
- Intranet Model, which performs transactions within an organization through the organization's network.

- C2C (Consumer-to-Consumer), which performs consumer-to-consumer transactions through the Internet.

Here, the term "business" mainly refers to a company, but is not limited to it. We can generalize it to any organization, even a governmental agency. In this dissertation, we do not distinguish business from a company and an organization.

1.1.2 e-DOCPROS System Requirements

e-DOCPROS introduces six new features into TEXPROS to support and accommodate the above business models, all of which are multiple user system, data exchange standard, data flow and dynamic processing, accessibility, scalability and security.

1.1.2.1 Multiple User System. TEXPROS mainly focuses on personal document management. e-DOCPROS addresses the challenges inherent in a multiple user that operates over the Internet with large numbers of users. Hence, a multiple user system is a must for e-DOCPROS.

1.1.2.2 Data Exchange Standard. There are two kinds of requirements that determine what type of data exchange standard is required in e-DOCPROS. First, since e-DOCPROS supports B2B, which exchanges data among organizations, a data exchange standard is required. Second, since a data exchange standard is required within e-DOCPROS to allow all of the subsystems to exchange data, so as to define the document objects and transform messages in the processing flow. XML (Extensible Markup Language) is employed here to fulfill this requirement. XML will be discussed in Chapter 5.

1.1.2.3 Data Flow and Dynamic Processing. There are several subsystems in e-DOCPROS, such as the classification and extraction subsystem (ADoCES) [88], the storage and filing subsystem and the retrieval and browsing subsystem. Here, e-DOCPROS employs XML to integrate all of the subsystems together and make data flow simply and naturally. In e-DOCPROS, a document consists of two parts [45]. One is document data, which details the document itself. The second part represents the actions that must be applied to, or applied on, a document, which is or will be assigned based on the knowledge base or user specified rules. Software agents then will perform the corresponding actions on a document to transform, eventually, the document into a completed state.

Through knowledge base and the learning capability, the system can process known types of documents, update the rules to existing types or derive rules for new document types to achieve dynamic document processing.

1.1.2.4 Accessibility. Information sharing is an essential feature of the information management system in this multiple user environment. Once a system supports multiple users, especially B2B business model, accessibility is the most important element in the system. e-DOCPROS defines three levels of accessibilities, ownership, view and role. All three levels of accessibilities are integrated into e-DOCPROS system by using XML. With XML, we can achieve this goal easily. It cannot be achieved in the TEXPROS system.

1.1.2.5 Scalability. Even with diverse business models and the complex nature of Internet-based systems, e-DOCPROS is a scalable system. e-DOCPROS is configurable and parameterized to achieve this goal.

1.1.2.6 Security. Security is always an important element in human life. Individuals desire privacy. In business, security is even more critical. Concern for security is extreme when we perform online business, such as B2B, B2C and C2C. Anti-virus is another hot topics for Internet-based system. e-DOCPROS allows users to upload any kind of documents onto the server, so virus-free documents or anti-virus software are vital to the system.

1.2 Comparison Between TEXPROS and e-DOCPROS

TEXPROS is a personal document processing system based on the existing technologies. It employs a dual-model approach to modeling office documents. e-DOCPROS is an automatic knowledge-based e-Business document management system that supports B2B, B2C, C2C and Intranet business models. All subsystems are Webtop applications, designed by using ASP (Application Service Provider) model and implemented by Java programming language. There were six new features described earlier that are introduced into e-DOCPROS. These features do not exist in TEXPROS; hence, e-DOCPROS is more suited to an e-Business solution for office document management. e-DOCPROS increases the capability of document filing and information sharing and provide the collaborative environment for document processing.

1.3 e-DOCPROS Approach

e-DOCPROS is an automatic, knowledge-based, e-Business, document management system. The system provides functional capabilities for automatically classifying,

categorizing, storing, retrieving and reproducing documents, as well as extracting, browsing, retrieving and synthesizing information from a variety of documents.

This dissertation explores transformation of a personal document processing system into an Internet-based multiple user document processing system. Chapter 2 describes the four e-DOCPROS business models and the importance of the Internet to e-Business document processing. The system requirements of e-DOCPROS are detailed in Chapter 3. Here XML is used to describe the document, data flow and dynamic processing. Scalability and security are also introduced due to the nature of a multiple user system and e-Business system. All of those features are integrated through XML. Chapter 4 describes the triple model, multiple user system and information sharing in e-DOCPROS. Chapter 5 details on XML and how e-DOCPROS employs it to integrate the system together. Chapter 6 focuses on system architecture and the implementation of e-DOCPROS. More details are covered in this chapter with regards agent-based implementation and ASP (Application Service Provider) model. All subsystems are designed and implemented as an agent-based, Webtop application, which can achieve portability and cost efficiency. Both paper document and electronic document processing in e-DOCPROS are discussed here. Chapter 7 gives the conclusions of this dissertation and discusses future works.

CHAPTER 2

e-DOCPROS BUSINESS MODELS

e-Business is a real-life reality that has evolved rapidly over a short period of time. e-Business can be categorized into four types, each of which existed long before e-Business became a reality. They are the followings: 1) business between companies and 2) business between companies and consumers and 3) business within an organization, and 4) business between consumers themselves. Those types of business models decisively influence the form of the e-Business models. Therefore, correspondingly, four types of e-Business models are formulated, which are B2B, B2C, C2C and Intranet model.

2.1 Business-to-Business Model

It is estimated that business-to-business sales of products and services online will grow from $131 billion in 1999 to $1.5 trillion in 2003 [46]. B2B revenues will climb to $7.29 trillion by 2004. At that time they will also represent 7% of the $105 trillion total global sales transactions [47,48].

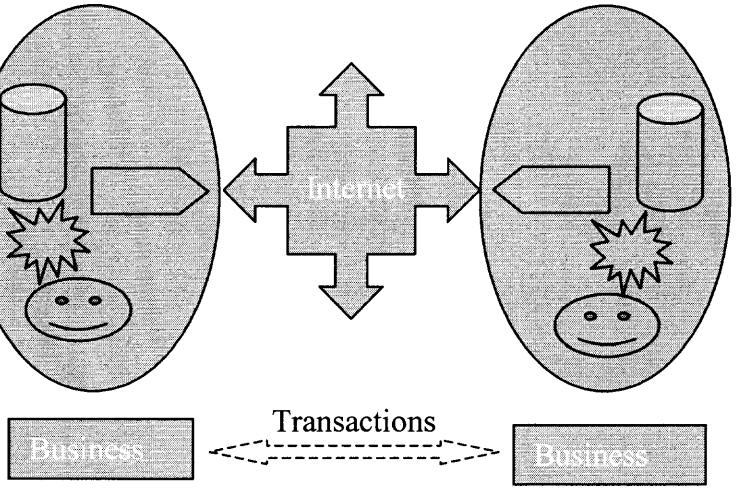


Figure 2.1 B2B business model

In the B2B (Business-to-Business) model, business-to-business transactions are performed through an Extranet. An Extranet consists of multiple Intranets connected via the Internet. Figure 2.1 depicts B2B business model. All transactions, including data exchange between organizations, are performed through the Extranet.

2.2 Business-to-Consumer Model

In the B2C (Business-to-Consumer) model, business-to-consumer transactions are performed through the Internet. Typically, this is what most people think of as e-business. But, e-Business is much more than selling products on the Web. B2C includes selling products, technical support and many kinds of services, e.g., hotmail.com, fax.com.

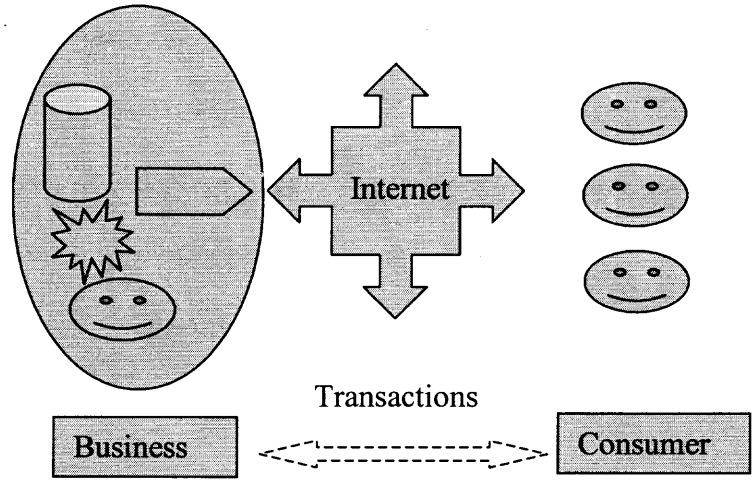


Figure 2.2 B2C business model

2.3 Intranet Model

Intranet Model performs transactions within an organization through the organization's network. The Intranet uses Internet protocols and standards for electronic

communication. People on the Intranet are able to see organization specific Web sites. These Web sites are separated from the rest of the world by firewalls and other security measures. People from outside of the organization are not allowed to see these private pieces of information.

IBM is using its "Refurbishing Computer Warehouse" Web site to sell PCs coming off lease. The site allows employees to view the machines' specifications and then purchase them online with a credit card or through traditional methods such as telephone. These offerings are restricted to employees and therefore should not be accessible nor visible to the outside world [1].

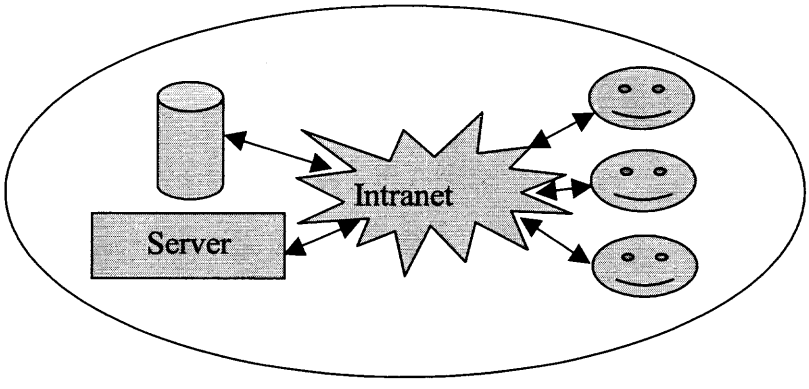


Figure 2.3 Intranet business model

2.4 Consumer-to-Consumer Model

In the C2C (Consumer-to-Consumer) model, transactions are among consumers and are performed via the Internet. The central Web sites provide service and facility that allow customers to perform transactions through the Internet. Examples of C2C sites are trade.com and bluecycle.com.

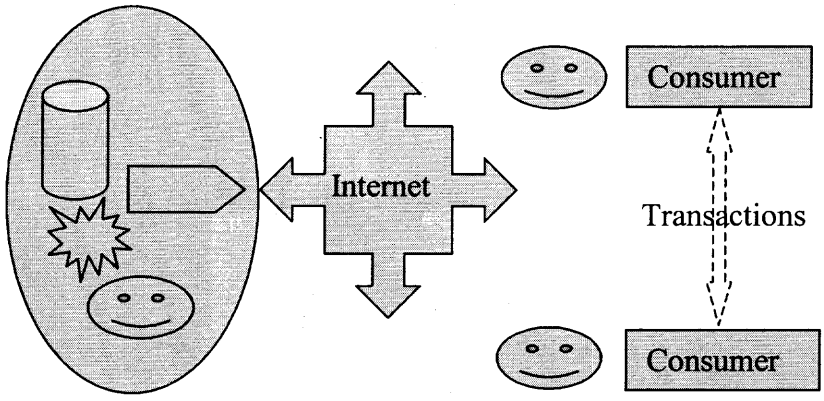


Figure 2.4 C2C business model

e-DOCPROS provides the core functionalities necessary to support these four business models. e-DOCPROS can be easily re-configured or trimmed to meet different desired application domains. The four business models can be easily integrated into e-DOCPROS and benefit the document filing and retrieval subsystems in several ways. First, e-DOCPROS supports multiple user system, which promises a much wider application domain. It gives access to a much wide variety of information than before and the user can browse and exchange information in an extensible scope. Secondly, Internet-based e-DOCPROS provides a more convenient way for users to search and retrieval information with the users outside e-DOCPROS. This is possible because they are all connected to the Internet. Thirdly, the industrial standard of data exchange, XML, provides e-DOCPROS with the capability easily to exchange information with users within or outside of the system.

CHAPTER 3

e-DOCPROS SYSTEM REQUIREMENTS

e-DOCPROS is an Internet-based information management system, which supports B2B, B2C, C2C and Intranet business models. It introduces six new features into TEXPROS to support and accommodate these business models. These new features are that it allows 1) multiple users, 2) data exchange standard, 3) data flow and dynamic processing, 4) accessibility, 5) scalability and 6) security. Among these, the most important one is its ability to support multiple users, which determines the need for all of the other features. The following sections detail the requirement of each of them.

3.1 Multiple User System

In contrast to TEXPROS, which mainly focuses on personal document management, e-DOCPROS supports multiple users over the Internet, even large numbers of users. Figure 3.1 gives an example of an e-DOCPROS document folder organization. Here, the two organizations, Organization A and Organization B, are both inside the system. For each organization, there are two groups, Group AA, Group AB, Group BA and Group BB respectively. There are several users in each group, for example, in Group AA there are User AAA and User AAB. For each user, he/she only can view the folders under his/her control. In this dissertation, we use he/his to represent both he/she and his/her. Users can create their own folder structure according to their needs. For example, User AAA created the following folders, Memo, Letter, Email and Journal and etc. There are subfolders under each of them. For example, under Memo, we have From_Smith, which holds all memos whose sender is Smith. Similarly, the folder, To_Frank, holds all memos

whose receiver is Frank. Folder Journal can be organized based on the author or publisher, for example, Author_Tom holds all journal articles that have Tom as the author. Similarly, Publisher_ACM holds all journal articles for which the publisher is ACM.

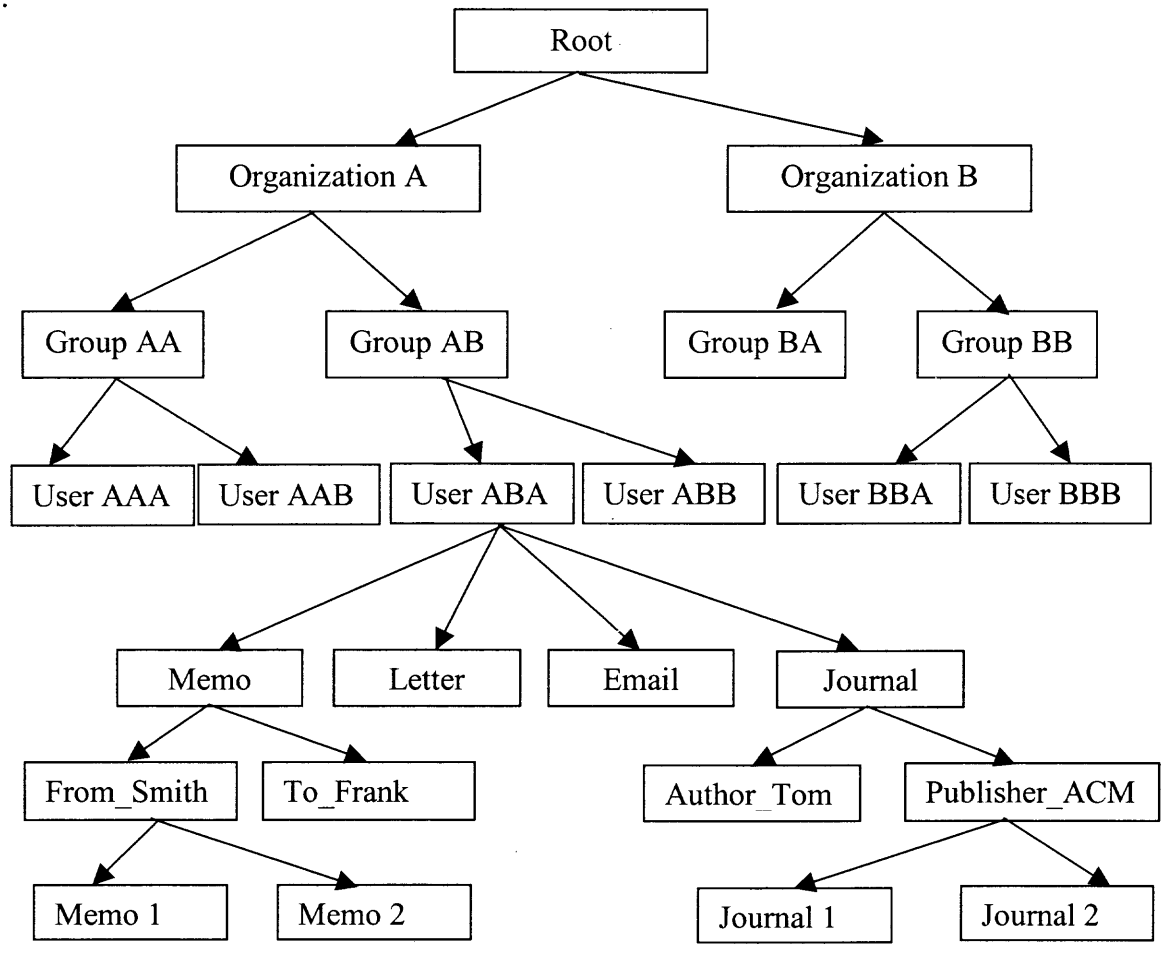


Figure 3.1 e-DOCPROS multiple user system

e-DOCPROS supports multiple levels in the folder organization: organization level, group level, user level and document level. An administration application provides users some mechanisms to manipulate the whole folder organization, which includes system folders and application folders. End users can only manipulate those folders

below themselves. Also, the correct privilege needs to be assigned to a different type of user.

3.2 Data Exchange Standard

There are two kinds of requirements that determine what data exchange standard is required in e-DOCPROS. First, since e-DOCPROS supports a B2B business model, in which exchanges data among organizations, a data exchange standard across all partners is required. Second, within e-DOCPROS, a data exchange standard also is required to define the document objects and transform the message in the processing flow among all the subsystems. In this dissertation, we combine these two types of requirements together by using XML.

Extensible Markup Language (XML) is a self-descriptive markup language, in which data are embedded within two quoted tags, a start-tag and an end-tag. Any data enclosed by the pair of the start-tag and the end-tag is referred to as an element. The start-tag is delimited by using the characters '<' and '>'. The end-tag is delimited by using the characters '</' and '>'. For example,

```
<Email>
    <From>
        <fisrtname>Mike</firstname>
        <middlename>J.</middlename>
        <lastname>Woods</lastname>
    </From>
    <To>
```

```
        <fisrtname>Frank</firstname>
        <middlename></middlename>
        <lastname>Green</lastname>
    </To>
    <Subject>Project Meeting</Subject>
    <Time>
        <year>1999</year>
        <month>01</month>
        <date>18</date>
    </Time>
    <Body>
        Project meeting in Large Conference Room at 3:00PM today.
    </Body>
</Email>
```

The name of an element is defined by using semantic meaning for the data. <From> can be recognized easily as the sender of the email. An element can have subelements in it, such as the markup text <From> contains three markup components, <firstname>Mike</firstname>, <middlename> J. </middlename> and <lastname>Woods </lastname> recursively. The tags of XML contain the meaning of the data in the element, the user and software can determine the meaning of the document by reading through the tags. So XML can be used to represent e-DOCPROS's frame templates.

XML will be detailed in Chapter 5.

3.3 Data Flow and Dynamic Processing

There are several subsystems in e-DOCPROS, such as the classification and extraction subsystem (ADoCES), the storage and filing subsystem and the retrieval and browsing subsystem. XML is employed to integrate all the subsystems and make the data flow simply and naturally. Through the knowledge base, dynamic processing of the documents is achieved. In e-DOCPROS, a document consists of two parts [45]. Based on object-oriented concept [55], one is data of the document, which details the document itself. The second part is the actions to be applied to, or applied on the documents. The action is or will be assigned based on user-specified rules or the knowledge base; then the collaborative agents will perform the corresponding actions on it and finally, place the document in the complete state.

3.4 Accessibility

Information sharing is an essential feature of the information management system. Once a system supports multiple users, especially B2B business model, accessibility is the most important element in the system. Figure 3.2 shows that e-DOCPROS supports different user levels, such as organizations, groups and end users. For different user levels, the data access privileges are different. The organizational hierarchy is maintained in e-DOCPROS. The higher level users can access the lower level users' data, but not vice versa. For example, the manager of Group AA can access the User AAA's data. As an exception, the system may provide restrictions on data sharing, for example, email and other personal information in an end user's folder may not accessed by the upper level manager. Another way for the user to control his documents and folders is to assign

access privileges to the other users, upper level manager, or even users outside the organization. Those privileges are public and private. Public accessibility can be public in the group, in the organization or public to the whole e-DOCPROS system or public to the whole universe, which means that all users on the Internet can access the information. By default, the system is to make folders and documents private to the owner, and to direct or indirect upper level users.

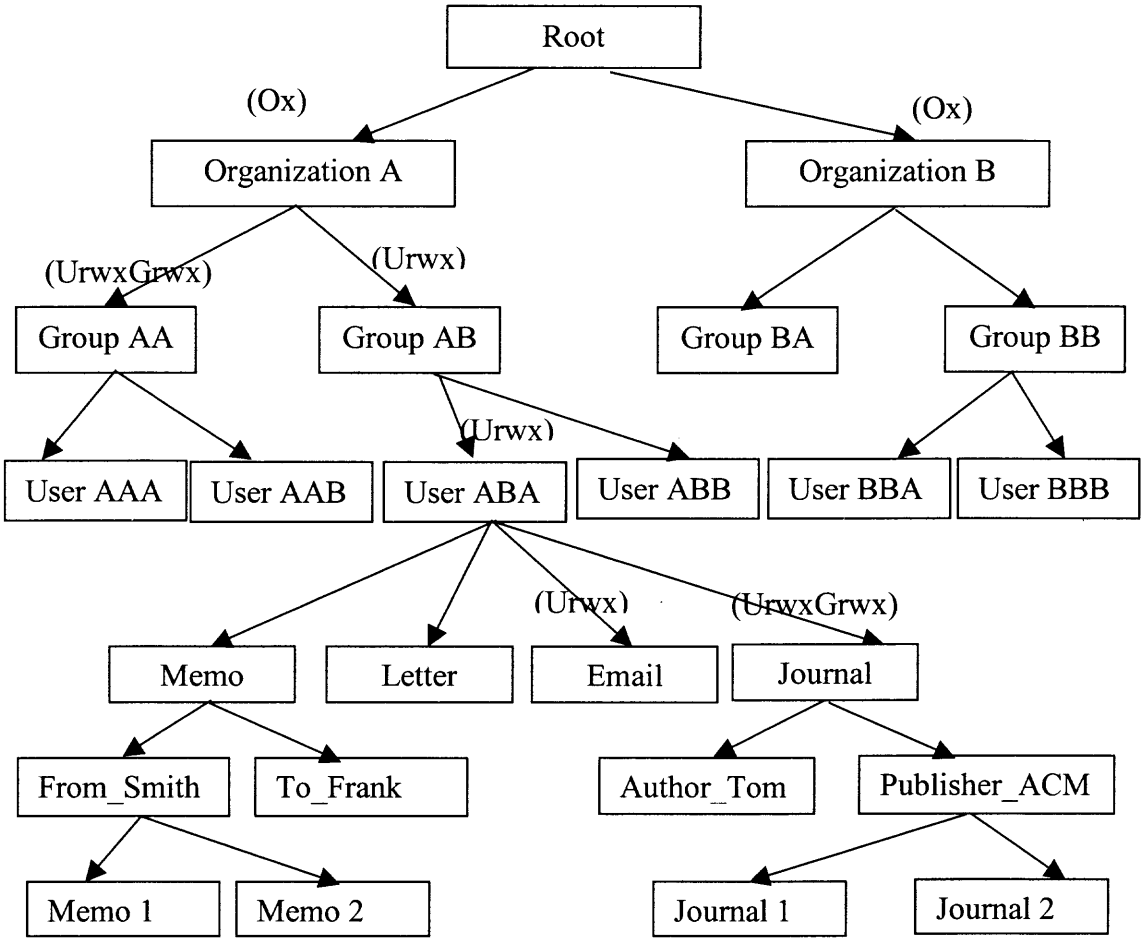


Figure 3.2 e-DOCPROS access control

Here, we define accessibility by different levels, denoted as U, G, O. U (User) is the owner of the document. G (Group) is the peer of the owner of the document in the same group. O (Other) represents the others. Here, r, w are read, write privilege to the

folder or document, x is the privilege to access the folder. Ur means that the owner has the read privilege to this folder. Uw means that the owner has the write privilege to this folder or document. Ux means that the owner can go into the folder. Grwx means that the information is shared between the same level of users within the same group, such as User AAA and User AAB, who can read, write to this folder or document. Figure 3.2 gives an example of e-DOCPROS data access control.

e-DOCPROS defines as follows the three levels of accessibility:

- Ownership: the creator of the document will be the default owner. In addition the system supports manual assignment of the ownership of a document to other users in the same organization, group or outside users based on the rules in the knowledge base. In some cases, the ownership will be promoted or demoted based on the scope of the document or the change of access list. For example, user A in Order department creates a document, and he will be the owner of the document. Once this document is directed to Finance department, user B in Finance department will fill out related financial information, but this kinds of information may be read only by the previous user, such as user A. Hence, the system automatically promotes the ownership of this document as the super user (or roles) of Order department and Finance department. The system reassigns the corresponding ownership or roles to those users involved in this document. The same will apply to the demotion of the ownership of a document.
- Role: in e-DOCPROS, we create roles and assign the corresponding accessibilities to those roles and associate the roles to users accordingly.

For different parts of the document, we can control access easily through roles. This is a nature way to achieve accessibility.

- View: for different users, even though they look at the same document, the different appearance will show to them according to their profiles. e-DOCPROS defines the different views on a document and provides access and trimmed document to different users.

All the three levels of accessibility are naturally integrate into the system by using XML. In the TEXPROS system, it cannot be obtained.

3.5 Scalability

Due to the diverse business models and the characteristics of Internet-based system, e-DOCPROS is a scalable system as well. e-DOCPROS is configurable and parameterized to achieve this goal.

- Hardware level: it includes servers and networking bandwidth.
- Software level: it includes two aspects. The first part is system software, such as operating system, system applications supporting hardware and networking. The second part is e-DOCPROS software applications, which is configurable and adaptive to the system workload. Also, the system can intelligently obtain knowledge regarding to the system and the workload and stores the knowledge in the knowledge base. It then provides report and suggestions if necessary.

3.6 Security

Security is always an important element in human life. Security is extremely important when we perform online business, such as B2B, B2C and C2C.

Most security threats on the Internet can be classified into one of the following four categories:

- Loss of data integrity;
- Loss of data privacy;
- Loss of services;
- Loss of control, in which case services are used by authorized person in an uncontrolled way.

Anti-virus is another hot topics for Internet-based system. e-DOCPROS allows user to upload any kind of documents onto the server, so a virus-free document or anti-virus software are vital to the system.

CHAPTER 4

e-DOCPROS MULTIPLE USER FOLDER ORGANIZATION

Since e-DOCPROS focuses on supporting multiple users over the Internet, even large number of users, a multiple user system is a must.

4.1 Requirements of Multiple User System

To support different types of business models, e-DOCPROS must be a multiple user system. For the B2B model, there must be data exchanges among organizations. Inside the organization, there are several types of users. For the B2C model, many users perform transactions with the organization through the Internet. The same is hold for the C2C and the Intranet models. For these reasons, a multiple user system is a must for e-Business system. For better document filing and retrieval, the system needs to identify the users, and grant the access privilege and control the access of the users.

Besides supporting multiple users, e-DOCPROS also needs the following three basic requirements from document filing point of view. The first is the need of a flexible and dynamic document model. It is impractical to build a personal document management system for a single user. In other words, e-Business systems will be used by different users. Different users understand and organize their documents in different ways. For greater retrieval efficiency and effectiveness, the system should be able to use all the knowledge the use has about the documents to be retrieved. But it is impossible for a single user to capture all the information that can be derived from the documents. Therefore, documents should be stored based on the user's knowledge about them.

This allows the system to match the user's knowledge about the documents to be retrieved against the descriptions (from the user's point of view) of the documents in the document base. This matching will become more difficult if the models are predefined, such as the case with most of the systems. As a result, it will be difficult for the user to specify queries because he understands documents in a different way. This causes many vague queries to be issued, and in turn reduces the efficiency and effectiveness of document and information retrieval. A predefined document model will also cause the system to be domain-dependent. It is hard for the system to be used by different users, such as e-Business systems over the Internet.

The second requirement is the need of document filing model. Document organization plays an important role in reducing the search space based on user provided information about the retrieved documents, rather than searching through the whole document space. Obviously, higher efficiency and effectiveness of document searches can be achieved when the user knows how the documents are organized. Therefore, a document filing model is useful to capture the user's domain knowledge of document organization. If we allow the user to define the document filing model, the system can organize the document the same way that the user would do in the physical world. As a result, the user is familiar with the document organization, and can provide helpful information to reduce the search space.

The third requirement is the system must be capable of supporting various information retrieval techniques. Today's office documents include various media types. And, the automatic semantic interpretation of some media types, such as images and videos, is far from a well-developed application. An e-Business document processing

system should provide a platform for various information retrieval techniques. Meanwhile the possibility of using various text-based information retrieval techniques for providing semantic content-based retrieval of multimedia document should also be investigated.

Through an examination of these requirements, e-DOCPROS needs to accommodate both levels of requirements. The first is that the system has the capability of automatic document filing and retrieval. The second is a multiple user system, in which the system can identify the users and/or organizations, and can define the interface between organizations and users, can control data access and information sharing.

4.2 The Triple Model

TEXPROS employs a dual model to classify an office document. This model is suitable for a personal document processing. When adapting it for e-Business era, e-DOCPROS must use a multiple user system over the Internet. In this new environment, the dual model is no longer applicable.

e-DOCPROS employs a flexible triple model approach to describe, classify, categorize, file and retrieve information over the Internet. The triple model consists of an organization type hierarchy, document type hierarchy and a folder organization, all of which can be defined by users. Organization type hierarchy (OTH) describes the structure of an organization, the management hierarchy and the way the organization manages the system and users. Document type hierarchy (DTH) describes the conceptual structure of documents. A folder organization is the real world structure to organize and store document in the system. e-DOCPROS uses Universal Resource Identifier (URI) to

identify and locate the information. In this way, the information can be identified uniquely and efficiently over the Internet. A URI (Universal Resource Identifier) is a string of characters which identifies a resource. It can come in one of two flavors: URL (Universal Resource Locator), or URN (Universal Resource Name) [82-84].

The triple model provides a flexible and dynamic modeling method since the user defines it. This allows the system to adapt to various application domains. In this way, a document can be described and stored as the user expects. Therefore, the user can specify queries based on whatever he knows about the document, such as the content, layout and logic structure, conceptual structure, and domain knowledge of the documents to be retrieved.

4.2.1 Organization Type Hierarchy

By using generalization and inheritance in an e-Business system, the organization classes are organized as the organization type hierarchy (OTH), which captures the structure of the organization and the hierarchy of the users and functional entities inside the organization. Each organization class is represented by an organization template, which describes the common properties in terms of attributes of the organization classes. Once the organization is created, the system obtains the user's specifications and generalizes an outline of the organization and each subdivision of the organization, even all the users, which is called organization template. The relationship between organization instances is parent-child relation, which defines the hierarchy of the organization, called the organization type hierarchy (OTH). Figure 4.1 to Figure 4.5 shows that the organization template and organization instances of two organizations within an e-DOCPROS system.

Figure 4.1 shows that there are two organizations in e-DOCPROS system, designated as Organization A and Organization B. For each organization, there are two groups, Group AA, Group AB, Group BA and Group BB. There are several users in each group, for example, Group AA has User AAA and User AAB. While the organization is created, e-DOCPROS generalizes the organization template based on the users' specification, then instantiates all folders pertaining to the subdivision and users inside the organization as organization instances.

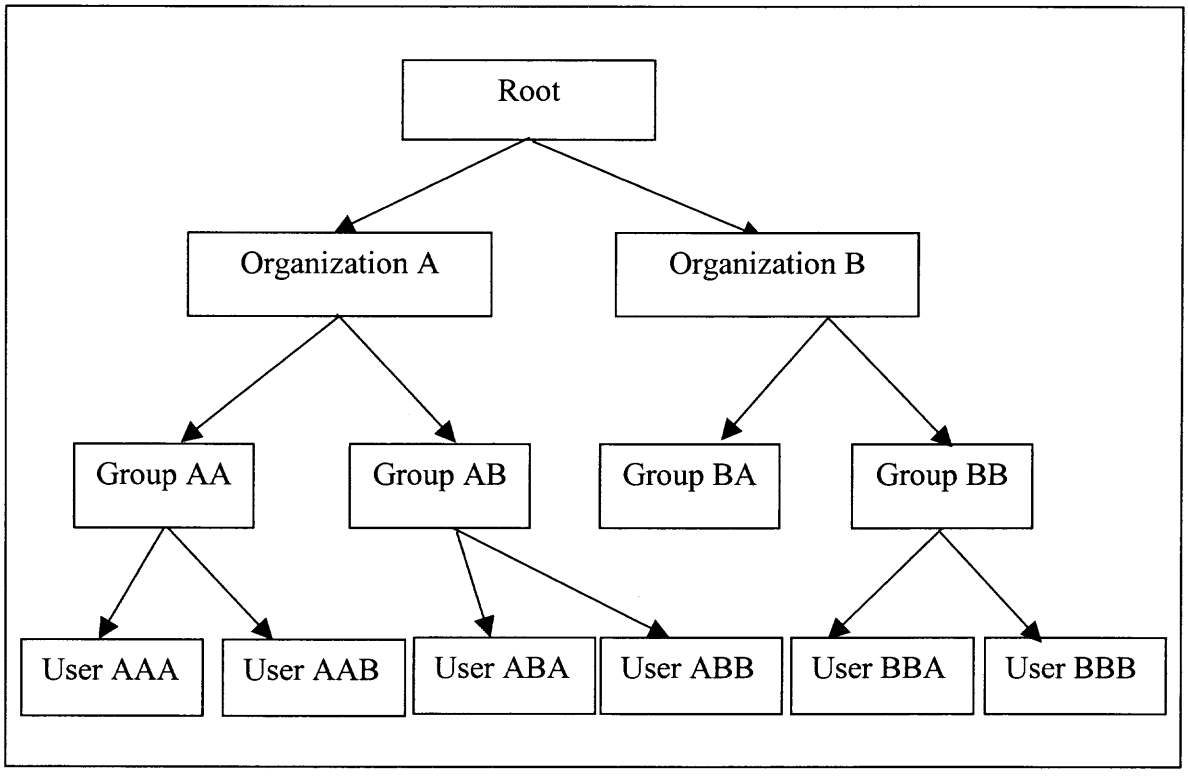


Figure 4.1 The structure of two organizations in e-DOCPROS

Figure 4.2 shows the organization template of the system. OI_ID presents organization instance ID, which uniquely identify the organization instance.

Parent_OI_ID represents the OI_ID of the parent organization instance. The attribute of Location is a compound one, consisting of two subcomponents. One is Universal Resource Identifier (URI), the other is the local network address. Figure 4.3 depicts the organization instance of ROOT, whose Parent_OI_ID is empty. This means that the whole e-DOCPROS only can have one ROOT organization instance, and all the other organizations are its children or descendants. Figure 4.4 describes the organization instance of Group AA, whose parent organization instance is Organization A (with Parent_OI_ID 5). Figure 4.5 shows the organization instance of User AAA, whose parent organization instance is Group AA. From these figures, the tree structure of Organization A is presented by those organization instances.

Attribute Name		Type
OT_ID		Integer
OT_Name | Type		String
[OT_Description]		String
OI_ID		Integer
Parent_OI_ID		Integer
Language		String
Date		Date
Location	URL	URL_String
	Local Address	String

Figure 4.2 The organization template

Attribute Name		Value
OT_ID		1001
OT_Name		ROOT
[OT_Description]		The root of e-DOCPROS
OI_ID		10001
Parent_OI_ID		NULL
Language		English
Date		10/18/2000
Location	URI	http://www.e-DOCPROS.COM/
	Local Address	//myserver/public_html/

Figure 4.3 The organization instance of ROOT

Attribute Name		Value
OT_ID		2001
OT_Name		Group AA
[OT_Description]		Group AA in Organization A
OI_ID		20010 (OI_ID of Group A)
Parent_OI_ID		20001 (OI_ID of Organization A)
Language		English
Date		10/20/2000
Location	URI	http://www.e-DOCPROS.COM /OrganizationA/GroupAA
	Local Address	//myserver/public_html /OrganizationA/GroupAA

Figure 4.4 The organization instance of Group AA

e-DOCPROS has the capability to keep track of all the users' specification and store the information in its knowledge base where it keeps a profile for each organization, down to all users. This kind of profile information can be used for later reference and dynamically to create and maintain the tree hierarchy of the organization. Also, the system can generalize common profile information as the default profile for the new organizations or new users.

Attribute Name		Value
OT_ID		3001
OT_Name		User AAA
[OT_Description]		User AAA in Group AA
OI_ID		30001 (OI_ID of User AAA)
Parent_OI_ID		20010 (OI_ID of Group AA)
Language		English
Date		10/28/2000
Location	URI	http://www.e-DOCPROS.COM /OrganizationA/GroupAA /UserAAA
	Local Address	//myserver/public_html /OrganizationA/GroupAA /UserAAA

Figure 4.5 The organization instance of User AAA

4.2.2 Document Type Hierarchy

In an e-Business system, there are many kinds of documents. By identifying the common properties for each document class, document can be categorized into different document classes. The frame template is employed to describer the common properties of each

document class. The relation between frame templates is parent-child, one-to-many relationships that depict the hierarchy of the document classes, called document type hierarchy (DTH) [49]. This means that a parent may have several children, and each child can have at most one parent. A child document type inherits all attributes from its parent.

Attribute Name		Type
FT_ID		Integer
FI_ID		Integer
Parent_FT_ID		Integer
Logo		String
Address	Street	String
	City	String
	ZipCode	String
	Country	String
DepartmentName		String
To		String
From		String
Date		Date
Subject		String
Content		String
CC		Email_Address

Figure 4.6 Frame template definition for memo type

A frame template is composed of a group of attributes. Each attribute may be of simple or composite type. For example, a frame template of memo type may consist of the attributes, From (sender), To (receiver), Date, Subject, Content, etc. as shown in Figure 4.6. The attribute of Address consists of Street, City, ZipCode and Country.

Figure 4.7 gives an example of frame instance of memo type. Sender can be further decomposed into FirstName, MiddleName and LastName attributes.

e-DOCPROS uses XML to present the frame templates, also frame instance. XML is detailed in Chapter 5.

Attribute Name		Value
FT_ID		101
FI_ID		1000
Parent_FT_ID		100
Logo		NJIT
Address	Street	123 Main Street
	City	Newark
	ZipCode	07102
	Country	USA
DepartmentName		CIS
To		All PH. D students
From		Director of PH.D program
Date		08/18/1998
Subject		Qualify Exams
Content		Coming
CC		faculty@cis.njit.edu

Figure 4.7 An example of memo type frame instance

4.2.3 Folder Organization

In e-DOCPROS, folders are heterogeneous repositories of organization instances and frame instances, which are called folder organizations. The folder organization is defined by a user and corresponds to the user's view of the document organization. Each folder

has a user-defined criterion for automatic document filing. A predicate-based representation of the documents is used to specifying criteria for folder organization [49].

4.2.4 The Relation Inside Triple Model

Figure 4.8 illustrates the relationship between the organization type hierarchy, the document type hierarchy and the folder organization. The organization type hierarchy is represented by the organization template. The document type hierarchy is represented by the frame template. Both the organization type hierarchy and the document type hierarchy are self-referenced, which means that they have a parent-child relationship inside each category. All instances of organization instances and frame instances are deposited into folder organization. Through folder organization, e-DOCPROS integrates organization and document together [54].

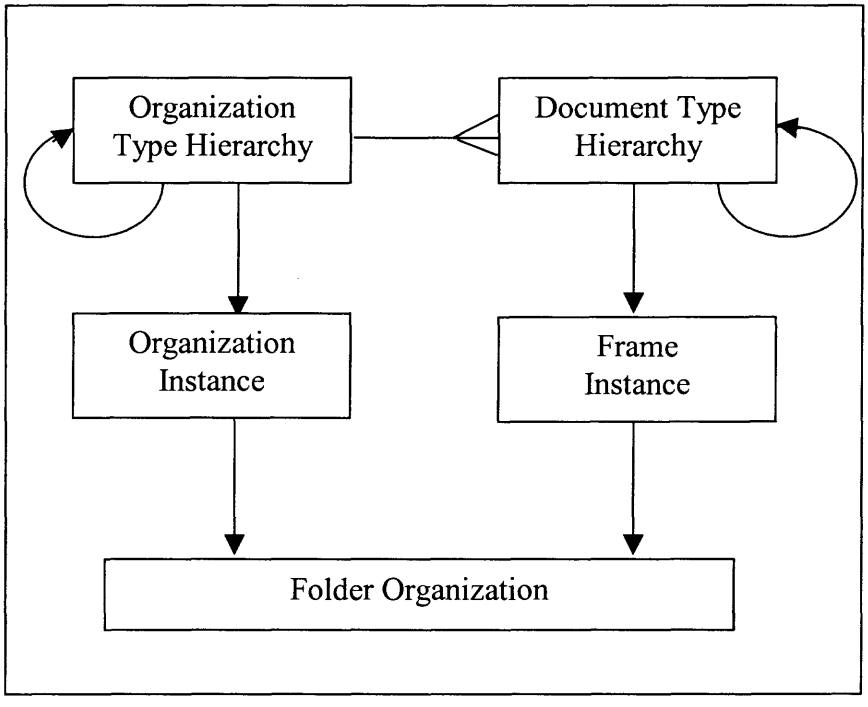


Figure 4.8 The relation inside the triple model

4.2.5 The Representations of Four Business Models

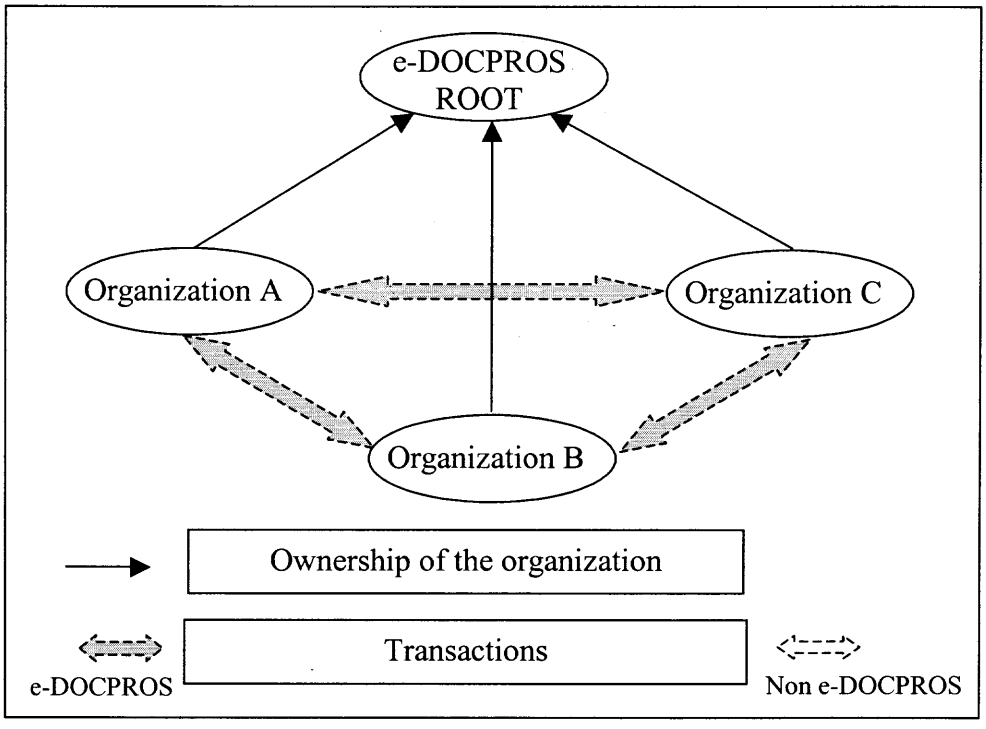
The triple model can easily supports all four business models as shown in Figure 4.9 to 4.11. Figure 4.9 illustrates the use of the triple model when applied to the B2B model. Figure 4.9 (a) gives a representation of B2B model in the same e-DOCPROS system. Figure 4.9 (b) depicts that B2B model conducted among two e-DOCPROS systems and other systems.

Figure 4.10 illustrates how B2C model works using the triple model. Figure 4.11 depicts the way the C2C model is supported by the e-DOCPROS. The triple model holds for the Intranet model even though transactions are performed through the organization's network. In addition, e-DOCPROS supports a combination of these business models.

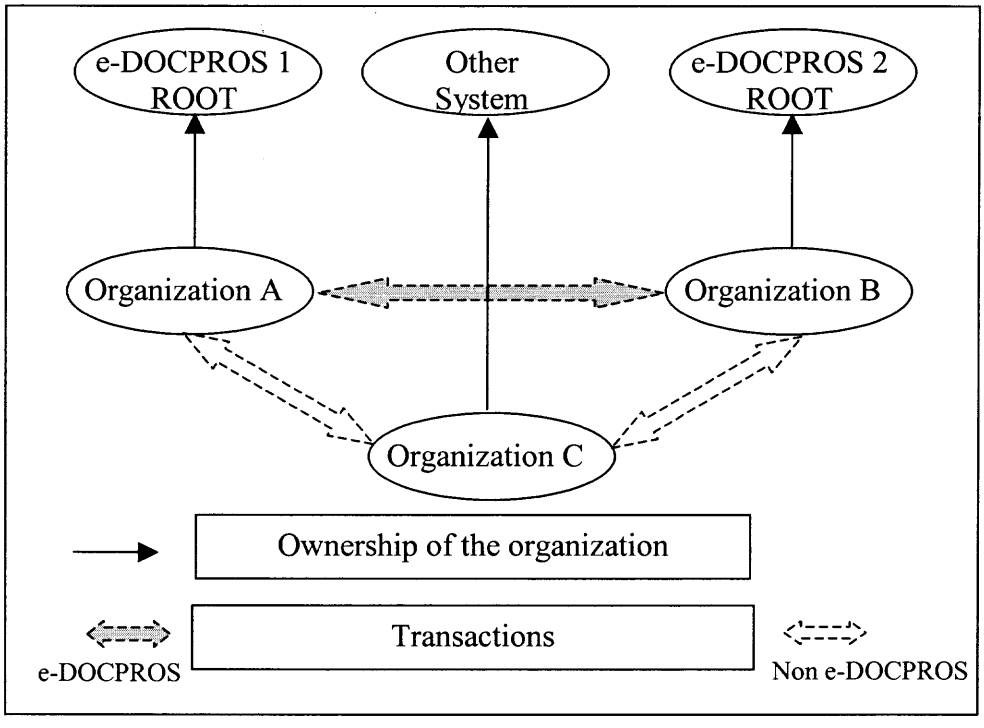
As we browse e-DOCPROS from the root and to the user level, each leave of the organization instances represents an individual user in the system. Within this folder, the user can create his folder structure and store documents there. Hence, if only one single user is viewed, the system appears very much like TEXPROS.

4.3 Information Sharing

The main reason we wish to transform TEXPROS into e-DOCPROS, an e-Business document processing system, is to achieve information sharing. e-DOCPROS supports all four business models, defining two kinds of information sharing. One is data exchange among more than one organization, which may be within one e-DOCPROS system, between two e-DOCPROS systems or between e-DOCPROS and the other systems. The second is information sharing between users within one e-DOCPROS system.



(a) B2B model in the same e-DOCPROS



(b) B2B model among different systems

Figure 4.9 B2B model represented by the triple model

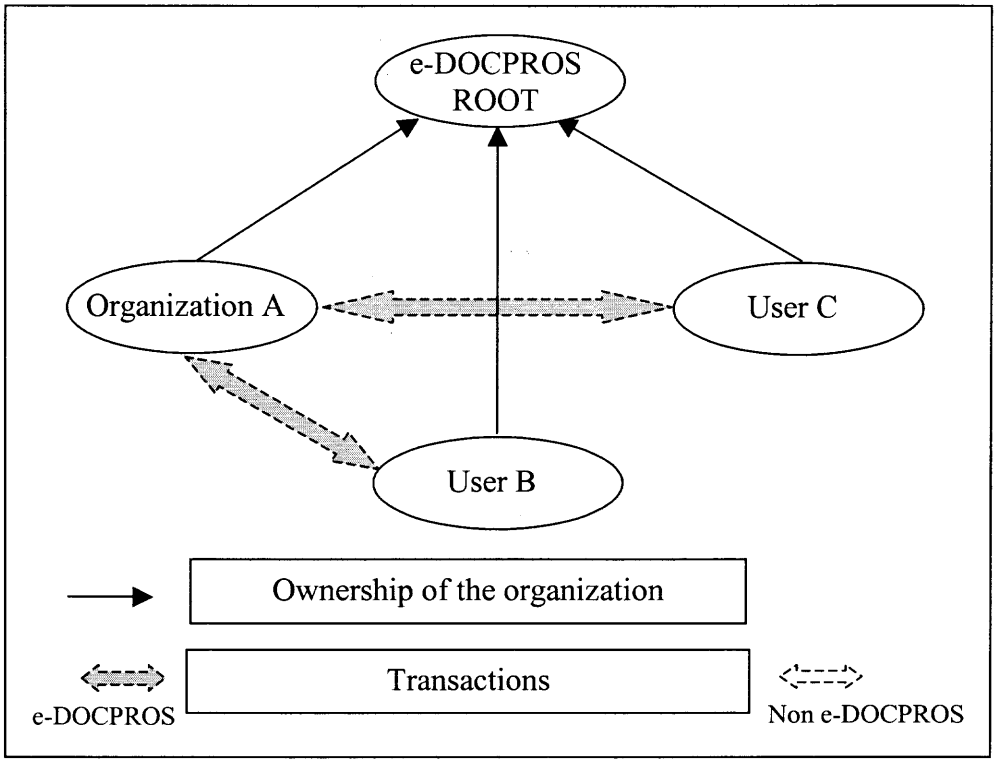


Figure 4.10 B2C model in the same e-DOCPROS

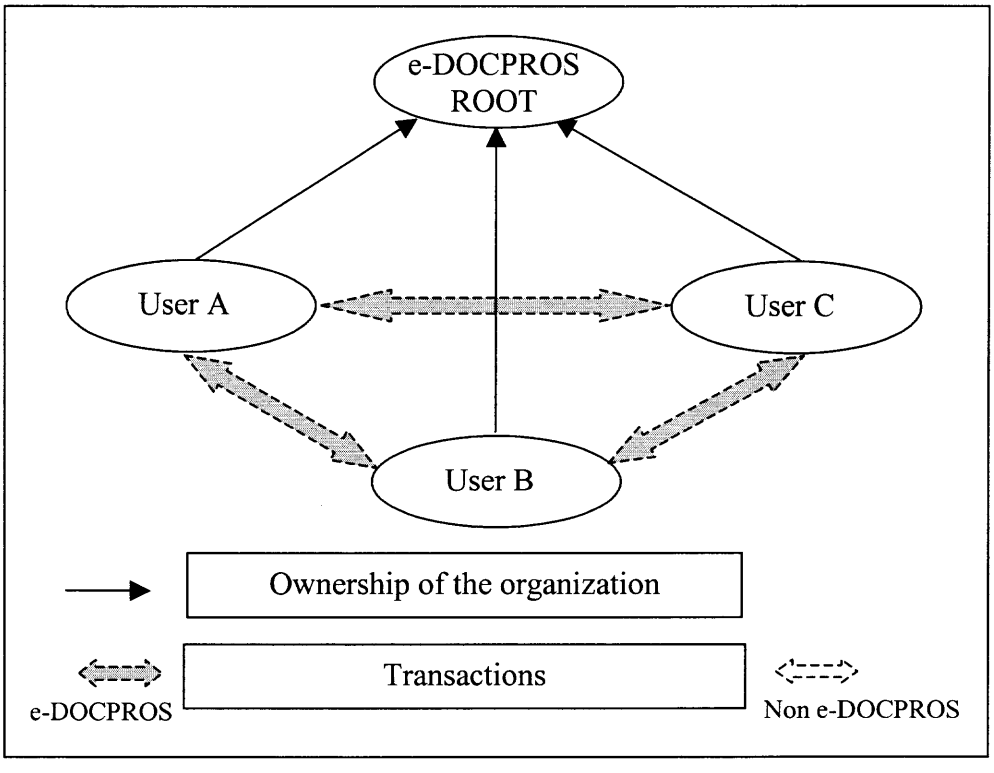


Figure 4.11 C2C model in e-DOCPROS

4.3.1 Information Sharing Among Users Inside One Organization

e-DOCPROS can acquire knowledge from users regarding to how information will be shared within the system. The default information sharing rule is that the manager of the user shares the user's information and information is shared with peers and the peers' managers.

Definition 4.1 (User) A user is an individual with basic privileges in an organization using the e-DOCPROS system.

Definition 4.2 (Manager) A manager is an individual who supervises users in an organization within e-DOCPROS system.

Definition 4.3 (Peer) User A's peers are users who are in the same group as User A.

The system can generate the peer list easily from organization instances.

e-DOCPROS supports those default information sharing rules, however the system assigns the access privilege only to the user himself, such as email folders and some other personal documents. Only the shared information defined by the group or group manager will be data shared. The information sharing rules are critical to collaborate design and concurrent document processing.

Figure 4.12 depicts information sharing between peers. Once a new document, Memo 2, is created by User AAA, the system categorizes it as a memo and extracts all related information from the original document, instantiates a frame instance of it. It will then be filed into the correct folder, To_Frank, by the agent-based file system. At the same time, the access control agent will assign access right of Memo2 to all peers. All the parent folders, such as Memo, To_Frank were created before Memo2 is filed, and already

assigned the correct access privileges. Hence, User AAB can access Memo 2 created by User AAA immediately.

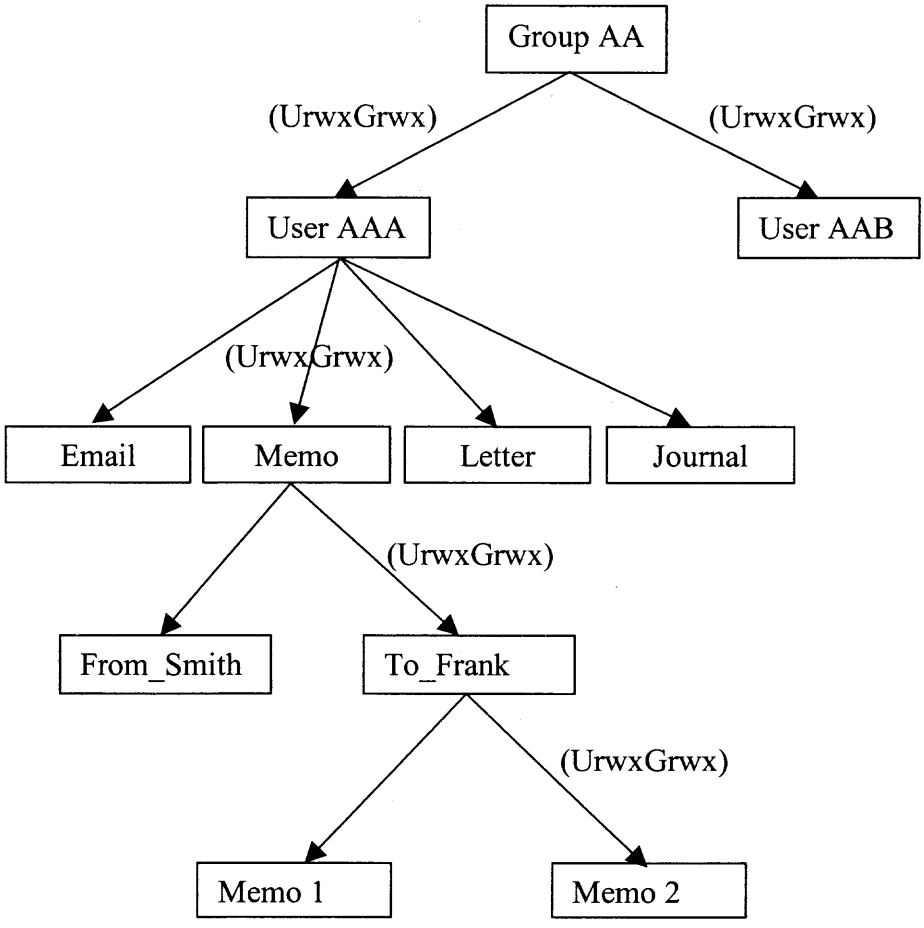


Figure 4.12 Information sharing among peers

Figure 4.13 depicts information shared between groups. Suppose there are two groups in Organization A, (Group AA and Group AB). Group AA has two users, User AAA and User AAB. Group AB has two users, User ABA and User ABB. The default the system will not allow data sharing between groups. However, it can be configured to allow information sharing between groups. For example, User AAB creates a document, which is a paper published in ACM Journal 2. The system will categorize it as a Journal

paper and file it into the folder for Journal 2 under Publisher_ACM, whose parent folder is Journal. The access control agent will assign access privileges for Paper 2 to Group AB; hence, all users under Group 2 will have access this document, such as User ABA and User ABB. The access rights include read access, write access and execute access. The system obtains this information through user interaction or from the knowledge base. Through these mechanisms, the system achieves information within the e-DOCPROS system.

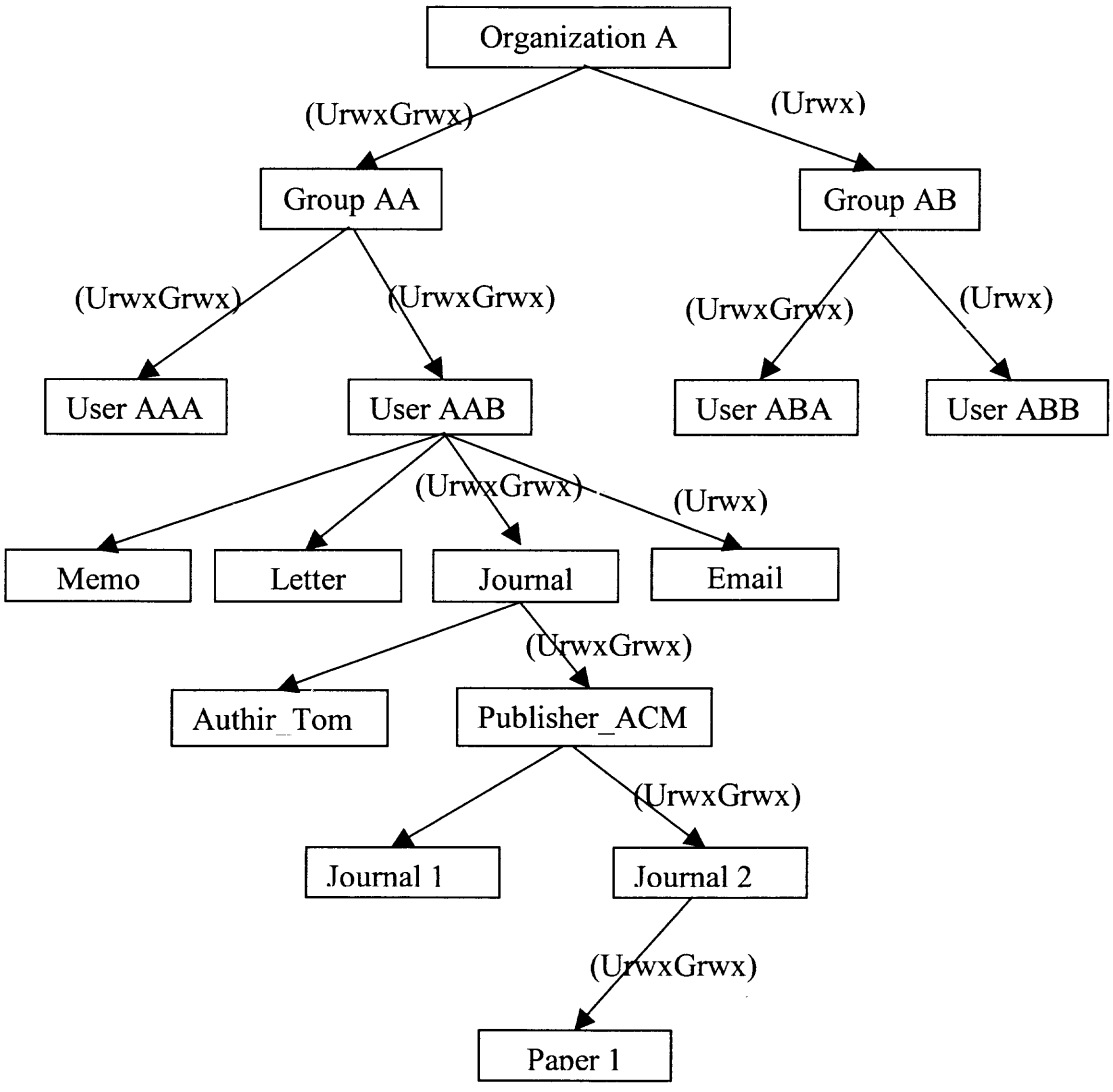


Figure 4.13 Information sharing between groups

Access control also includes the control to different parts of a document. e-DOCPROS has the capability to assign access status to individual parts of the document. A user can load the document into his workspace as read only or changeable. If one user opens the document as changeable, the other uses can only load it read only. While a user is changing the document, or portion of the document, e-DOCPROS assigns an access status flags to lock the document or the portion of the document that is changing. Through this mechanism, e-DOCPROS prevents the conflict of updating the same document or same portion of the document at the same time. Also, it makes collaborative design and document concurrent processing possible [95]. e-DOCPROS employees XML to achieve this goal easily.

4.3.2 Information Sharing Among Organizations

There are several types of information sharing among organization. One is information sharing between two organizations within the same e-DOCPROS system. Another is information sharing between organizations that are in the different e-DOCPROS systems. A third is information sharing between two organizations, in which one uses e-DOCPROS, the other uses another system.

Two approaches can be used to achieve information sharing between these two organizations. The first is an inbound-outbound folder solution. The second is a folder access control solution. Figure 4.14 illustrates an Inbound/Outbound queue solution. Each organization, for instance, Organization A and organization B, has two queues -- the inbound queue and the outbound queue. For documents created by User AA in Organization A, the data will be shared with User BA in Organization BA. Agent G1

first puts those documents into the Outbound queue of Organization A. Agent G0, which is in charge of Organization A's outbound queue, will replicate the document and place it into Organization B's inbound queue. Agent B2, which is in charge of organization B's inbound queue forwards the document to User BA in Organization B. At this point, User BA will be able to see the document. This mechanism makes a clean and secure way to perform information data sharing between organizations. What kinds of information will be data shared and how it will be shared is predefined by the two organizations. e-DOCPROS employs XML as the standard for information exchange, which is detail in Chapter 5.

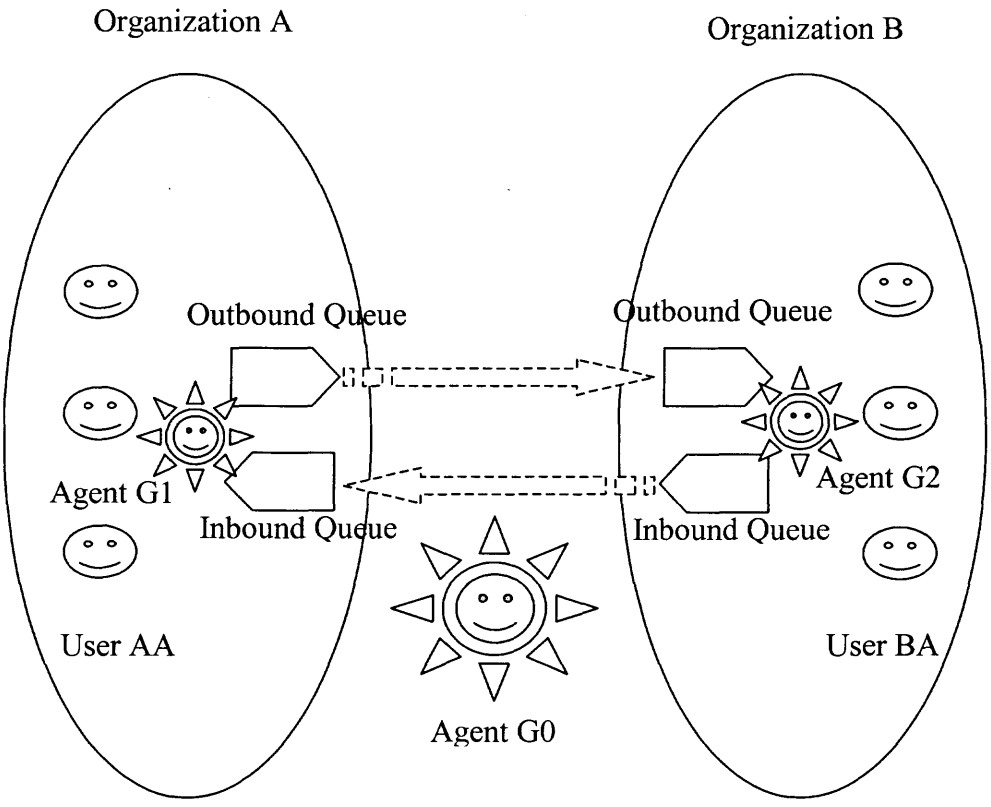


Figure 4.14 Agent-based Inbound/Outbound queue solution

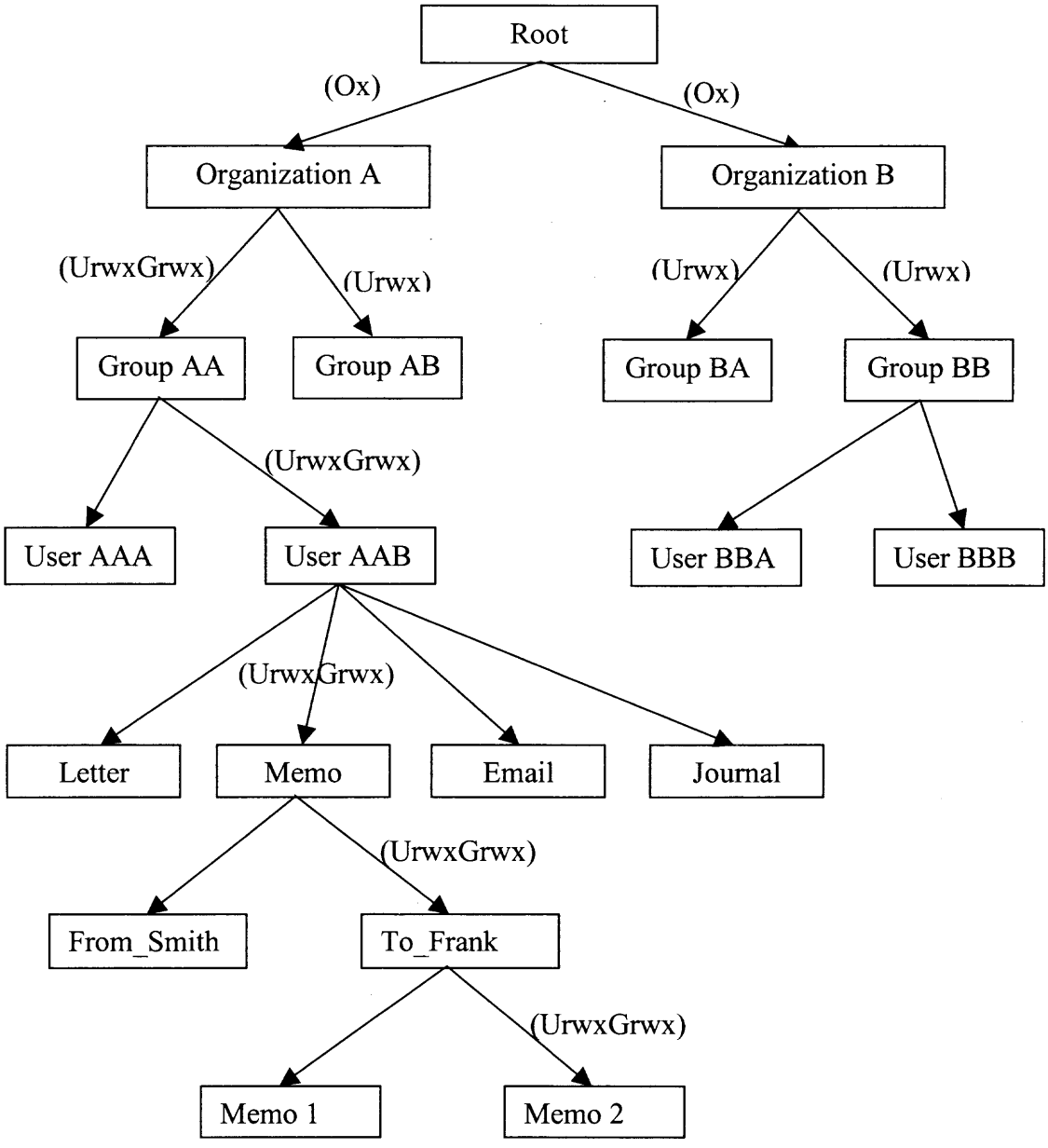


Figure 4.15 e-DOCPROS Access Control

Figure 4.15 illustrates an access control solution as an example of a second mechanism. This type of information sharing can only occur among organizations within the same e-DOCPROS system. The system assigns the correct access privilege to those folders and documents appropriate to the data sharing.

The inbound-outbound queue solution is more generic and is applicable to any kind of organizational configuration. For example, two organizations within the same e-DOCPROS system, or two organizations that each using their own e-DOCPROS systems, and organizations using an e-DOCPROS systems and another systems. However, the second mechanism is only applicable for those organizations that use the same e-DOCPROS system. Depending on the size and configuration of the system, e-DOCPROS system can utilize either or both mechanisms for information sharing among organizations.

4.4 Organization Maintenance and Reorganization

The structure and the size of an organization changes from time to time. The ability of e-DOCPROS to support an organization's maintenance and reorganization is more critical to the success of the system. e-DOCPROS allows an organization to add new users and new groups. Also, the system has the functionality to reorganize the hierarchy of an organization.

4.4.1 Organization Maintenance

e-DOCPROS learns from users how an organization is structured and what kinds of hierarchy the organization has. When a new group or department is created, the system can assign the default profile to it. If new users are added, the system can create folders for their user and assign default profiles for each of them based on the user specific setting or gets the default profiles from the knowledge base and assigns them to each user. If a user is promoted, the system will add privilege or access rights to him based the

knowledge base. On the other side, if a user is demoted, the corresponding privilege and access rights are removed from him. The system employs agents to perform those tasks.

4.4.2 Organization Reorganization

Along with the success of an organization, the expansion may be necessary. New groups are created and new users are added system, which was discussed above. Over time, an organization may reorganize its structure, for instance, split one group into two and separate their functions into two functional groups. The system has the capability to reassign and validate the hierarchy of the organization, its information access control and its information sharing rules.

There is a dedicated agent for each organization to check the hierarchy of the organization. Once a group is split into two groups, the agent will be invoked and restructure the hierarchy and maintain the integrity of the organizational structure. After the new hierarchy is created, all users involved are assigned to new groups. For each user, there is an agent to assign the correct privileges and access control based on the information sharing rule or organization rules. These agents assure the system's integrity. Periodically, some agents will globally check the integrity of the system and maintain its integrity.

CHAPTER 5

e-DOCPROS REPRESENTATION AND DATA EXCHANGE STANDARD

There are two kinds of requirements that determine data exchange standards in e-DOCPROS. First, e-DOCPROS supports the B2B business model, in which data is exchanged among organizations; hence, a data exchange standard across all the partners is required. Second, within e-DOCPROS, among all of its subsystems, a data exchange standard also is required to define the document objects and transform the messages in the processing flow. In this dissertation, we combine these two types of requirements by using XML.

5.1 Extensible Markup Language

XML (Extensible Markup Language) is the next generation of Web description language, which has many advantages over HTML (Hyper Text Markup Language) [52,53].

5.1.1 The Evolution of Markup Languages

A document specified in an open structure can be easily manipulated and exchanged. The benefit is that we can use ubiquitous software (such as, Web browser) to process and manipulate open structured documents.

In the early 1960, the Graphic Communication Association (GCS) create the GenCode to develop generic typesetting codes for dealing with heterogeneous vendors with different type data typesets. IBM introduced the Generalized Markup Language (GML) for generalizing the generic document types for solving problems that are machine and application dependent. The GML is a high level language for formatting

documents of different types. The GML uses the tags for specifying layout instructions. In the early 1980, the GenCode and GML were combined to form a standard for processing textual documents, in attempt to unify methods for defining, specifying and using document markup [51].

In 1986, the Standard Generalized Markup Language (SGML) [56] has promised to make documents modular and interchangeable. The two key elements of SGML are its syntax and semantics rules. The syntax rules are based on the IBM's GML syntax styles. The semantic rules are derived from the typesetting rules of the GCS. SGML is a markup language for describing other languages. What SGML does do, however, is describing the relation of components within a document. SGML is used extensively in publishing.

HTML stands for Hyper Text Markup Language and is the publishing language of the World Wide Web [58, 59, 60, 61]. HTML was created based on a small set of SGML's markup concept. HTML has a common tag set for presenting data, but does not give a description of the semantic meanings of the data. It does not have certain features, such as the extensible feature, the semantic structure, and the data validation, which were provided by SGML. HTML is still used on the Internet because it is simple to create, and almost totally platform and viewer independent. HTML is a markup language that tells the clients, in general terms, how the information should be presented. However, HTML is not useful for describing the data with semantic information and data structure efficiently and consistently over the network.

XML stands for Extensible Markup Language and was derived from SGML [52,53]. In 1998, the World Wide Web Consortium (W3C) recommended the use of the Extensible Markup Language (XML). The data in XML is self-descriptive, which means

that each data has a self-schema to describe data. XML is a standardized text format designed specially for transmitting structured data over Internet applications.

A markup language is a tagging system that can be used to disassemble a document's structure in a friendly, but platform-independent way. A document is broken down before it is transmitted electronically from one location to another location, and then reassembled upon the arrival of the disassembled components at a new location. The tagging system is extensible in the sense that it provides users with a mechanism for creating their own ontology.

5.1.2 Advantage of XML over HTML

Here are some reasons why we need to switch from a HTML-based document world to one based on XML-based documents.

- Browser Presentation:

  XML can provide more and better facilities for browser presentation and performance through the use of style sheets.

- Information Accessibility:

  Information is more accessible and reusable due to the flexibility of XML.

- Richer Content:

  Through the use of new markup elements it is possible to create richer content that is easier to use.

- SGML Compatibility:

  Since XML files are compatible with the SGML standards, they can be also used outside the Web in an SGML environment.

- Tailored Document Types:

  Document providers and authors are able to create their own document types using XML and are not restricted to the set of markup elements in HTML. It is also possible to invent new markup elements.

5.1.3 Advantage of XML over EDI

EDI (Electronic Data Interchange) is a standard for the inter-organizational computer-to-computer exchange of structure information. EDI has been with us for many years, but it has not reached its full potential, since it has very little support outside the major supply-chain driven industries such as automobile manufactures. In concept, EDI is a way of delivering business documents electronically. It does this by providing coding and transport mechanisms. The coding used has a delimited format, in which each field is separated by a reserved character. There are two basic standards for EDI – ANSI X.12 used in North America, and EDIFACT used through most of the rest of the world. Those messages are carried over Value Added Networks (VANs) between the trading parties. These are managed networks, providing a secured and robust service at a high cost. The Internet, by contract, currently provides a less secure, less robust service but at low cost. Of course, much work is being done to increase the security and robustness of the Internet, but it does not yet compare to a VAN. However, one EDI problem stems from its delimited nature of the coding. The message itself is not self-describing, the format of the message is rigid and all trading parties must agree upon this rigid data exchange format. The result of this is that EDI messages must be 100% compatible in a structure it

understands. If we want to use EDI across industries, the EDI message must be a superset of all the possible requirements [52].

EDI was a good system, but it needs a large investment for every participating company, making it unavailable to smaller businesses. The costs for EDI were high because there were repeated costs for every installation. For example, a company trading with 50 partners has to install 50 EDI interfaces. EDI is also a two-way protocol, which was fine in client-server area, but in the emerging pervasive computing environment, EDI in its traditional form will not survive. There are efforts underway to move EDI to the Internet [1, 62].

XML is also suitable for data exchange. B2B business model requires data exchange between organizations. To accomplish this, XML has many advantages over EDI. Combining these facts together, XML can be employed as a Web description language and as a standard for data exchange, which is the most attractive part of XML. e-DOCPROS employs XML as the document definition language, which integrates document definition and document dynamic processing together.

5.2 Document Object Model

Document Object Model (DOM) is a model in which a document (such as a Web page, or a memo) contains objects (e.g. text elements, images, links) that can be manipulated, which means that we have the capability not only to access it, but to change or add to it. The DOM provides a means for working with XML documents (and other types of documents) through the use of codes, and a way to interface with that code in the programs. For example, the DOM enables us to create documents and parts of

documents, navigate through documents, move, copy and remove parts of documents, as well as add or modify attributes.

DOM has been recommended by World Wide Web Consortium (W3C) [45]. Using DOM it is possible to remove, alter or add an element to a given document. It is also possible to change the content of an element or remove, alter or add an attribute. Through DOM, it is possible to get specific information of the properties of a document.

For example, we have a <order>XML below.

```
<?xml version="1.0"?>
<order number="10001">
  <date> 08/18/2000</date>
  <customer id="8001">Company XYZ</customer>
  <item>
    <part-number warehouse="Warehouse 100">CD101-08A<part-number>
    <description>CD Exchanger</descrition>
    <quantity>18</quantity>
  </item>
<order>
```

Figure 5.1 describes the structure of the object model of an order XML. Some of the elements have been made into objects, as shown by the shaded boxes, and some have been made into properties of those objects, as shown by the white boxes. If we write code to deal with an order, this object model would make it easier to process that information, and would probably even include methods to provide some functionality for us.

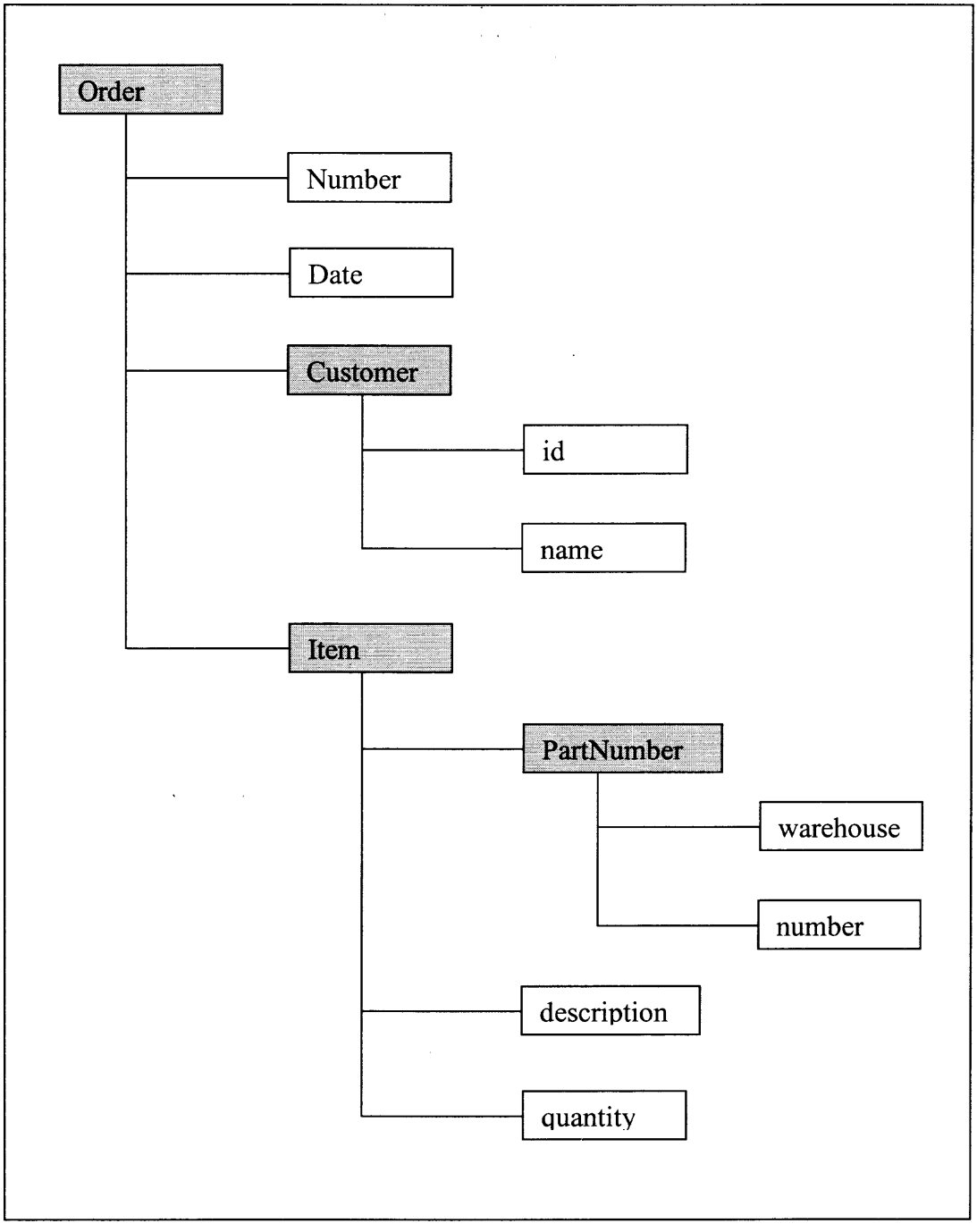


Figure 5.1 The object model of an order XML document

The Document Object Model (DOM) takes a generic approach to model any XML document, regardless of how it is structured. Figure 5.2 depicts the architecture of

the XML DOM. The DOM is usually added as a layer between the XML parser and the application that needs the information in the document, meaning that the parser reads the data from the XML document and then feeds the data into a DOM. The DOM is then used by a higher-level application. The application can do whatever it wants with this information, including putting it into another proprietary object model, if so desired [52].

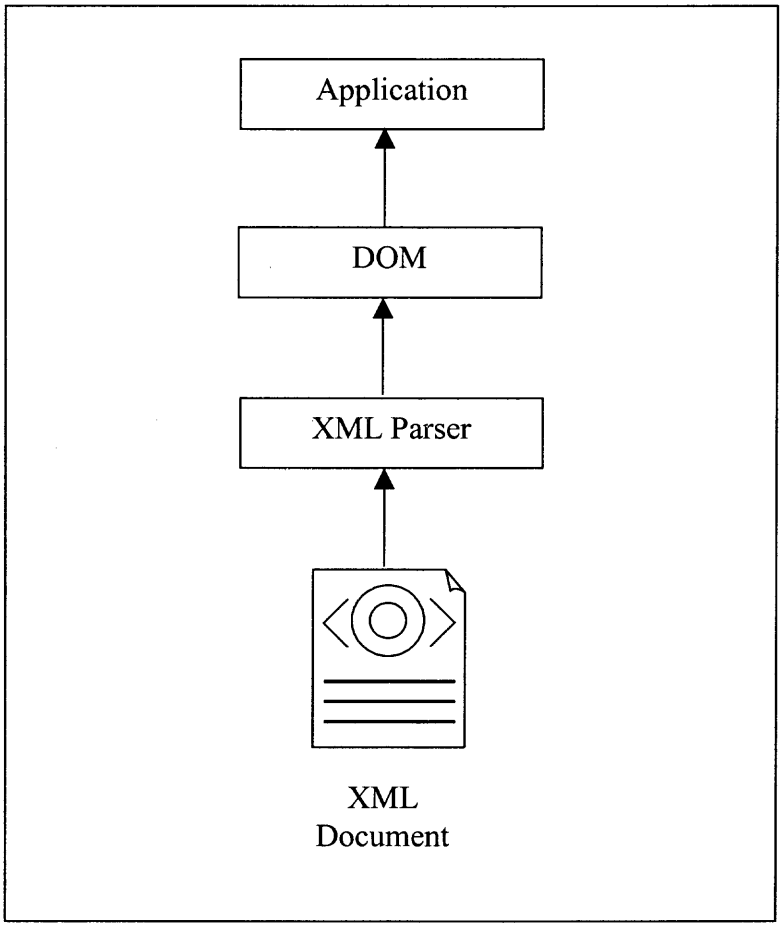


Figure 5.2 The Architecture of the XML DOM

e-DOCPROS defines the document based on DOM. It encapsulates the data of the document and provides the interface to manipulate the data by using OOA/OOD technology [55].

5.3 Document Type Definitions

Document type definitions (DTDs) are used to specify a set of rules to define the way XML documents should be structured. Being able to define such rules will become more important as we exchange, process, and display XML documents in a wider environment, such as in a business-to-business or e-commerce scenario. Using DTDs will allow us not only to assure XML documents follow the syntax rules of the XML specification, but also that they follow our own rules regarding content and structure.

5.3.1 Well-Formed XML Document

Well-formed XML documents are those that comply with basic syntax and structural rules of the XML 1.0 specification. Those rules include the rule that every start-tag must have an end-tag, such as <string of tag> and </sting of tag> should always be matched pair; tags cannot be overlapped, which means that the child elements should be closed before the parent element is closed. For example, the following statement is not a well-formed XML statement.

<name>

<firstName>John<lastName>Smith</firstNmae></lastNname>

</name>

We have <name>tag, which has <firstNmae> and <lastName> child tags. We have to close <firstName> by using </firstName>, then we can close <lastName> tag by using </lastName>, then close the parent tag of <name> by using </name>. The well-formed XML statement should be written as the follows:

<name>

<firstName>John<firstName>

<lastNmae>Smith</lastName>

</name>.

Another rule of a well-formed XML document is that an XML document can have only one root element. In the above <name> document, the <name> element is called the root element. This is the top-level element in the document, and all the other elements are its children or descendants. An XML document must have one and only one root element: in fact, it must have a root element even if it has no content.

Any document that is well formed is easier to use and reuse. The well-formed documents can be transmitted electronically through the network from one location to the other and reassembled efficiently and accurately at the receiving locations. A well-formed document provides the mechanism for specifying each data's boundary. The XML parser is able to identify efficiently the beginning tag and ending tag and extracting the data between tags [52].

5.3.2 Document Type Definition

Separating the XML data description from individual applications allows all cooperating applications to share a single description of the data, know as the XML vocabulary. A group of XML documents that share a common XML vocabulary is known as a frame template (document type), and each individual document that conforms to a frame template is a frame instance (document instance). This is similar to the basic principle of object-oriented (O-O) concepts [55], in which objects are grouped and described as an

object class (compare to a frame template), each individual object conforming to that class description is known as an instance of that object (compare to the frame instance).

DTDs are XML documents that can either be incorporated within the XML document containing data, or exist as a separate document. They define the rules that set out how a document should be structured, what elements should be included, what kind of data may be included and what default values to use. Furthermore, the rules of the vocabulary description in the DTD, and how those rules are applied by a validating parser, are well-defined and standardized. This alone can go a long way toward a more reliable data exchange, particularly between diverse business partners. It is no longer necessary for each partner to create personalized customer tools – both partners can use the same standard XML tools and technologies to handle their shared data vocabulary. This is the most important characteristic of the XML as a data exchange standard.

DTD is used to validate XML documents. Valid XML documents are well-formed documents that also comply with syntax, structure, and other rules as defined in DTD. Multiple documents and applications can share DTDs. Having a central description of the XML data and a standardized validation method lets us move both data description and validation code out of numerous individual applications. The data description code becomes the DTD, and the validation code is already present (and optimized) in the validating XML parser. This greatly simplifies our application code, thus improving both performance and reliability. Having standardized XML vocabularies for common documents makes DTDs reusable. Shared DTDs are the foundation of XML data interchange and reuse.

e-DOCPROS XML uses DTDs to define document types, which is presented in Extended Backus Naur Form (EBNF), a formal language for syntax description. We use cardinality operators to indicate how many times an element may occur. There are in total four cardinality operators, which are [none], [?], [*] and [+]. The absence of a cardinality operator character, [none], indicates that one, and only one, instance of the child element is allowed (required). The notation [?] indicates that zero or one child element is allowed (optional singular element). The notation [*] indicates that zero or more child elements are allowed (optional elements). The notation [+] indicates one or more child elements are allowed (required elements). For example, in the following <PersonName> document, the absence of cardinality operators for <FirstName> and <LastName> means that there must always be exactly one of each of these child elements in every <PersonName>. Some person may not have a middle name, and some persons may have more than one middle name. Also, the title is optional, there may be one title or there is none.

```
<!ELEMENT PersonName
  (
      (Mr|Ms|Dr|Rev)?, FirstName, MiddleName*, LastName, (Jr|Sr|III)?
  )
>
```

5.4 The XML Representation of the Triple Model

e-DOCPROS employs a flexible triple model approach to describe, classify, categorize, file and retrieve information over the Internet. The triple model consists of an organization type hierarchy, a document type hierarchy and a folder organization, all of

which can be defined by users. e-DOCPROS employs XML to define and describe all objects in the triple model, which includes organization type hierarchy (OTH), document type hierarchy (DTH) and folder organization. In this way, eDOCPROS-XML integrates all subsystems and provides a standard of information exchange between organizations and among the subsystems.

5.4.1 Semistructured Data

The data in a computer system can be represented in different forms, either in an unstructured form of data in a plain ASCII file system or in a highly structured form of data in a relational database system. These data could be raw, such as the byte stream of sound or image. These data could be structured so that they can be stored in a relational database system with rigid types. Some structured data such as MS Word documents or HTML documents are only used for presentation purpose [51].

In an e-Business environment, documents are processed over the Internet with multiple document repositories. Those documents differ from traditional relational or object-oriented data. However, they are more like the semistructured data model [64-69]. Semistructured data are neither raw nor strictly typed. In another words, they are neither table-oriented as in a relational model nor sorted-graph type as in an object database. Semistructured data are generally described as a format that does not conform to a rigid schema. The data are represented by a collection of atomic or complex objects. The value of an atomic object is a basic type, such as integer, string, date and float. For example, "20", "Joe Smith", "10/18/1999" and "12.5" are atomic objects. The values of complex object are sets of (attribute, value) pairs [52]. For example, a memo can be described as

Memo:

{

Logo: "NJIT"

Department: "CIS"

From: "Director of Ph.D Program"

To: "Ph.D Students"

To: "Faculty"

Date: {

Month: "08"

Date: "18"

Year: "1998"

}

Subject: "Qualify Exams"

Content: "Coming soon"

CC: "Chairperson of CIS"

}

This structure has some characteristics: repeatable attributes, composite attributes and dynamic changing attributes. The repeatable characteristic means that attributes can be defined to appear multiple times. For example, the memo object can have multiple "To" attributes. The composite attribute means that the attributes can be further divided into smaller elements, such as the data attributes of the memo object containing month, date and year attributes. The dynamic changing attributes means that we can add or delete

some attributes. For example, the memo object can add additional (attribute, value) pairs, such as To: "Dean of Engineering School" or delete CC attributes.

Semistructured data are beyond the relational data barrier, since their specifications are quite flexible. For example, the semistructured data specification allows multiple occurrences of an attribute, composite attributes, the same attribute name of different types in different objects and so on. Frame instances are semistructured data, which can have an irregular structure. e-DOCPROS employs XML to represent the frame instance.

5.4.2 The XML Representation of The Organization Type Hierarchy

In an e-DOCPROS system, the organization classes are organized as the organization type hierarchy (OTH), which captures the structure of the organization and the hierarchy of the users and functional entities inside the organization. Each organization class is represented by an organization template, which describes the common properties in terms of the attributes of the organization classes. Once the organization is created, the system will obtain the user's specifications and generalize a synopsis for the organization and each subdivision of the organization, including all of the users, which is called organization template. The relationship between organization instances is the parent-child relation, which defines the hierarchy of the organization, called organization type hierarchy.

There are three types of organization templates – the root, the organization and the user organization template, respectively. An element declaration is used to define a

new element and specify its allowed content. Each organization type is defined as an element in eDOCPROS-XML.

```
<!-- ======= The DTD of Root Type of Organization Template ======= -->
<!ELEMENT  Root (OT_ID, (OT_Name|Type), OT_Description?, OI_ID,
        Parent_OI_ID, Language, Date, Location+ >
<!ELEMENT  OT_ID                    (#PCDATA) >
<!ATTLIST  Root  OT_Name            CDATA  #FIXED "ROOT" >
<!ATTLIST  Root  Type               CDATA  #FIXED "ROOT" >
<!ELEMENT  OT_Description           (#PCDATA) >
<!ELEMENT  OI_ID                    (#PCDATA) >
<!ATTLIST  Root  Parent_IO_ID       CDATA  #FIXED "" >
<!ELEMENT  Language                 (#PCDATA) >
<!ELEMENT  Date                     (#PCDATA) >
<!ELEMENT  Location                 (#PCDATA) >
<!-- ======= END of the DTD of Organization Template  ======= -->
```

Figure 5.3 The DTD of root type of organization template

Figure 5.3 depicts the DTD the of root organization template. The "<!ELEMENT Root (OT_ID, (OT_Name|Type), OT_Description?, OI_ID?, Parent_OI_ID, Language, Date, Location+ >" is the root element declaration of the root organization template. This document type contains several child elements, such as OT_ID (Organization template id), OT_name (Organization Template name) or type, OT_description, OI_ID (Organization Instance id), Parent_OI_ID, Language, Date and Location. These elements

must appear in the given sequence defined in the DTD. The notation [?] followd by OT_Description means that OT_Description can have zero or one occurrences. The notation [+] means that Location can have one or more than one occurrence, but at least one. The OrganizationTemplate element is defined to specify the corresponding e-DOCPROS's organization template. For the Root organization template, only one root exists in the system and its parent_oi_id is null. Here ATTLIST (Attribute List Declaration) combined with #FIXED keyword with a default value of NULL is used to invoke this rule.

```
<!-- ======= The DTD of Organization Type of Template Organization ======= -->
<!ELEMENT Organization (OT_ID, (OT_Name|Type), OT_Description?, OI_ID,
        Parent_OI_ID, Language, Date, Location+ >
<!ELEMENT OT_ID                 (#PCDATA) >
<!ATTLIST Root OT_Name          CDATA #FIXED "ORGANIZATION" >
<!ATTLIST Root Type             CDATA #FIXED "ORGANIZATION" >
<!ELEMENT OT_Description        (#PCDATA) >
<!ELEMENT OI_ID                 (#PCDATA) >
<!ELEMENT Parent_IO_ID          (#PCDATA) >
<!ELEMENT Language              (#PCDATA) >
<!ELEMENT Date                  (#PCDATA) >
<!ELEMENT Location              (#PCDATA) >
<!-- ======= END of the DTD of Organization Template ======= -->
```

Figure 5.4 The DTD of organization type of organization template

Figure 5.4 describes the DTD of organization type of organization template. The content of <Type> and <OT_Name> has the default value of "ORGANIZATION". Also, all organizational instances must have a parent_oi_id.

<!-- ======= The DTD of User Type of Organization Template ======= -->
<!ELEMENT Organization (OT_ID, (OT_Name|Type), OT_Description?, OI_ID, Parent_OI_ID, Language, Date, Location+ >
<!ELEMENT OT_ID (#PCDATA) >
<!ATTLIST Root OT_Name CDATA #FIXED "USER" >
<!ATTLIST Root OT_Name CDATA #FIXED "USER" >
<!ELEMENT OT_Description (#PCDATA) >
<!ELEMENT OI_ID (#PCDATA) >
<!ELEMENT Parent_IO_ID (#PCDATA) >
<!ELEMENT Language (#PCDATA) >
<!ELEMENT Date (#PCDATA) >
<!ELEMENT Location (#PCDATA) >
<!-- ======= END of the DTD of Organization Template ======= -->

Figure 5.5 The DTD of user type of organization template

Figure 5.5 depicts the DTD of a user type of organization template. The content of <Type> and <OT_Name> has the default value of "USER". Also, all organization instances must have one and only one parent_oi_id.

Through the definition of those organizational templates, the system can maintain a consistent tree structure of the organization. Figure 5.6 – Figure 5.8 give three examples of organizational instances.

```
<eDOCPROS-XML>
<?xml-stylesheet type="text/css" href="Root.css"?>
<!DOCTYPE Root SYSTEM "http://www.e-DOCPROS.com/OT_Root.dtd">
    <Root OID="&1">
        <OT_ID OID="&2">1001</OT_Name>
        <Type OID="&3">ROOT</Type>
        <OT_Description OID="&4">Root of e-DOCPROS system</OT_Description>
        <OI_ID OID="&5">10001</OI_ID>
        <Parent_IO_ID OID="&6"></Parent_IO_ID>
        <Language OID="&7">English</Language>
        <Date OID="&8">10/18/1999</Date>
        <Location OID="&9">http://www.e-DOCPROS.com/</Location>
        <Location OID="&10">//myserver/public_html</Location>
    </Root>
</eDOCPROS-XML>
```

Figure 5.6 The organization instance of Root

All organizational instances and frame instances in e-DOCPROS are represented by using eDOCPROS-XML, which begins with <eDOCPROS-XML> and ends with

</eDOCPROS-XML>. For each instance, it usually includes two other entities, one is CSS (Cascading Style Sheets), the other id DTD (Document Type Definition).

```
<eDOCPROS-XML>
<?xml-stylesheet type="text/css" href="Organization.css"?>
<!DOCTYPE Organization SYSTEM "http://www.e-DOCPROS.com/OT_Organization.dtd">
    <Organization OID="&1">
        <OT_ID OID="&2">2001</OT_Name>
        <Type  OID="&3">ORGANIZATION</Type>
        <OT_Description OID="&4">The folder of Organization A</OT_Description>
        <OI_ID OID="&5">20001</OI_ID>
        <Parent_IO_ID OID="&6">10001</Parent_IO_ID>
        <Language OID="&7">English</Language>
        <Date OID="&8">12/18/1999</Date>
        <Location OID="&9">http://www.e-DOCPROS.com/OrganizationA</Location>
        <Location OID="&10">//myserver/public_html/OrganizationA</Location>
    </Organization>
</eDOCPROS-XML>
```

Figure 5.7 An example of Organization organization instance

XML is more concerned with the structure of the document, not its presentation. XML, whether in data or document format, still often needs some mechanism to display the tags in a presentation that communicates the intentions of the tags more efficiently to

the viewers. The Cascading Style Sheets (CSS) in general serves the purpose of improving the expression of XML data, and setting most media characteristics, for example font size and family, position and so on. DTDs serve to validate XML.

Figure 5.6 details the organization instance of root of e-DOCPROS system. In one e-DOCPROS system, there should be only one root instance. OID is an object identifier that uniquely specifies each element. The system populates OID sequentially. The <Type> element must be the "ROOT" for a root organization instance, which is the definition of the root in DTD. If the organization instance of the root has the value other than "ROOT" for the <Type> element, XML parser will validate it based on DTD of the root and report an error. The <Parent_OI_ID> element is the same as the <Type> element; there is no value for it. If one is given, XML parser will report an error. The <Location> elements are repeated and one is the URI (Universal Resource Identifier) representation and the other one is the local address representation.

Figure 5.7 gives an example of the organization instance of Organization A. The value of the <Type> element must be "ORGANIZATION", and the value of the <Parent_OI_ID> element must be the OI_ID of the root organization instance. For an organization instance of a Group, its <Parent_OI_ID> must have the values of its upper level organization's OI_ID.

Figure 5.8 depicts an organization instance for the User AAA. The value of <Type> has to be "USER" for a user organization instance. The <Parent_OI_ID> element is the direct group to which the user belongs. e-DOCPROS uses the <Parent_OI_ID> element to maintain the hierarchical structure of an organization.

```
<eDOCPROS-XML>
<?xml-stylesheet type="text/css" href="User.css"?>
<!DOCTYPE User SYSTEM "http://www.e-DOCPROS.com/OT_User.dtd">
    <User OID="&1">
        <OT_ID OID="&2">3001</OT_Name>
        <Type  OID="&3">USER</Type>
        <OT_Description OID="&4">The folder of User AAA</OT_Description>
        <OI_ID OID="&5">30001</OI_ID>
        <Parent_IO_ID OID="&6">20010</Parent_IO_ID>
        <Language OID="&7">English</Language>
        <Date OID="&8">12/18/1999</Date>
        <Location OID="&9">http://www.e-DOCPROS.com/Organization_A/Group_AA/User_AAA</Location>
        <Location OID="&10">\\myserver\public_html\Organization_A\Group_AA\User_AAA</Location>
    </User>
</eDOCPROS-XML>
```

Figure 5.8 An example of User organization instance

5.4.3 The XML Representation of The Document Type Hierarchy

In an e-Business system, there are many kinds of documents. By identifying the common properties for each document class, documents can be categorized into different document classes. The frame template is employed to describe the common properties of each document class. The relation between frame templates is parent-child, one-to-many relation, which depicts the hierarchy of the document classes, called document type

hierarchy (DTH) [49]. This means that a parent may have several children, and each child has at most one parent. A child document type inherits all attributes from its parent. A frame template is composed of a group of attributes. Each attribute may be of a simple or composite type.

Figure 5.9 gives an example of an XML representation of frame template for the memo type. The <Parent_FT_ID> element specifies the parent template of the memo type, which is a generic document. The generic document is defined as the root of the document type hierarchy that does not have any attributes. The root document is called a virtue document type [55]. The memo type directly inherits from the attributes of the generic document type. The logo element is an empty element that does not have a value associated with it but it contains a list of attributes: SRC, ALT, and ALIGN. The SRC attribute specifies the image's location and references an image entity. The #REQUIRED means that this attribute is required to have a value. The ALT attribute is the notation of the image. The #IMPLIED means that this attribute is optional. The ALIGN attribute specifies the alignment of the image which is to be positioned to LEFT, CENTER, or RIGHT in the document space. The default value is CENTER alignment.

<!-- ======= The DTD of Memo Frame Template ======= -->
<!ELEMENT Memo (FT_ID, FI_ID, Parent_FT_ID, Logo, Address, DepartmentName, To+, From+, Date, Subject, Content, CC*) >
<!ELEMENT FT_ID (#PCDATA) >
<!ELEMENT FI_ID (#PCDATA) >
<!ELEMENT Parent_FT_ID (#PCDATA) >

```
<!ELEMENT  Logo EMPTY >
<!ATTLIST  Logo
        SRC ENTITY #REQUIRED
        ALT CDATA  #IMPLIED
        ALIGN (LEFT|CENTER|RIGHT) "CENTER">
<!ELEMENT  Address (Street, City, ZipCode, Country) >
<!ELEMENT  Street              (#PCDATA) >
<!ELEMENT  City                (#PCDATA) >
<!ELEMENT  ZipCode             (#PCDATA) >
<!ELEMENT  Country             (#PCDATA) >
<!ELEMENT  DepartmentName      (#PCDATA) >
<!ELEMENT  To (PersonName+, Email?) >
<!ELEMENT  From (PersonName+, Email?) >
<!ELEMENT  PersonName (SingleName | (FirstName, MiddleName?, LastName)) >
<!ELEMENT  FirstName           (#PCDATA) >
<!ELEMENT  MiddleName          (#PCDATA) >
<!ELEMENT  LastName            (#PCDATA) >
<!ELEMENT  SingleName          (#PCDATA) >
<!ELEMENT  Email               (#PCDATA) >
<!ELEMENT  Date                (#PCDATA) >
<!ELEMENT  Subject             (#PCDATA) >
<!ELEMENT  Content             (#PCDATA) >
<!ELEMENT  CC                  (#PCDATA) >
```

```
<!-- ======= END of Memo Frame Template ======= -->
```

Figure 5.9 The DTD of memo frame template

Figure 5.10 gives an example of an XML representation of a frame instance of a Memo. The element of the <Parent_FI_ID> has the value of 40001, which is the FI_ID of the generic document.

```
<eDOCPROS-XML>
<?xml-stylesheet type="text/css" href="Memo.css"?>
<?DOCTYPE Memo SYSTEM http://www.e-DOCPROS.com/Memo.dtd>
  <Memo IOD="&1">
    <FT_ID OID="&2">4001</FT_ID>
    <FI_ID OID="&3">40018</FI_ID>
    <Parent_FT_ID OID="&4">40001</ Parent_FT_ID >
    <Logo OID="&5" SRC="njit.gif"/>
    <Address OID="&6">
      <Street OID="&7">123 M. L. K. Blvd.</Street>
      <City OID="&8">Newark</City>
      <ZipCode OID="&9">07102</ZipCode>
      <Country OID="&10">USA</Country>
    </Address>
    < DepartmentName OID="&11">CIS</ DepartmentName>
    < To OID="&12">
```

```
        <PersonName OID="&13">
            <SingleName OID="&14">Ph.D Students<SingleName>
        </PersonName>
        <Email OID="&15">phd@cis.njit.edu</Email>
    </To>
    < From OID="&16">
        <PersonName OID="&17">
            <FirstName OID="&18">Joe<FirstName>
            <LastName OID="&19">Smith</LastName>
        </PersonName>
        <Email OID="&20">Jsmith@cis.njit.edu</Email>
    </From>
    <Date OID="&21">10/18/1999</Date>
    <Subject OID="&22">Ph.D Qualify Exams</Subject>
    <Content OID="&23">Coming</Content>
    <CC OID="&24">faculty@cis.njit.edu</CC>
  </Memo>
</eDOCPROS-XML>
```

Figure 5.10 An example of memo frame instance

5.4.4 The XML Representation of The Folder Organization

In e-DOCPROS, folders are heterogeneous repositories of the organization instances and frame instances, which are called the folder organizations. The folder organization is

defined by a user and corresponds to the user's view of the document organization. Each folder has a user-defined criterion, which is called a predicate, for automatic document filing. A predicate-based representation of a document is used to specify the criteria for the folder organization [49]. A folder organization is an extended Directed Acyclic Graph (DAG) with predicates [85]. A DAG is a set of folders (vertices) that are interconnected by a set of arrows (edges). For a graph G, we denote the vertex set by V and the edge set by E and write G=(V, E), V={v1, v2, ..., vn} and E={e1, e2, ..., em}. The DAG nodes are called folders that can contain zero or more hyperlinks to document or subfolders. DAG's leafs are represented by hyperlinks. Each hyperlink links to a specified document. The DAG has a distinct node called the root, which is also the root of the e-DOCPROS system. Each folder is explicitly defined by the predicates that contain its domain predicates and its parent folder's predicates. A domain predicate of a folder is the local constraint criteria that are specified on the folder. The conjunction of the domain and parent's predicates of the filing path from the rooted folder to the designated folder determines whether an organization instance or a frame instance belongs to that folder. Figure 5.11 gives an example of folder organization in e-DOCPROS. A folder can be a folder that represents an organization instance or a folder that includes zero or more frame instances. A folder can include nested folder and document hyperlinks. XML is employed to represent the folder organization and easily resolves the repetitive and irregular structure of the folder organization.

Figure 5.12 depicts the folder definition employing eDOCPROS-XML. A folder can be nested and contains are all of the subfolders under it. There can be subfolders and/or organization instances or frame instances in one folder. FO_ID is the unique

identifier of the folder. The Parent_FO_ID is the parent folder's FO_ID, which is null for the root. The organization instance is directly associated with a folder and includes the OI_ID in it. The frame instances are linked to folders, which may include subfolders and frame instances. The predicate is the conditions that are used to associate an organization to a folder or file a frame instance into a folder.

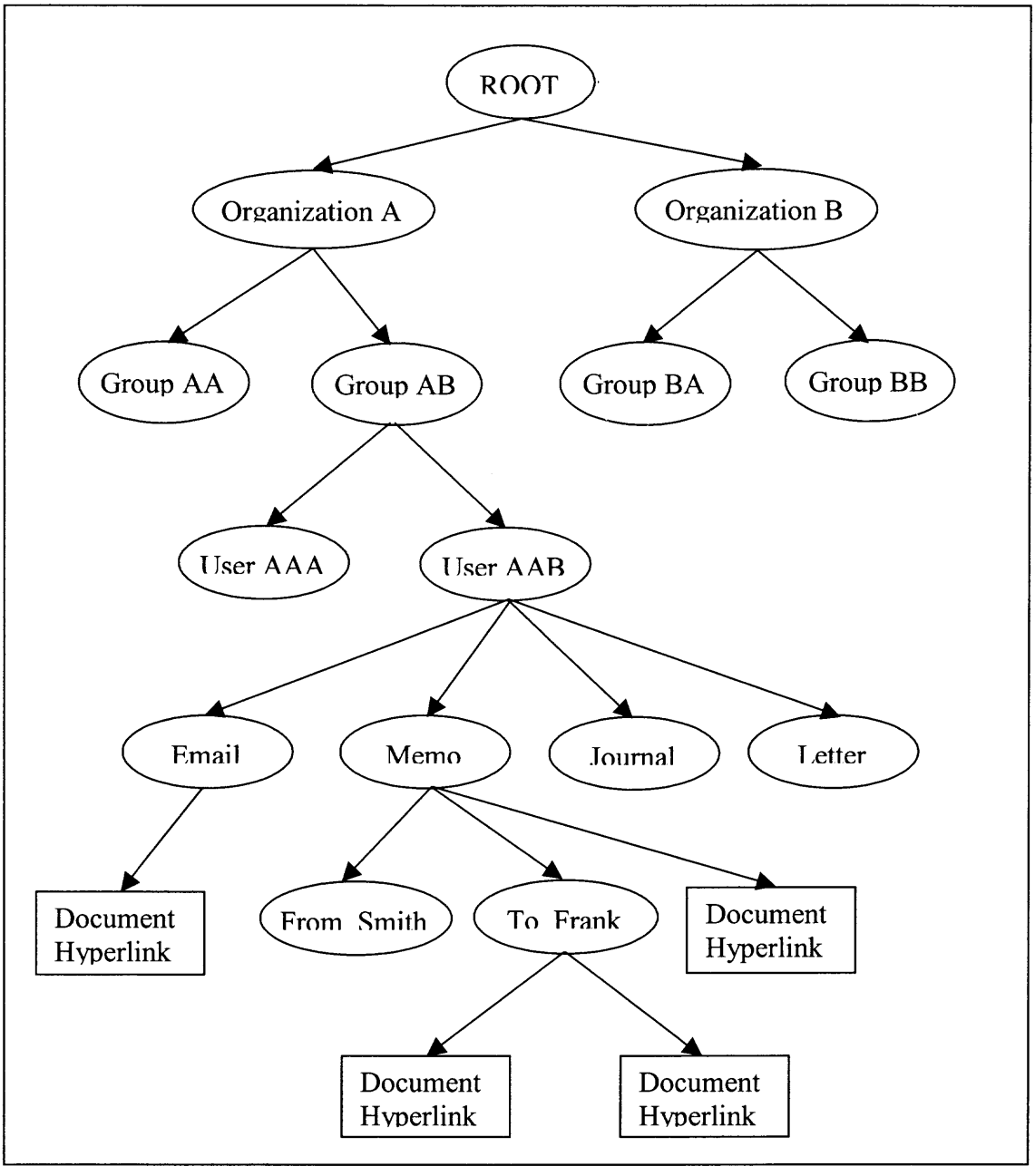


Figure 5.11 An example of folder organization

```
<!-- ======== DTD of Folder ================== -->
<!ELEMENT Folder (Folder*, FD_ID, Parent_FD_ID?, Name, Description?,
Predicate*, OI_ID*, FI_ID*)>
<!ELEMENT FD_ID (#PCDATA)
<!ELEMENT Parent_FD_ID (#PCDATA)
<!ELEMENT Name (#PCDATA)
<!ELEMENT Description (#PCDATA)
<!ELEMENT Predicate CDATA >
<!ELEMENT OI_ID (#PCDATA)>
<!ELEMENT FI_ID (#PCDATA)>
<!-- ======== End of the DTD of Folder ================== -->
```

Figure 5.12 DTD of folder

Figure 5.13 gives an example of the folder view of User A. The folder of User A is associated to the organization instance of User A. There are four folders under it, which are Email, Memo, Letter and Journal.

```
<eDOCPROS-XML>
( Folder View of the user A)
  <Folder Predicate="&Owner='User A'" Name="User A">
    <Folder Predicate="&FrameTemplate.S='Email'" Name="Email">
    <Folder Predicate="&FrameTemplate.S='Memo'" Name="Memo">
      <Folder Predicate="&Receiver.S='Frank'" Name="To_Frank">
```

```
        <FI_ID>400001</FI_ID>
        <FI_ID>400002</FI_ID>
    </Folder>
    <Folder Predicate="&FrameTemplate.S='Letter'" Name="Letter">
    <Folder Predicate="&FrameTemplate.S='Journal'" Name="Journal">
    </Folder>
  </Folder>
</eDOCPROS-XML>
```

Figure 5.13 An example of the folder tree of User AAA

The folder of Memo has the predicate that the frame template is 'Memo', which means all frame instances have the properties that the owner is 'User A' and the frame template type 'Memo' will be filed under this folder. The same for the predicate of folder To_Frank, which holds all frame instances for whom the receiver is Frank, also it should meet the parent folders' predicates. Under the folder of To_Frank, there are two frame instances, which are memo1 and memo2, whose FI_Ids are 400001 and 400002 respectively.

For each folder, e-DOCPROS assigns one agent to it and performs automatic filing. There are several learning agents to facilitate organization creation and maintenance, and document filing, extraction and storage. Knowledge regarding the organization type hierarchy and document type hierarchy are built into the knowledge base. Through the knowledge base and learning capability, e-DOCPROS provides document dynamic processing.

CHAPTER 6

e-DOCPROS SYSTEM ARCHITECTURE AND AGENT-BASED WEBTOP IMPLEMENTATION

e-DOCPROS allows users to upload documents onto the Web server. The subsystems will process the incoming documents based on a given sequence or knowledge base accordingly.

6.1 e-DOCPROS System Architecture

e-DOCPROS is a multiple user system. It includes two major parts, which are organization management and document management. Figure 6.1 depicts e-DOCPROS's system architecture. A user can create the organizational structure through GUI application. e-DOCPROS system is implemented using Java and ASP concepts [74-77, 81]. A user only needs to use any ubiquitous browser with Internet access to access the system. Users with organization maintenance privileges can create the organization structure and accounts for all users who belongs to the organization through Webtop applications. Individual users can access the workspace created for him.

Once an accounted is created for a user, he can begin to user the system. The user can upload any document onto the Web server. The system will process the incoming documents intelligently. For a general purposed document processing system, consists of a classification and extraction subsystem, a filing and storage subsystem, a retrieval and browsing subsystem and a reproducing and synthesizing subsystem. They all work collaboratively based on a given sequence. There are many learning agents necessary to get the knowledge from the user or from the system.

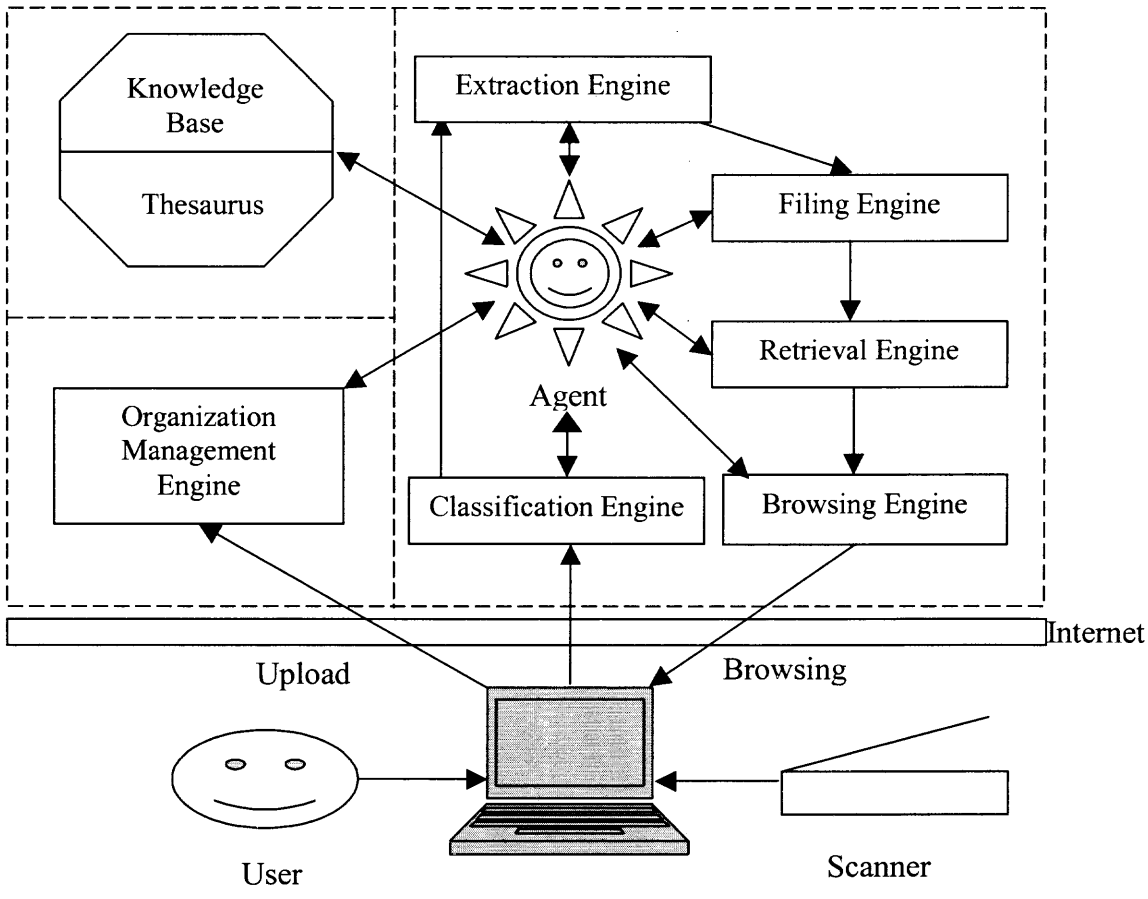


Figure 6.1 e-DOCPROS system architecture

6.2 e-DOCPROS Agent-based Webtop Implementation

e-DOCPROS is implemented using Java and ASP (Application Service Provider) concepts. Through those technologies, e-DOCPROS can support the four business models, B2B, B2C, C2C and the Intranet model cleanly and can be deployed easily over the Internet, Intranet or an Extranet.

6.2.1 Application Service Provider Model

Application Service Provider (ASP) model is employed in e-DOCPROS system to achieve the cost-effectiveness and accessibility.

Only a year ago, few people had heard the term "ASP". Today, it is one of the hottest buzzwords in the business and computer communities, almost as big as the "World Wide Web" was a few years ago [36]. The ASP, short for Application Service Provider, is poised to change the way we buy, maintain, and upgrade our software. It could even change computers themselves.

ASPs are a return to centralized computing, but with a twist. Applications are stored on a remote server and distributed to personal computers via the Internet and Web browser. More important, though, ASP software products are rented – one does not actually own a copy of these applications, but instead pays a monthly fee to access them via the Internet. If one uses ASP for a word processor, for example, one does not have to buy a shrink-wrapped software package. Forget about installation, and you do not need to monitor the company's Web sites for patches, bug fixes, and upgrades. It all happens automatically.

Using an ASP-provided application is not much different than using a program stored on your own PC. To use a Web-based scheduling program, for instance, you connect to the Internet and log onto the site that stores the application. The program's interface might appear in a Web browser, or open in its own window. As you enter data, it is transmitted to a remote server where the program engine is stored. The program sends data back to your PC, and the display is changed.

ASPs also have the property of a built-in security feature. If your computer is lost or stolen, or crashed, your company's data is safe on your provider's server, not on the lost machine's hard drive.

6.2.1.1 Webtop Applications. In the APS model, the only thing a user needs is a browser with Internet access. All applications run on the provider's server. Those applications displayed in the browsers are called Webtop applications. Webtop ASP applications have exploded. At the heart of the Webtop phenomenon are email services. Using any of these, you can outsource your email so that it is not bound to any specific desktop system or company servers.

Using Wireless Access Point (WAP)-enabled cellular phone or PDA, the users can access their data from anywhere and are not restricted to the office. This will eventually change the way we manage information.

Collaborative desktop tools are flourishing on the Internet now, and are one of the best ways to get a taste of what ASPs have to offer. The system can have several applications, they can be used collaboratively with exchanges between those applications, thus improve the efficiency. Through ASPs, team members can use the system from any location, as long as they are running a Web browser with the Internet access. They can share the data instantly and access the data collaboratively. For example, in a team design of a large project, different members access the different part of the project, but the team as whole has an interdependency. They need to access the up-to-date data timely. From user's perspective, Office Online works exactly the same as the regular version of Office, except that data and applications are stored remotely.

6.2.1.2 The Next Standard of Software. ASP came into our vocabulary in 1999. In 2000, ASP businesses are becoming more mainstreaming. ASP will become a standard soon. People may wonder why they ever went to the trouble of buying software [36].

An era of ASP-based software solution could have implications for every aspect of the computer market. The ASPs may profoundly affect the computer industry in one other way. Today, less than half of homes in United States have access to broadband Internet like DSL or cable modems. That is a problem for ASPs. It is believed that good ASP products will drive broadband connectivity, and conversely, more broadband access will allow other ASP solution to flourish.

6.2.2 Agent-based Implementation

e-DOCPROS employs agents [71, 72] to perform organization maintenance, and document filing and retrievals. Also, there are many learning agents necessary to capture the users' behavior and knowledge of the organizational structure and the documents and store them into knowledge base.

6.3 Paper Document and Electronic Document Processing

e-DOCPROS supports both paper documents and electronic documents. Paper documents still play an important role in the current office environment. A user must scan the paper document into an image file, such as TIF format file. Then the user uploads it onto the Web server. The system uses OCR (Optical Character Recognition) technology to transfer the image file into a computer readable ASCII file with layout information. Classification subsystem identifies the document type based on the sample base. Once it

determines the document type, it will extract the information of the document into a frame instance of the document. The filing and storage subsystem reads the information of the frame instance of the document, and then automatically file the documents into the appropriate folders based on the predicates of those folders. The system can record the user's behavior of creating folders and keep this information in its knowledge base. At the same time, it generates predicates for the newly created folders. Each folder is assigned one dedicated agent to perform the filing. After that, the user can use the retrieving and browsing subsystem to look up the documents.

XML is employed to transform the message among the subsystems and ach of the agents. The domain of the status of a document is defined in DTD of the document and the processing actions. The sequence of the actions is also defined in the DTD of the document. The agents read the specific documents with the dedicated status and process them accordingly. After all actions are applied, the document is in a complicated and stable status. The complete status includes two types of values, which are completed successfully, and processed with errors. When an error occurs during the processing, the agent will report the error back to the users. When this occurs, the user can take action manually. e-DOCPROS provides the capability to learn the way a user fixes problems, so that, at a later time, it can refer it to the user-solved problems and perform the filing process automatically. Component technology is new concept of software development. Components are those that are deployable units of executions, manipulated by tools, developed by third parties, and often instantiated as objects [49]. ASP model dramatically changes the way we design, implement, and deploy software. e-DOCPROS implements all of the subsystems as agent-based Webtop ASP applications.

CHAPTER 7

CONCLUDING REMARKS AND FUTURE WORKS

7.1 Concluding Remarks

This dissertation presents e-DOCPROS, an automatic knowledge-based e-Business document management system, which supports B2B, B2C, C2C and Intranet business models. A triple mode is proposed to model e-Business office automation. Organization type hierarchy (OTH) and document type hierarchy type hierarchy (DTH) are integrated together through a folder organization (FO). The triple model makes e-DOCPROS a multiple user system operating at the state-of-art. The data access and information sharing mechanism makes e-DOCPROS suitable for collaborative design and processing. All subsystems of e-DOCPROS are Webtop applications, designed using ASP model (Application Service Provider) and implemented by Java programming language.

e-DOCPROS is a more flexibility and knowledge-base system than existing systems. It is designed to be a dynamic office automation system which allows us to manage the documents effectively and efficiently.

7.2 Future Works

e-DOCPROS needs the following future works:

7.2.1 Document Dynamic Processing

e-DOCPROS can use XML to define the actions to be applied to the documents. The system has the learning capability to obtain the user's knowledge of how the document

should be processed during the learning stage. Once the system gets trained, it can process the incoming documents automatically. However, there are some variations between the incoming documents and the existing document types. Hence, the system must consult the knowledge base and dynamically generate new rules to apply the variations, and update the knowledge base. Here we call this capability of dynamic processing.

Another case for dynamic processing is that the system can process the incoming documents with the new document type. The system still consults the knowledge base, try to generate a new rule for this type of documents and try to eliminate the user interaction. This will give the system more usefulness.

7.2.2 Personalization

Personalization is to build a meaningful one-to-one relationship by understanding the needs of each individual and helping satisfy a goal that efficiently and knowledgeable addresses each individual's need in a given context. To extend this point, it is about the mapping and satisfying of a user's goal in a specific context with a service's goal in its respective context [94]. e-DOCPROS employees learning agents to acquire a user's behaviors and knowledge of the documents, then stores them into knowledge base. The system generates user specific profiles for each individuals or organizations. How we will measure the success of personalization is another difficulty.

7.2.3 Internationalization

Internationalization is to adapt the system to global world. Through internationalization, the system can obtain wide usage and gain more success. There are several issues involved, such as languages, techniques, culture, and so on. e-DOCPROS uses OCR to convert paper based documents into computer-readable content. Once adapting it into internal environment, OCR should be able to support multiple languages. Some special design and implementation are considered to accommodate more requirements. For example, zip code is defined as integer type in TEXPROS and defined as string type in e-DOCPROS. It is necessary to acquire most requirements, design and implement to achieve this goal.

REFERENCES

1. Daniel Amor, *The E-Business Revolution: Living and working in an Interconnected World*, Hewlett-Packard Professional Books, 1999.

2. N. Bianchi, P. Mussio, M. Padula, and G. R. Rinaldi, "Multimedia Document Managemente: An Anthropocentric Approach," *Information Processing & Management*, Vol. 32, no 3, pp. 287-303, 1996.

3. A. Celentano, M. Fugini, and S. Pozzi, "Querying Office Systems about Document Roles," in *Proc. Of the 14th Annual Int. ACM/SIGIR Conf. On Research and Development in Information Retrieval*, Chicago, Illinois, pp. 183-189. October 1991.

4. A. Celentano, M. Fugini, and S. Pozzi, "Knowledge-Based Document Retrieval in Office Environment: The Kabiria System," *ACM transactions on Office Information System*, Vol. 13, no. 3, pp. 237-268, July 1995.

5. S. Christodoulakis, M. Theodoridou, M. P. F. Ho, and A. Pathria, "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and System," *ACM Trans. On Office Information System*, Vol. 4, no. 4, pp. 345-383, 1986.

6. P. Dadam and V. Linnemann, "Advanced Information Management (AIM): Advanced Database Technoloyy for Integrated Applications," *IBM Systems Journal*, Vol. 28, no. 4, pp. 661-681, 1989.

7. S. Gibbs and D. Tsichritzis, "A Data Modeling Approach for Office Information System," *ACM Transactions on Office Information System*, Vol. 1, no.4, pp. 299-319, October 1983.

8. C. Meghini, R. Fausto, and C. Thamos, "Conceptual Modeling of Multimedia Document," *Computer*, Vol. 24, no. 10, pp. 23-29, 1991.

9. S. Pierre and H. Safa, "Models for Storing and Presenting Multimedia Document," *Telematics and Informatics*, Vol. 13, no. 4, pp.233-250, 1996.

10. S. Pozzi and A. Celentano, "Knowledge-Baesd Document Filing," *IEEE Expert*, pp. 34-35, October 1993.

11. G. Salton, *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley, Massachusetts, 1998.

12. G. Salton and M.J.McGill, *Introduction to Modern Information Retrieval*, McGraw Hill, New York, 1983.

13. S. TSUJIMOTO and H. ASADA "Understanding Multi-articled Documents," *10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, 1990.

14. J. Higashino, H. Fujisawa, Y. Nakano and M. Ejiri, "A knowledge-based segmentation method for document understanding," *Proceedings of 8th International Conference on Pattern Recognition*, pp. 745-768, 1986.

15. John F. Cullen, Jonathan J. Hull and Peter E. Hart, "Document Image Database Retrieval and Browsing using Texture Analysis". *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pp. 718-721, August 1997.

16. Xiaolong Hao "Automatic Office Document Classification and Information Extraction", *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, October 1995.

17. X. Hao, J.T.L. Wang, M. Bieber, and P. A. Ng, "A Tool for Classifying Office Document," *Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence*, pp. 427-439, November 1993.

18. X. Hao, J.T.L. Wang and P. A. Ng, "Nested Segmentation: An approach for Layout Analysis in Document Classification," *Proceedings of the Second IAPR Conferences on Document Analysis and Recognition*, pp. 319-322, Tsukuba Science City, Japan, October 1993.

19. ChingSong Wei "Knowledge Discovering for Document Classification Using Tree Matching in TexPros", *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1996.

20. C.Y. Wang, Q. Liu, and P.A. Ng, "Browsing in an Information Repository," *Proceeding of 2nd World Conference on Integrated Design and Process Technology*, IDPT-Vol2, pp. 48-56, 1996.

21. C.Y. Wang, Q. Liu, and P.A. Ng, "Intelligent Browser for TEXPROS," *ISATED Proceeding of International Conference on Intelligent Information Systems (IIS'97)*, IEEE Computer Society Press, pp. 388-398, December 1997.

22. Xien Fan and Peter A. Ng, "Personal Document Management and Retrieval: A Knowledge-Based Approach," *Journal of Systems Integration*, vol. 8, No. 3, 1998.

23. Xien Fan, Peter A. Ng, "A Dual Model Approach for Modeling Office Documents," *Proceedings of International Workshop on Issues and Applications of Database Technology*, 1998.

24. J. Wang and P.A. Ng, "TEXPROS: An Intelligent Document Processing System," *International Journal of Software Engineering and knowledge Engineering*, Vol.15, No. 4, pp. 171-196, April 1992.

25. J. Wang, F. Mhlanga, Q. Liu, W. Shang and P. Ng, "An Intelligent Documentation Support Environment," *Proceedings of the Fifth International Conference on Software Engineering and Knowledge Engineering*, San Francisco, California, pp. 429-436, June 1993.

26. X. Li, J. Hu, X. Fan, C. Y. Wang and P. A. Ng, "Automated Document Filing and Retrieval," in *Proceedings of the Third World Conference on Integrated Design & Process Technology*, 1998.

27. Xuhong Li, Jianshun Hu, Zhenfu Cheng, D.C. Hung and Peter A. Ng "Automatic Document Analysis and Understanding System," *The First International Conference on Enterprise Information System*, 1999.

28. Xuhong Li, Jianshun Hu, Zhenfu Cheng, Simon Doong, D.C.Hung and Peter A. Ng "An Integrated Document Processing System: Document Classification and Information Extraction," *The Fourth World Conference on Integrated Design and Processing Technology, incorporating IEEE International Conference on System Integration*, 1999.

29. Jianshun Hu, Xuhong Li, Simon Doong, D.C. Hung and Peter A. Ng, "A Thesaurus Model for Document Processing System: The TEXPROS Approach," *The Fourth World Conference on Integrated Design and Processing Technology, Incorporating IEEE International Conference on System Integration*, 1999.

30. Jianshun Hu, Xuhong Li, D.C. Hung, Simon Doong and Peter A. Ng, "Managing Knowledge for an Intelligent Document Processing System", *The First International Conference on Enterprise Information System*, 1999.

31. Q. Liu, An Office Document System With the Capability of Processing Incomplete and Vague Queries, *Ph.D. dissertation*, Dept. of Computer and Information Science, New Jersey Institute of Technology, Newark, New Jersey, August 1994.

32. Q. Liu and P. Ng, "A Browser of Supporting Vague Query Processing in an Office Document System," *Journal of System Integration*, Vol. 5, No. 1, pp. 61-82, 1995.

33. Q. Liu and P. Ng *Document Processing and Retrieval: Text Processing*, Kluwer Academic Publishers, Norwell, Massachusetts, 1996.

34. E. Ukonnen "On approximate string match," *Proceedings of the International Conference on Foundation of Computation Theory*, Borgholm, Sweden, August 1983.

35. Yuan Y. Tang, Chang De Yan and Ching Y. Suen, "Document Processing for Automatic Knowledge Acquisition," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No.1 pp. 3-21, 1994.

36. Dave Johnson, "Application Service Provider," *Laptop Buyer's Guide & Handbook*, Bedford Communication, Inc., pp. 72-83, May 2000.

37. G. Nagy, S.C. Seth, and S.D. Stoddard, "Document analysis with an expert system," in E.S.Gelsema and L.N.Kanal, Eds., *Pattern Recognition Practice II*, pp. 149-159, New York: Elsevier, 1986.

38. H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis," *Proceedings of the IEEE, 80(7),* pp. 1079-1091, July 1992.

39. E. Lutz, H.V. Kleist-Retzow, and K. Hoernig, "MAFIA-An Active Mail-Filter-Agent for an Intelligent Document Processing Support," *Multi-User Interface and Applications,* eds., S.Gibbs and A.A.Verrijn-Stuart, Elsevier, Science Publishers, North Holland, pp. 16-32, 1990.

40. A. Dengel and G. Barth, "High Level Document Analysis Guided by Geometric Aspects," *International Journal of Pattern Recognition and Artificial Intelligence.* Vol 2. No 4 pp. 641-655, 1988.

41. A. Dengel, "Document image analysis-expectation-driven text recognition," in *Proceeding of Syntactic and Structural Pattern Recognition.* (SSPR90), pp. 78-87, 1990.

42. Andreas Dengel and Gerhard Barth. ANASTASIL: "A hybrid knowledge-based system for document layout analysis," *Proceedings of the 11th International Joined Conference for Artificial Intelligence*, pp. 1249-1254, Michigan, 1989.

43. E. Riloff. "Using Cases to Representing Context for Text Classification," *Proceeding of AAAI Spring Symposium on Cased-Based Reasoning and Information Retrieval*, 1993.

44. E. Riloff and W. Lehnert. "Information Extraction as a Basis for High-Precision Text Classification," *ACM Transactions on Information Systems*, 12:269-333, 1994.

45. W3C, "Document Object Model (DOM) Requirements: W3C Working Draft," April 12, 2000.

46. Forrester Research Press Release, September 13, 1999, http://www.forrester.com.

47. Forrester Research Press Release, October 1, 1999, http://www.forrester.com.

48. Gartner Group Press Release, January 26, 2000, http://www.gartner.com.

49. Xien Fan, Knowledge-based Document Filing for TEXPROS, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1998.

50. Leon Jololian, A Meta-Semantic Language For Smart Component-Adapters, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1999.

51. Jr-Tian Lin, Collaborative Software Agents Support for the TEXTPROS Document Management System, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, December 1999.

52. David Hunter, *Beginning XML*, Wrox Press Ltd, Birmingham, UK, 2000.

53. Connolly D. and Bosak J., *Extensible Markup Language (XML) 1.0 specification*, W3C Recommendation, Feburary 1998.

54. Barker R., *CASE Method: Entity Relational Modeling*, Addison-Wesley Publishing Co., 1990.

55. Booch G., *Objected-Oriented Analysis and Design with Applications*, 2nd edition, Addison-Wesley, 1994.

56. ISO 8879, *ISO 8879:1986, Information Processing – Text and Office System – Standard Generalized Markup Language (SMGL)*," 1986.

57. Kalakota R., and Whindton A.B., *Electronic Commerce: a Manager's Guide*, Addison-Welsey Longman, ACM Press, New York, 1996.

58. Berners-Lee T., "Information Management: A Proposal," *CERN*, March 1989.

59. Berners-Lee T., "Keynote Address," *Seybold*, San Francisco, February 1996.

60. Alex Homer, Chris Ullman and Steve Wright, *Instant HTML*, Wrox Press Inc., 1998.

61. Smiss J.B. and Weiss S.F., "An Overview of Hyper Text," *Communication of the ACM*, Vol. 31, No. 7, 1988.

62. Sokol P.K., *From EDI to Electronic Commerce: A Business Initiative*, McGraw-Hill, 1995.

63. Khare R. and Rifkin A., "X Marks the Spot: Using XML to Automate the Web," *IEEE Internet Computing*, Vol 1, Number 4, pp. 78-87, July/August 1997.

64. Abiteboul S., Cluet S., Christophides V., Moerkotte G., and Simeon J., "Querying documents in object database," *International Journal on Digital Libraries*, 1(1), pp. 5-19, 1997.

65. Albiteboul S., Quass D., McHugh J., Widom J., and Wiener J., "The Lorel Query Language for Semistructured Data," *International Journal on Digital Libraries*, 1(1):pp. 68-88, 1997.

66. Albiteboul S., "Querying semi-structured data," *ICDT*, 1997.

67. Buneman P., "Tutorial: Semistructured Data," *PODS*, 1997.

68. Fernandez M., Florescu D., Levy A., and Suciu D., "A Query Language and Processor for a Web-site Management System," *Proceedings of the Workshop on Management of Semi-structured Data*, 1997.

69. Fernandez M., Popa L., and Suciu D., "A Structured-based Approach to Querying Semi-structured Data," *Proceedings of the Workshop on Database Programming Language*, 1997.

70. Lieberman H., "Autonomous Interface Agents," *CHI,* 1997.

71. Malone T. W., Grant K. R., Lai K.Y., "Agent for Information Sharing and Coordination," In *Software Agent, ed J.M. Bradshaw, Melo Park, AAAI Press*, Calif., 1997.

72. Jiming Liu, Ning Zhong, *Intelligent Agent Technology: System, Methodologies and Tools*, Proceedings of the 1st Asia-Pacific Conference on IAT, 1999.

73. D. Comer and D. Stevens, *Internetworking with TCP/IP Volumn III Client/Server Programming and Applications*, Prentice-Hall, 1997.

74. Farley J., *Java Distributed Computing*, O'Reilly & Associates, 1997.

75. Cay S. Horstmann, Gary Cornell, *Core Java 2: Volume I – Fundamentals*, Prentice-Hall, NJ, 1999.

76. Cay S. Horstmann, Gary Cornell, *Core Java 2: Volume II – Advanced Features*, Prentice-Hall, NJ, 2000.

77. H.M. Deitel and P.J. Deitel, *Java: How to Program, Third Edition*, Prentice Hall, 1999.

78. Alex Homer, Chris Ullman and Steve Wright, *Instant HTML: Programmer's Reference*, Wrox Press Ltd., Birmington, UK, 1998.

79. Jalal Feghhi, Jalil Feghhi and Peter Williams, *Digital Certificates: Applied Internet Security*, Addison-Wesley, 1999.

80. Jill Dyche, *e-Data: Turning Data into Information with Data Warehousing*, Addison-Wesley, 2000.

81. Danny Ayers, Hans Bergsten, et, *Profession Java Server Programming*, Wrox Press Ltd., 1999.

82. Marc Abrams, *World Wide Web: Beyond the Basic*, Prentice Hall, Upper Saddle River, NJ, 1998.

83. Elrod, Scott, Gene Hall, et al, "Responsive Office Environments," *Communication of ACM*, pp. 85-85, July, 1993.

84. Spreitzer, Mike and Marvin Theimer, "Scalable, Secure, Mobile Computing with Location Information," *Communication of ACM*, pp. 27, July 1993.

85. James A. McHugh, *Algorithmic Graph Theory*, Prentice Hall, NJ, 1990.

86. Simon Doong, A Folder Organization Model for Office Information System: Exploring Its Architecture Expressive Power and Predicate-Based Filing, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1998.

87. Chih-Ying Wang, The Intelligent Browser For TEXPROS, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1998.

88. Xuhong Li, Automatic Document Classification and Extraction System (ADoCES), *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1999.

89. Jianshun Hu, Knowledge Management for TEXPROS, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, May 1999.

90. Hong Shen, HYTEXPROS: A Hypermedia Information Retrieval System, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, December 1999.

91. Yin Dong, A More Efficient Document Retrieval Method For TEXPROS, *Ph.D. Dissertation*, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, New Jersey, August, 2000.

92. Q. Liu, An Office Document System With The Capability of Processing Incomplete and Vague Queries, *Ph.D. Dissertation*, Department of Computer and Information Science, New Jersey Institute of Technology, Newark, New Jersey, August, 1994.

93. Q. Liu and P. Ng, "A Browser of Supporting Vague Query Processing in an Office Document System," *Journal of System Integration*, Vol. 5, No. 1, pp. 61-82, 1995.

94. Doug Riecken, "Personalized Views of Personalization," *ACM Communications*, Vol 43, Number 8, August 2000.

95. Krithi Ramamritham and Panos K. Chrysanthis, *Advances in Concurrency Control and Transaction Processing: An Executive Briefing*, Computers, September 1996.