國立交通大學

資訊工程學系

博士論文

基於本體論數位學習內容擷取之研究

A Study of Ontology-based Learning Content Retrieval on e-Learning Environments

研 究 生：時文中

指導教授：曾憲雄 博士

中華民國九十七年五月

基於本體論數位學習內容擷取之研究

A Study of Ontology-based Learning Content Retrieval on e-Learning Environments

研 究 生：時文中　　Student：Wen-Chung Shih

指導教授：曾憲雄 博士　　Advisor：Dr. Shian-Shyong Tseng

國 立 交 通 大 學

資 訊 工 程 學 系

博 士 論 文

A Dissertation
Submitted to Department of Computer Science
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science
May 2008
Hsinchu, Taiwan, Republic of China

中華民國九十七年五月

國立交通大學

資訊工程系博士班

論文口試委員會審定書

本校 資 訊 工 程 系 時 文 中 君

所提論文 基 於 本 體 論 數 位 學 習 內 容 擷 取 之 研 究

合於博士資格水準、業經本委員會評審認可。

口試委員： 曾憲雄

楊朝棟 葉耀明

指導教授： 曾憲雄

系 主 任： 曾建超

中華民國 九十七 年 五 月 廿六 日

基於本體論數位學習內容擷取之研究

研究生：時文中　　　　　　　　　　　　　　　　　　指導教授：曾憲雄 博士

國立交通大學 資訊學院

資訊工程學系

摘要

內容擷取對於數位學習應用發展是非常重要的一環。然而，一般的資訊檢索技術著重於關鍵詞層面的處理，對於詮釋資料、文件結構與概念層面著墨較少。事實上，數位內容的擷取若能支援教育理論的應用，例如：教學策略、學習風格等，對於教學成效的提升將更有助益。有鑑於此，本論文採取較高層次的觀點來處理數位學習內容。希望能針對新興的數位學習環境(格網環境、環境感知學習、同儕網路)，提出有效的內容擷取方法。近年來，本體論已成為探討知識領域概念化的重要模型。因此，本文提出基於本體論的模型與方法，從上層概念的觀點來組織與檢索內容，然後將此模型實現在三種新興的數位學習環境。首先，我們針對集中式格網環境提出「由下而上」的方式，為大量且分散各地的內容建立索引。其次，我們為環境感知學習設計了一個支援教學策略的查詢擴充法。接下來，我們探討同儕網路上的等待回覆時間設定問題。此外，我們也說明如何建置本文所需的本體論，並利用「庶民分類法」來調整本體論與註記內容。綜上所述，我

們將這些想法應用在教材設計，提出了一個「維基式」方法。我們實作了這些方法，並針對國小師生進行相關實驗。評估結果顯示本文提出的方法可以達到有效且快速的內容擷取。問卷調查結果也顯示使用者認為這套工具有助於取得適合的學習內容。

關鍵詞：數位學習、內容擷取、本體論、維基、庶民分類、環境感知、格網計算、同儕網路

A Study of Ontology-based Learning Content Retrieval on e-Learning Environments

Student: Wen-Chung Shih Advisor: Dr. Shian-Shyong Tseng

Department of Computer Science

College of Computer Science

National Chiao Tung University

Abstract

Content retrieval is important for the development of e-learning applications. However, existing information retrieval techniques focus on keyword-level processing, rarely considering metadata, structural information and concepts of contents. In fact, content retrieval can increase learning performance by supporting pedagogical theories and models, such as instructional strategies, learning styles, etc. Therefore, this dissertation addresses learning content from a high-level viewpoint, and proposes effective and an efficient content retrieval approach for emerging e-learning environments, such as grid environments, context-aware learning and peer-to-peer networks. In recent years, ontology has been widely investigated to model and conceptualize domain knowledge. Hence, an ontology-based model is proposed to represent, organize and retrieve content from a high-level viewpoint.

Then, the proposed approach is realized in three emerging e-learning environments. First, a bottom-up method is proposed for centralized grids to create indices of contents in widely spread repositories. Next, a knowledge-based query expansion method is designed to support instructional strategies for context-aware ubiquitous learning. Then, the due-time setting problem for retrieval in P2P networks is addressed and solved by an interactive method. In addition, the ontology construction is described. A folksonomy-based method is proposed to refine the ontology and annotate the learning content. Next, the above-mentioned ideas are applied to teaching-material design by a Wiki-based method. The aforementioned methods are implemented, and experiments are conducted in an elementary school. The results of evaluation show that the proposed method is efficient and effective. Surveys of user opinions also show that the proposed approach can assist learners to retrieve appropriate learning content.

Keywords: e-Learning, Content retrieval, Ontology, Wiki, Folksonomy, Context-awareness, Grid computing, P2P networks

誌　謝

「或生而知之；或學而知之；或困而知之：及其知之，一也。」　　- 中庸 -

本論文能完成，首先要感謝的是我的指導教授 曾憲雄博士。我在研究方面若有一絲一毫的突破與創新，都要歸功於 曾教授對教育理念的堅持及在學術研究的創見。 曾教授不論在教學研究或為人處事，都為學生樹立了典範。在跟隨 曾教授學習的過程中，我得以領略學術研究的奧妙，也體會了從抽象層次來發現、解決問題的重要性。如果沒有 曾教授循循善誘地費心指導與鼓勵，就沒有本論文的誕生。

接著要感謝 楊朝棟教授伉儷從我碩士班時期起，十多年來持續的鼓勵與幫助。 楊教授引領我進入 Grid Computing 的領域，毫無保留地指導所有觀念、實作細節與論文寫作技巧。這些為後續 e-Learning 方面的研究打下基礎，而 楊教授給我的啟蒙與鼓勵也讓我有信心堅持到最後。

我特別要感謝從論文研究計畫書口試、校內口試到最後口試一路給予我許多論文修改建議的 袁賢銘教授與 彭文志教授，讓我重新檢視本論文的研究貢獻。感謝在校內口試中， 梁婷教授對研究方法的提點；也感謝在最後口試中，給予我寶貴意見的 楊鎮華教授、 葉耀明教授以及 楊朝棟教授，對於論文方法與結果呈現，給於精闢的見解與指導。由於他們的指正，讓此論文最後的成果能夠更加完整並增加整體論文的可讀性。

本論文的核心內容是 2006 年起，每週在亞洲大學與 曾憲雄教授討論產生的。非常感謝亞洲大學資科系 廖岳祥主任一直給我誠摯的鼓勵與研究上的建議；感謝資訊學院的瓊如(Apple)與資工系的君怡這段期間的協助。

感謝交大知識工程實驗室的夥伴們，特別是瑞鋒、蓮芳、婷郁給我的支援與幫助。同時也感謝系辦的耀萱協助畢業相關事宜。還要感謝暨大圖書館 張帆人館長、 吳幼麟館長、 黃光璿組長以及暨大所有長官同仁給我的支援、包容與鼓勵。另外，交大電資中心、台中榮總與藥檢局的同仁，謝謝您們過去給我的幫助。

要感謝的人太多，謹將本篇論文的完成，獻給我的每一位家人：特別是我的姊姊、姊夫；我的岳母以及小偉、慧中、小茹；賢內助靜怡與可愛的兒女御唐、華君；永遠支持我的母親。

最後要將本論文獻給在天上的 父親。

時文中

2008 年 5 月於埔里

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1 Introduction

During the past decade, information technologies have advanced at an amazing pace. Web 2.0, characterized by the techniques of blog, wiki, folksonomy, RSS, mashup, etc., has been widely discussed and referred to as the second generation of web-based services [1]. Another representative is emerging computing environments, such as grids, P2P networks, wireless sensor networks. For example, grid computing [2, 3], which supports resource-sharing and can overcome the limitations of computing power and storage capacity in traditional platforms. Therefore, grid computing technologies provide possibilities for supporting innovative applications, such as e-Learning. In fact, more and more effort has gone into the field of e-Learning grid, using grid technologies in the context of e-Learning. Among these, ELeGI (European Learning Grid Infrastructure, 2004-2008) is the most representative project [4].

The Retrieval of learning contents means searching for desired learning content in Learning Object Repositories (LOR). The importance of learning content retrieval lies in two aspects. First, content retrieval is pervasive in learning and teaching activities. While teachers need to retrieve related content for instruction, students want to find more relevant content for learning. Second, the "Taxonomy for educational objectives", proposed by Bloom, indicates that the abilities of comprehension, interpreting, clarifying, representing, etc. are important pedagogical objectives, which can be learned from the content retrieval process.

With the promising development of various e-Learning environments, there will be a great demand to find desired teaching materials from repositories in the e-Learning environment. The difficulties of learning content retrieval result from the tremendous amount of content and the various features of emerging e-Learning environments. A dominant approach, used to cope with the web information overload problem, is using keyword-based information retrieval technologies to search for web pages. However, this approach has limitations when they are naively applied to learning content retrieval. First, it returns too many results which are not relevant, just like typical search engines. Second, it does not utilize domain know-how, context information, content structures, etc., to improve the performance of content retrieval. Third, it is weak in supporting high-level educational theories and models, such as intelligent tutoring systems, instructional strategies, learning styles, etc.

In this thesis, learning content retrieval is formulated as a general problem, the Learning Content Retrieval Problem (LCRP), which has several parameters to model various e-Learning environments. Specifically, this problem is specialized to three emerging environments: sensor networks, centralized grids and P2P networks. In addition, a four-layer content model is proposed to summarize various methods of learning content organization. In short, content can be organized in viewpoints of keywords or image features, metadata, structures and concepts. Based on this model, an ontology-based approach to solving the LCRP problem is proposed, aiming at fast and precise content retrieval. The main idea is to increase precision by ontology-based semantic search, and to reduce search time by ontology-based indexing. The idea of ontology building is based on folksonomy, in order to alleviate the heavy burden of experts and knowledge engineers. This framework consists of two phases. In the Construction phase, ontology and indices are constructed to facilitate semantic search. Next, users' queries are interactively verified in the Search phase, and desired content is retrieved fast and precisely.

The contributions can be summarized as follows. First of all, the learning content retrieval problem is formulated as a general problem model, which can not only represent existing e-Learning environments, but also be compatible with future e-learning environments. Second, an ontology-based approach to this problem is proposed, and is realized in three emerging environments. Also, a method for ontology self-organizing is proposed for this approach. Then, all these proposed techniques are applied to teaching material design, in a wiki-based rapid prototyping manner. Finally, technical experiments and satisfaction surveys are conducted to evaluate this approach. To sum up, this solution brings educational benefits for related stakeholders. For students, they can find relevant content to improve their learning. Teachers can reuse teaching materials to enhance their instruction. Furthermore, educational experts can organize learning content with ease.

The rest of this thesis is organized as follows. In Chapter 2, the preliminaries and related researches are reviewed. Then, the formulation of the LCRP problem model is presented in Chapter 3. Chapter 4 explains the derivation of the underlying ontology. Next, the three realizations on e-learning environments are introduced in Chapters 5, 6 and 7, respectively. Then, an application of teaching material design is shown in Chapter 8. Finally, the concluding remarks are given in Chapter 9.

Chapter 2 Preliminaries and Related Work

In this chapter, the preliminaries related to this thesis are reviewed, including information retrieval, middleware of Grid computing, ubiquitous learning, Sharable Content Object Reference Model (SCORM), Wiki technology and ontology construction approaches.

2.1 Information Retrieval

General information retrieval methods are mainly designed for web pages. Nevertheless, documents in specific domains may need tailor-made methods to improve their retrieval performance. For example, FAQ (Frequently Asked Questions) search [5-7] and patent retrieval [8-10] are widely investigated to find more efficient methods.

Teaching materials are also a special kind of documents. They are distinct in two aspects. First, they are for educational purposes. Second, they have standardized format, such as Learning Objects, Content Packages. To share and reuse teaching materials, several standards have been proposed recently. Among these, SCORM (Sharable Content Object Reference Model, http://www.adlnet.org/) is the most popular standard for learning contents. It was proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997. This standard consists of several specifications developed by IEEE LTSC (Learning Technology

Standards Committee, http://ltsc.ieee.org/wg12/), IMS (Instructional Management System, http://www.imsproject.org/), AICC (Aviation Industry CBT Committee, http://www.aicc.org/), etc. SCORM Metadata refers to the IEEE's LOM (Learning Object Metadata, http://ltsc.ieee.org/wg12), and describes the attributes of teaching materials. IEEE LOM v1.0 includes nine categories: General, LifeCycle, Meta-Metadata, technical, educational, rights, relation, annotation, and classification.

Inverted file indexing has been widely used in information retrieval [11-14]. An inverted file is used for indexing a document collection to speed up the searching process. The structure of an inverted file consists of two components: the vocabulary and the posting list, as shown in Figure 2.1. The vocabulary is composed of all distinct terms in the document collection. For each term, a list of all the documents containing this term is stored. The set of all these lists is called the posting list. However, the structure of a document is not considered in this model.

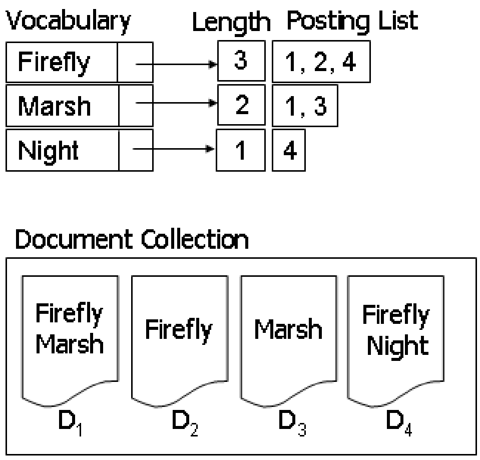


Figure 2.1 A sample document collection and a corresponding inverted index

Storage requirements of inverted indices [15] have been evaluated based on B+-tree and posting list. Five strategies of the index term replication were discussed. This approach is extended to analyze the storage requirement of the proposed approach in this thesis. In [16], 11 different implementations of ranking-based text retrieval systems using inverted indices were presented, and their time complexities were also investigated.

The meta-search approach has been studied in the context of distributed information retrieval [17]. This approach consists of Query Distribution and Result Merging phases. Furthermore, the Document Retrieval problem is divided into two sub-problems: Database Selection problem and Document Selection problem. However, Distributed indexing is not suitable for common Grid architectures. Also, Distributed IR is less efficient in searching.

The use of ontology to overcome the limitations of keyword-based search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 1990s. One way to view a semantic search engine is as a tool that gets formal ontology-based queries from a client, executes them against a knowledge base (KB), and returns tuples of ontology values that satisfy the query. In this view, the information retrieval (IR) problem is reduced to a data retrieval task. While this conception of semantic search brings key advantages already, our work aims at taking a step beyond.

A purely Boolean ontology-based retrieval model makes sense when the whole information corpus can be fully represented as an ontology-driven knowledge base. But, there are well-known limits to the extent to which knowledge can be formalized this way. Boolean search does not provide clear ranking criteria, without which the search system may become useless if the retrieval space is too big.

2.2 e-Learning Grid

Grid computing systems are transparent resource-sharing infrastructures, which can overcome the limitations in traditional e-Learning platforms, such as scalability, interoperability, availability, etc. Grid middleware plays an important role in grid infrastructures, which provides upper applications with transparent resources. Many middleware platforms have been developed, such as Globus Toolkits, Condor, etc. Currently, the trend of grid technology is toward service-oriented grid architecture, which represents the convergence of grid computing and Web services. The distinguished specifications include OGSA/OGSI and its successor, WSRF [18].

Conventionally, e-Learning platforms were developed independently. Therefore, learning objects and functions are platform-dependent, and the collaboration between different systems becomes difficult. The proposal of learning standards, such as SCORM, has improved the sharing of learning content. However, e-Learning still meets many challenges which address sharing and interoperability of learning resources. Grid computing is an appropriate solution to a resource-sharing e-learning infrastructure. Also, grid computing technologies provide possibilities for supporting innovative applications of e-Learning. For example, a medical college can provide students with three-dimensional simulation of human body anatomy using high performance grid computing systems, which is beyond the ability of traditional e-Learning platforms.

More and more effort has gone into the field of e-Learning grid, using grid technologies in the context of e-Learning [19]. There have been researches on e-Learning grid, such as the works by LeGE-WG. Among these, ELeGI (European Learning Grid Infrastructure, 2004-2008) is the most representative project with respect to e-Learning Grid. Its goal is to address and advance current e-Learning solutions through use of geographically distributed resources as a single e-Learning environment.

The Globus Toolkit developed by the Globus Alliance and many others all over the world is an open source software toolkit used for building Grid systems and applications. A growing number of projects and companies are using the Globus Toolkit to unlock the potential of grids for their causes. The Globus Toolkit has become a popular standard for Grid middleware to handle these four kinds of services:

- Resource management: Grid Resource Allocation & Management (GRAM)
- Information Services: Monitoring and Discovery Service (MDS)
- Security Services: Grid Security Infrastructure (GSI)
- Data Movement and Management: Global Access to Secondary Storage (GASS) and GridFTP

GRAM was designed to provide a single common protocol and API for requesting and using remote system resources, by providing a uniform, flexible interface to local job scheduling systems. The Grid Security Infrastructure (GSI) provides mutual authentication of both users and remote resources using GSI (Grid-wide) PKI-based (Public Key Infrastructure) identities. GRAM provides a simple authorization mechanism based on GSI identities and a mechanism to map GSI identities to local user accounts.

MDS was designed to provide a standard mechanism for publishing and discovering resource status and configuration information. It provides a uniform, flexible interface to data collected by lower-level information providers. It has a decentralized structure that allows it to scale, and it can handle static (e.g., OS, CPU types, and system architectures) or dynamic data (e.g., disk availability, memory availability, and loading). A project can also restrict access to data by combining GSI (Grid Security Infrastructure) credentials and authorization features provided by MDS. GridFTP is a high-performance, secure, and reliable data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on FTP, the highly-popular Internet File Transfer protocol.

Grid computing is considered to be an inexpensive and promising alternative to parallel computing, and it extends conventional parallel and distributed computing by utilizing computers on the Internet to compute. Consequently, the rise of grid computing provides a potential solution to the e-learning. Researchers have proposed to utilize data grid technologies to share learning materials. However, these approaches focused on the infrastructure of e-learning platforms [20], and did not address the issue of SCORM-compliant content management.

The Ganglia project grew out of the University of California, Berkeley's Millennium initiative (http://ganglia.sourceforge.net/). The Ganglia project is a scalable open source distributed system for monitoring status of nodes (processor collections) in wide-area systems based on clusters. It adopts a hierarchical; tree-like communication structure among its components in order to accommodate information from large collections of multiple clusters, such as grids. The information collected by the Ganglia monitor includes hardware and system information, such as processor type, load, memory usage, disk usage, operating system information, and other static/dynamic scheduler-specific details.

2.3 Ubiquitous Learning

The rising of u-learning results from the convergence of e-learning and ubiquitous computing. However, this topic is too new to get a well accepted definition. Hwang in his paper compared u-learning systems with m-learning systems, and proposed twelve models for u-learning activities [21]. Illustrative examples in that paper help readers understand what u-learning is like. In [22], existing u-learning applications were categorized as shown in Table 2.1, and a frame-based model was proposed to represent context-aware applications.

Table 2.1 Categorization of ubiquitous learning applications

Application Type	Example
Location-aware learning guidance	• Museum guide [23] • Tour guide [24] • Conference assistant [25]
Correlation-aware collaborative learning	• Japanese polite teaching [26] • Knowledge awareness map [27] • P2P content access and group discussion [28]
Task-aware supervised learning	• Requirement satisfied learning [29]

2.4 Sharable Content Object Reference Model (SCORM)

To share and reuse teaching materials, several standards have been proposed recently. Among these, SCORM (http://www.adlnet.org/) is the most popular standard for learning contents. It was proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997. This standard consists of several specifications developed by IEEE LTSC (Learning Technology Standards Committee, http://ltsc.ieee.org/wg12/), IMS (Instructional Management System, http://www.imsproject.org/), AICC (Aviation Industry CBT Committee, http://www.aicc.org/), etc. The SCORM specifications are a composite of several specifications developed by international standards organizations. In a nutshell, SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed [30, 31]. In SCORM, content packaging scheme is proposed to package the learning objects into standard teaching materials. The content packaging scheme defines a teaching materials package consisting of four components: 1) Metadata, which describes the characteristics or attributes of this learning content; 2) Organizations, which describe the structure of the teaching material; 3) Resources, which denote the physical files linked by each learning object within the teaching material; and 4) the (Sub) Manifest, which describes this teaching material, consisting of itself and other teaching materials. SCORM Metadata refers to the IEEE's Learning Object Metadata (LOM), and describes the attributes of teaching materials. IEEE LOM v1.0 includes nine categories: General, LifeCycle, Meta-Metadata, technical, educational, rights, relation, annotation, and classification.

2.5 Wiki Technology

The term "Wiki" originates from the Hawaiian "wee kee wee kee," which means "quickly." In the domain of computer science, a Wiki is a web-based hypertext system which supports community-oriented authoring, in order to rapidly and collaboratively build the content. The concept of Wiki was proposed by Ward Cunningham in 1995 as the Portland Pattern Repository, to create an environment for co-workers to share specifications and documents for software design.

Wiki is not the first technology for collaboration. Other collaborative technologies, such as discussion boards, have also been widely used for years. Nevertheless, the primary reason why Wiki is so attractive can be attributed to the successful application, Wikipedia [32]. Traditionally, an encyclopedia is built by a number of experts with a tremendous amount of time and money. However, Wikipedia is an innovative project which endeavors to build an online open-source encyclopedia based on Wiki and GNU Free Document License (http://www.gnu.org/licenses/#FDL). This system began in 2001, and the number of English items exceeds 500,000 in 2005. The rapid growth of the Wikipedia system shows that the concept of the Wiki is both viable and feasible. In addition, there are many related projects based on Wiki, such as Meta-Wiki, Wiktionary, Wikibooks, Wikiquotes, to name a few (http://meta.wikimedia.org/wiki/-Complete_list_of_Wikimedia_projects).

The attractive characteristics of Wiki, which favor its use, can be summarized as follows.

- Rapidness. The Wiki pages can be rapidly constructed, accessed and modified, in hypertext form.
- Simpleness. A simple markup scheme (usually a simplified version of HTML) is used to format the Wiki pages, instead of the complicated HTML.
- Convenience. Links to other pages, external sites, and images can be conveniently established by keywords. Moreover, the targets of the keywords, links, need not exist when the links are built. They can be appended later.
- Open Source. Each member can create, modify and delete the Wiki pages at will. Wiki content is not reviewed by anyone before publication, and is updated upon being saved.
- Maintainability. Wiki maintains a version database, which records its historical revision and content, thus enabling version management.

To run a Wiki-based site, it is necessary to deploy a Wiki platform. The requirements of a Wiki platform include editing, links, version management, sandboxes (test-bed), and search functions. Many Wiki platforms have been developed and used in various fields. For example, MediaWiki (http://www.mediawiki.org/wiki/MediaWik), which is used by Wikipedia, is a widely used tool. PBwiki (http://pbwiki.com/), which is developed by PHP languages, is adopted by many libraries.

2.6 Ontology Construction for e-Learning

In recent years, ontology has been used to denote the representative concepts and associated relations among learning materials. The main educational applications of ontologies include content annotation, assessment model, etc. To manage a large number of learning materials, many Learning Content Management Systems (LCMS) have been proposed by means of the ontology-based approach [33-35]. The e-learning was considered as the "learning of organizational memory"; thus the ontology and semantic web technology were used to capitalize the learning knowledge and index the learning resources. In [36, 37], the concept map was applied to visualize the learner's thought and then connect the concepts to the learning contents in LCMS. Thus, the students were able to browse and explore the relevant leaning contents to extend their understanding via the concept maps. In QBLS [38], the ontology with domain model and pedagogical model was proposed to support the design and annotation of the learning resources. Thus, the QBLS can provide the efficient concept information and reasoning mechanisms for the adaptive learning system. It was mentioned that the reusing of existing ontology was a quite conclusive approach. However it also outlined the difficulty of finding acceptable match between different visions of a domain while using the ontology.

Besides the annotation of the learning contents for adaptive learning, in [39, 40] the ontology was used as the intelligent assessment model. After the exam, each learner can have a personal assessment result to indicate the misconception and possible remedial suggestions by referring the structure of concept map. The Salisbury [41] indicated that to achieve efficient performance in a higher level skill, it was required to have some basic perquisite concepts. Thus, the concept effect relationship was applied to annotate the concepts of test items and remedial learning information.

In summary, the effectiveness of the surveyed adaptive e-learning systems and assessment systems were highly dependent on the well defined domain ontology. However, how to construct an acceptable ontology for more complex or larger domain is still a challenging issue.

Ontology building mainly depends on the contribution of domain experts in the knowledge creation activity. Metadata extraction and merging is carried out manually by domain experts. Many tools have been developed for ontology developers to access ontologies, browse them, edit them, and propose modifications. However, some drawbacks do exist. This process is time-consuming and arduously. For this reason, recent researches turned to automatic and semiautomatic (as opposed to manually) ontology construction and maintenance. Automatic techniques for building and integrating ontologies have been studied for many years by the artificial intelligence community. More recently, several techniques specifically aimed at learning ontologies from text corpora or database repositories were proposed. Well-known research approaches to ontology building include [42]:

- Dictionary-based approach [43]: constructs the hierarchy of concepts based on a traditional dictionary, which presents the related concepts of words, including synonyms, etymology, etc.
- Conceptual clustering [44]: Concepts are grouped according to a semantic distance between each other to generate hierarchical relations.
- Association rules mining [45]: The frequency of an association of terms is computed in the text repositories. If the frequency of the association is close to the occurrence of individual terms, the association is transformed in an ontological relation.

- Formal concept analysis [42]: is a method for representation, analysis and management of data and knowledge. It can be used to build a hierarchy of terms and associated relations.

In order to assist the experts constructing ontology, traditional taxonomy-based ontology authoring tools such as Protégé [46], OilEd [47], JOE [48], and SWOOP [49] with Graphical User Interface have been developed to visualize the concepts and their associated relations. These tools are designed for individual user ontology construction with top-down domain analysis process. However, in some dynamic or complex domain, it is costly and time-consuming for individuals to construct an acceptable ontology. Therefore, the collaborative ontology construction approaches are proposed with different incremental ontology learning strategies.

With rapid growth of Web 2.0, one of the emerging visions is the "collective intelligence" of a community of users to contribute their knowledge. The folksonomies mean the user-generated classification keywords, emerging through bottom-up consensus [50]. Folksonomies have not been widely applied to the subject of ontology construction. Although folksonomies are not formal models for knowledge representation, the collaborative wisdom of resource categorization can be a good start point from which effective indices can be built. In this thesis, folksonomies are regarded as an ontology constructed by community. According to Wikipedia experience, we know that communities can provide knowledge more quickly and widely than small group of experts. Therefore recent researches tended to propose the collaborative folksonomy-based ontology construction approaches. Researches such as Ontolingua [51], Collaborative Ontology Building (COB) [52], and OntoWiki [53] construct a web space where members of the ontology developers community can access, browse, edit, and modify ontologies. Each member of community can contribute to ontology with their background knowledge. Although various kinds of knowledge can be rapidly collected from the community members, the system administrator still has to manage the ontology manually. Furthermore, the growth of the amount of data brings more conflicts and noises. The lack of a convergence methodology may result in ontology distortion.

Since the content modeled by the ontology is dynamically changing due to insertion, deletion and update, it should be constructed incrementally and revised periodically. Thus, the ontology maintenance is an important issue. In general, the ontology maintenance approaches include the ontology integration and ontology fusion. Ontology Integration means to maintain the original ontology structure and enriches it by integrating other ontologies. Conventionally, they combine the ontology editor and the online portal to allow experts cooperatively maintain the ontology with different management roles. Researches such as MarcOnt [54], Co-Protégé [55], and CODE [56] are well-known ontology integration approaches. However, there exist some drawbacks. Since the integration tasks have been done manually, it is costly and time-consuming for administrator even with clear management process. Moreover, it is impractical to manage the structure of the ontology manually if the scale of the ontology is large.

Ontology Fusion means to reconstruct a new ontology by fusing others rather than enriching the initial ontology. Traditionally, the automatic or semi-automatic ontology learning approaches have been proposed. Researches such as FCA-Merge [57], PROMPT [58] and Chimaera [59] are ontology fusion approaches. For example, PROMPT constructs the ontology by means of the metadata editing and concepts similarity computation. However, these automatic approaches tempt to be noise sensitive for new domain. If there exists some noise in the ontology, it is difficult to revise the ontology since they should follow the predefined constraints.

2.7 Peer-to-Peer Search

Search in peer-to-peer networks has been a flourishing research topic in recent years. A number of solutions to peer-to-peer search have been proposed (Nottelmann & Fischer, 2007; Parreira, Michel, & Weikum, 2007; Zhu, Cao, & Yu, 2006)[60-66]. Zhu et al. (2006) indicated that one of the difficulties in peer-to-peer search is the lack of global statistical information, thus impeding the straightforward application of the well-known vector space model approach to peer-to-peer networks. Content search was addressed in [67, 68]. In [69-71], information retrieval methodologies were conducted in peer-to-peer networks. In [72], they studied social network.

Most of these researches ignored information retrieval techniques. In addition, the issues of availability and trustworthiness have not been considered. Therefore, teaching material search in peer-to-peer networks requires a more suitable solution. Traditionally, teaching material retrieval approaches can be categorized into: keyword-based search and metadata-based search. In this paper, we retrieve teaching materials based on both the content keywords and metadata.

Issues related to due time setting has not been widely discussed in the literatures. In [73], the problem of sampling peer properties in peer-to-peer networks was addressed. They adopted an adaptive random walk approach to dealing with departing peers. If a query times out, they try another peer from a stack. In [74], techniques for peer-to-peer knowledge management were reviewed. For e-learning applications, that work focused on technologies of file sharing, distributed content networks and collaboration. To sum up, while related researches tend to find out factors which affect the response time, we propose that users can decide whether to wait for results or not.

2.8 Teaching Material Design and Rapid Prototyping

System development is a continuous and fundamental task in many domains, so effective and efficient approaches to system development are demanded by developers. Traditional system development life cycle paradigms, such as the waterfall model, focus on verifying the system requirements during the early stages, and then go through the whole process to generate the perfect product. However, this approach is not suitable when the requirement can not be clearly defined at the beginning. Therefore, evolutionary approaches, such as rapid prototyping, are proposed to alleviate the limitations.

There have not been a clear definition of rapid prototyping (RP) though this technology has been successfully used in many fields, including commercial, military and academic applications [75]. Generally speaking, RP is recognized as technologies which rapidly realize the conceptual model of a final product of system without incurring too much cost. The purpose of RP is to incrementally clarify the requirement and refine the prototype. The techniques for rapid prototyping could date back to the late 1980s and were mainly used in manufacturing. Nowadays, RP is applied for many other domains, even in educational applications. For example, RP has been used to create the lecture contents of the IT SoC certificate program [76].

Many RP methods have been proposed in the literature. Although these procedures are not exactly the same, they conform to the following workflow:

1. initial definition of requirements
2. rapid implementation of a prototype
3. user evaluation and requirement refinement

4. implementation of refined requirements
5. repeat step 3 and step 4 until completion

An early model simply relates traditional design steps to prototypes, consisting of need assessment and analysis, prototype building, prototype utilization, system installation and maintenance [77]. Another example, used to design computer-based courseware, is a three-stage model: analysis, development and evaluation [78].

RP has also been applied to instructional design for both generating high quality product and reducing development time. Jones and Richey reported eight RP applications in instructional design [79], including educational software design, instructional videos, etc. The aforementioned researches illustrate that RP is an effective approach to system development in educational applications. In the case of adaptive e-learning, educators are usually encouraged to rapidly develop various personalized teaching materials. The challenges result from varied requirements and timely pressure, which are similar to the motivation of RP. In this work, we adopt the concept of RP to design teaching materials, facilitating the rapid authoring process.

Due to the advances in information technologies and the requirements of courseware, more and more teachers are able and willing to design their own teaching materials and make them accessible on the Web [80, 81]. In addition, a growing number of large-scale projects aim to construct learning content repositories [82, 83]. For example, in 2002, the National Science Council of Taiwan approved a resolution on the “National Science and Technology Program for e-Learning,” planning to spend $120 million within a 5-year period [84]. These educational contents are mainly based on Sharable Content Object Reference Model (SCORM), which has become a popular standard for creating sharable and reusable teaching materials for e-Learning. With the popularization of e-Learning, how to find and reuse these existing materials becomes an important issue.

Teaching materials are one of the important elements in instruction and learning activities. A large amount of work has been devoted to the methodology of teaching-material design. Some scholars [85] presented an authoring environment for the development of course materials. However, this approach depends on educational experts to participate in the development process. In order to facilitate the reuse of SCORM learning objects and customization of course materials, a system named Teaching-Material Design Center [31] was proposed, reusing e-material from different providers and integrating them for a particular course. Nevertheless, this system still relies on human experts to design a satisfactory teaching material. The recent works of Coffey [86] and Wang [87] addressed the issues of courseware maintenance and enhancement. The former designed a meta-cognitive tool for courseware development and reuse, and the latter presented a course material enhancement process.

Briefly speaking, aforementioned approaches to designing teaching materials are time-consuming. Also, expensive human-resource costs are involved.

Chapter 3 Problem Formulation

This chapter presents the formulation of the general Learning Content Retrieval Problem (LCRP). First, a four-layer learning content model is proposed to describe the viewpoints about organization and representation of learning content. Then, the representation of SCORM-compliant learning content is defined to be used in this thesis. Next, the Learning Content Retrieval Problem is formulated in a general manner, with several parameters to model various e-Learning environments. Finally, the proposed ontology-based content retrieval approach is introduced.

3.1 Learning Content Model

Learning content retrieval can be conducted in different levels of abstraction according to the underlying content models. Based on the viewpoint of content processing, a four-layer content organization model is summarized in Table 3.1. The four layers are: Content Layer, Metadata Layer, Structure Layer and Concept Layer. It should be noticed that the representation of content can include multiple layers. For example, the SCORM standard includes viewpoints of Structure Layer and Metadata Layer.

Table 3.1 A four-layer learning content model

Features / Layer	Viewpoint	Representation of organization	Examples
Concept Layer	Concepts & relations	Ontology	Directory

Structure Layer	Content structures	• Level-wise representation • Tree	• LCMS [88] • SCORM
Metadata Layer	Metadata	Attribute-value pairs	• IEEE LOM • Dublin Core • EXIF
Content Layer	keywords	Vector Space Model	Full-text searching

3.2 Representation of SCORM-Compliant Content

This section presents the definitions of learning content used in this work, which are represented by the learning content model proposed in Section 3.1. Hereafter, the terms: Content Package, Learning Content, Teaching Material and SCORM-compliant document, are used interchangeably.

In the SCORM standard, a Content Package (CP) is defined as a package of learning materials, and a Learning Object Repository (LOR) is a database where the Content Packages are stored. In this thesis, a CP is modeled as a tree to represent the structural information of a CP. To enable content-based retrieval, the well-known Vector Space Model is applied to represent the text content. Also, metadata such as Classification is included in this model of a CP.

Traditionally, similarity is measured by the Vector Space Model (VSM) in information retrieval domain [11, 13]. In the VSM-based model, a document is represented as a vector. In general, a limited vocabulary of keywords is adopted to denote important words in documents. Each element of the vector corresponds to a keyword of the vocabulary. Therefore, the length of the vector for a document is equal to the size of the vocabulary. The value of each element is a weight denoting the importance of the keyword to the document. There are a number of methods to determine the weights. Among them, TF-IDF (Term Frequency – Inverse Document Frequency) [11, 13]is the most well-known method to assign weights. The TD-IDF method is based on two findings. First, words frequently used in a document, except stop words, are important keywords with respect to this document. Second, words frequently appearing in many documents are not important for the purpose of differentiation.

Definition 3.1. Teaching Material.

A **Teaching Material** is a structural document. In Content Layer, the leaf node represents physical content, which is represented by a vector = $<w_1, w_2, \ldots, w_{|V|}>$, where $|V|$ is the size of the vocabulary and the weighting scheme is TF-IDF. Internal nodes represent structural information of teaching materials. In addition, a teaching material is associated with a set of Metadata.

- Metadata, $\{M_k \mid k = 1 \text{ to } m, m \text{ is the number of metadata}\}$.

In Structure Layer, each teaching material is represented by level-wise vectors:

- Level i vector, L_i, where i is an integer greater than 0. The root node is located in Level 0, and the children nodes are located in Level 1, and so on.
- L_i is calculated by averaging the feature vectors of nodes at level i.

■

Example 3.1. Teaching Material.

An example of teaching material is modeled as a trinary tree with three levels, as shown in Figure 3.1. The leaf nodes contain the content, and the internal nodes represent the structural information. In addition, a CP is associated with a set of Metadata. Figure 3.2 illustrates an example of TM representation, where the vector represents weights of 10 keywords, while metadata denotes educational attributes. Hereafter, teaching materials, SCORM-compliant documents, and Content Packages are used interchangeably.

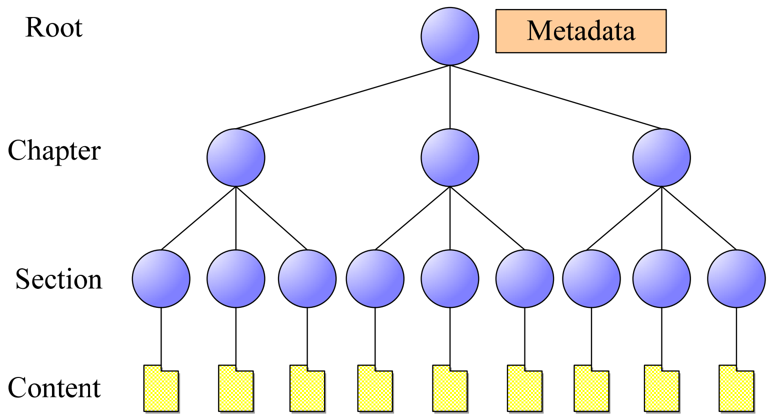


Figure 3.1 The example of a Content Package (CP)

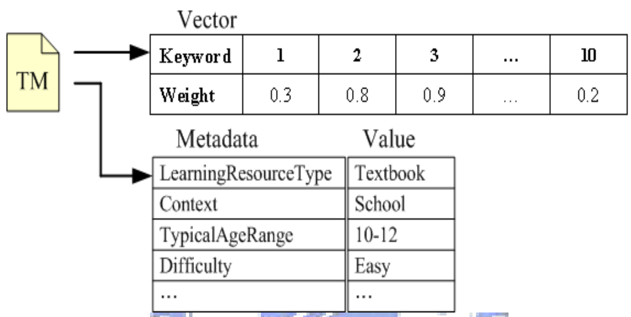


Figure 3.2 Another example of TM representation

Definition 3.2. Learning Object Repository.

A **Learning Object Repository** is a set of Content Packages located in the same site.

3.3 Learning Content Retrieval Problem

This section defines a general Learning Content Retrieval Problem.

Definition 3.3. Learning Content Retrieval Problem (LCRP).

Given a query and parameters, retrieve relevant learning content from repositories in an e-learning environment. The objective is to improve precision and recall. This problem is denoted by LCRP(O, N, P, C_x, C_t, B, T, S), where

O : Ontology representation

N : # Learning Object Repositories (LOR)

P : Probability that LORs are on-line

C_x : Context information

C_t : Content representation

B : Network bandwidth

T : Peer trust

S : Similarity function

■

This general problem formulation can be specialized for various e-learning environments, such as grids, P2P networks, sensor networks, etc. These sub-problems are depicted in Figure 3.3. LCR-sensor denotes the LCRP problem specialized for sensor networks, or ubiquitous learning environments. Specifically, the number of LOR is 1; Context information is available.

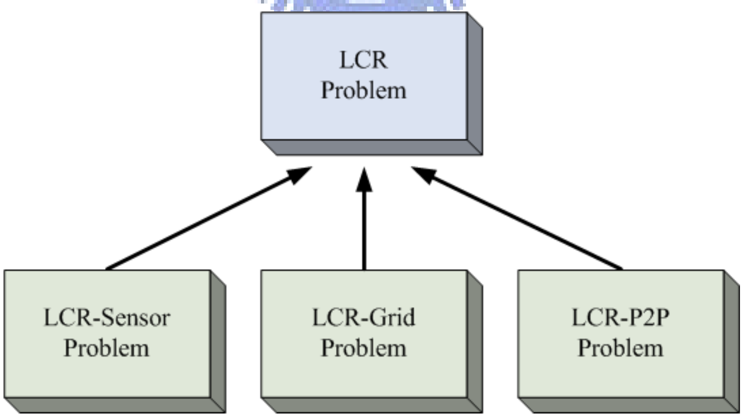


Figure 3.3 Hierarchy of Learning Content Retrieval Problems

3.4 Ontology-based Learning Content Retrieval

Although many researchers have proposed methods of information retrieval, few of them address the issue of learning content management on emerging e-Learning environments. Due to the importance of this problem, an ontology-based framework is proposed to solve this problem, as shown in Figure 3.4. The main idea is to increase precision by ontology-based semantic search, and to reduce search time by ontology-based indexing. The idea of ontology building is based on folksonomy, in order to alleviate the heavy burden of experts and knowledge engineers. This framework consists of three phases. In the ontology building phase, users' folksonomies are clustered into a hierarchical ontology, which can be referenced by the index creation phase, where a bottom-up method is designed to organize learning contents located in different sites on grids, according to the built ontology. An ontology-based global index is then created to facilitate semantic search. Finally, users' queries are interactively verified in the search phase, and desired content is retrieved fast and precisely.

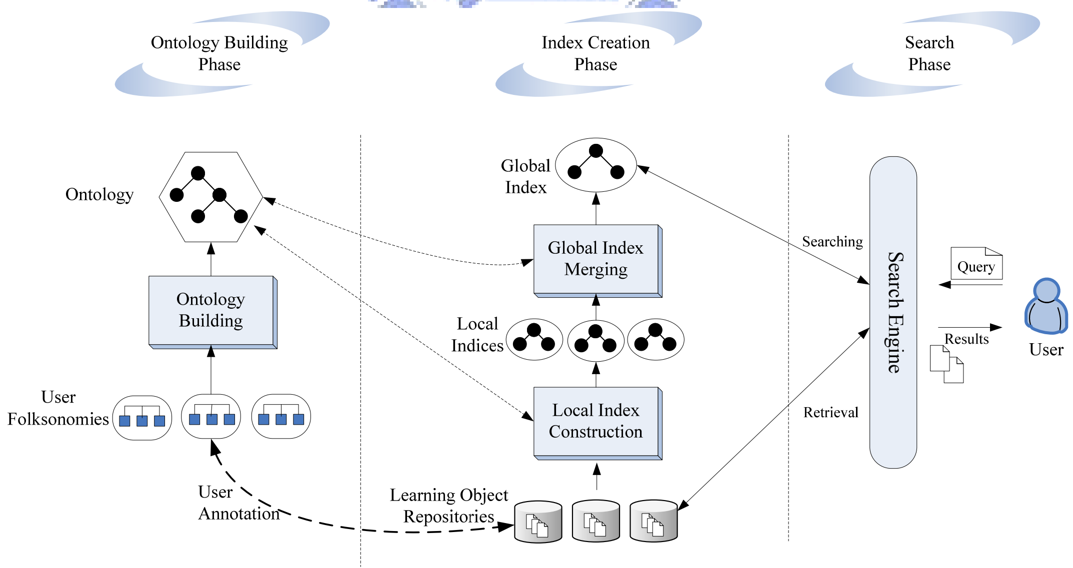


Figure 3.4 The framework of ontology-based content management

Ontology Building Phase

The purpose of this phase is to build an ontology which conceptualizes a collection of learning content. The idea is a bottom-up approach based on folksonomy by integrating user-defined tags. Folksonomies represent users' categorization knowledge, which can be used as initial cluster base to increase the precision of results. The difficulty of using this approach consists in designing a suitable similarity function for the contents. The similarity function should consider textual, metadata and structural information.

Traditionally, the approach to ontology building has been mainly to depend on the contribution of domain experts in the knowledge creation activity. Metadata extraction and merging is carried out manually by domain experts. Many tools have been developed for ontology developers to access ontologies, browse them, edit them, and propose modifications. However, some drawbacks do exist. This process is time-consuming and arduously. For this reason, recent research turned to automatic and semiautomatic (as opposed to manually) ontology construction and maintenance. Automatic techniques for building and integrating ontologies have been studied for many years by the artificial intelligence community. More recently, several techniques specifically aimed at learning ontologies from text corpora or database repositories were proposed.

The built ontology can facilitate semantic search to improve precision of information retrieval. As shown in Figure 3.5, the search engine receives the query submitted by users, and accesses the ontology-based index to find the semantically related documents. Then, the retriever gets these corresponding results, and sends them to the user. With the help of the ontology, the process of information retrieval can overcome the limitations of the traditional keyword-matching method.

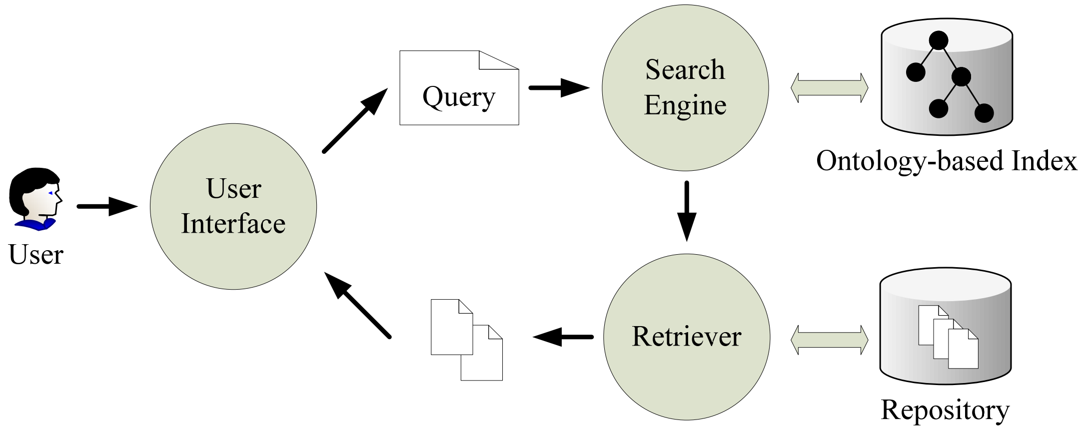


Figure 3.5 The overview of ontology-based information retrieval

Index Creation Phase

Indexing Trees are data structures that are designed to support SCORM documents management on e-Learning environments. The main idea is to reorganize SCORM documents according to their associated Metadata, and to utilize the centralized indexing structure for grid environments.

Search Phase

The Search phase is carried out by the search engine component, which receives queries from users, processes the queries, and presents results to the users. After users specify the desired documents from the returned results, the search engine accesses the site where the documents are stored and retrieves these documents for the users.

When a user submits a query which contains terms don't belong to the Vocabulary, the search engine will suggest some synonymous terms in the Vocabulary. The suggestion is based on a synonym dictionary.

The purpose of this phase is to minimize the time of query processing and content transmission when retrieving SCORM-compliant documents in a grid. To speed up the searching process, our idea is to use a centralized index, which is generated by reorganizing the existing documents based on a bottom-up approach, because this approach is suitable for the master-slave grid model and can effectively collect the information of existing documents from all sites in the grid. Furthermore, the indexing structure stores metadata and structural information, which increases the efficiency and precision of searching. To speed up the transmission process, the other idea is to present the ranked results with estimated transmission time, which is derived from grid monitoring tools. In this way, the document which has high ranking score but has a long estimated transmission times (maybe due to a low-bandwidth link) can be avoided by users.

Chapter 4 Folksonomy-based Index Creation

4.1 Context-aware Retrieval

With the fast development of wireless communication and sensor technologies, ubiquitous learning (u-learning) has emerged as a promising learning paradigm, which can sense the situation of learners and provide adaptive supports to students [21, 89, 90]. Context-awareness is one major characteristic of u-learning, where the situation or environment of a learner can be sensed. Advantages of context-aware learning are two-folded. In the passive aspect, it can alleviate environmental limitations. In the active aspect, it can utilize available resources to facilitate learning.

There are several types of applications for context-aware u-learning. A typical scenario is "learning with on-line guidance," as presented in [21], which considers the "identification of plants" unit of the Nature Science course in a elementary school. The context is in campus, and the human-system interaction is as follows:

- System: Can you identify the plant in front of you?
- Student: Yes.
- System: What is the name of this plant?
- Student: Ring-cupped oak.
- System: Do you see any insect on it?
- Student: Yes.
- System: Can you identify this insect?

- Student: No.
- …

The assumption is that the system is aware of the location of the student, and the nearby plants, by sensor technologies and built-in campus maps.

Retrieval of learning content, hereafter named Content Retrieval (CR), is an important activity in u-learning, especially for on-line data searching and cooperative problem solving. Furthermore, both teachers and students need to retrieve learning content for teaching and learning, respectively. Conventional keyword-based content retrieval schemes do not take context information into consideration, so they cannot satisfy the basic requirement of u-learning, which is to provide users with adaptive results. To support context-aware learning, learning content needs to be provided according to learners' contexts. For example, when a student can not identify an insect in the u-learning course, s/he can access a learning object repository for more information by submitting a query. As we can image, queries are most likely ambiguous and need refinement. If context information can be applied to refine the original query, it will be easier for learners to retrieve relevant content.

As shown in Figure 4.1, we classify the schemes of content retrieval into static and dynamic ones according to the adaptability of the retrieved results. For static CR, the retrieved result only depends on the query, independent of users and contexts. Dynamic CR can be further divided into personalized, context-aware and other schemes, according to the factors that are considered by the adaptive mechanisms of CR. Personalized CR is adapted to subjective factors of learners, such as user profile, preference, etc. In other words, the same query submitted by different persons could result in different results retrieved. Context-aware CR is adapted to objective factors of learners, like time, place, device, activity, peers, etc. Hence, the same query issued in different contexts could get different results.

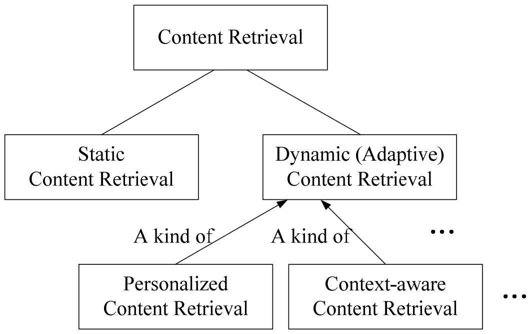


Figure 4.1 Classification of Content Retrieval

To support context-aware CR, the teaching materials stored in the repositories have to be organized according to their contextual information, in order for efficient retrieval. For example, the relevance of a teaching material about fern plants to a given query depends on the learner's location. In an outdoor learning activity, the desired contents are usually those addressing subject materials nearby. However, it is almost impossible to request experts to annotate all these contents with suitable context-aware metadata. Therefore, content annotation based on folksonomies and automatic techniques in a collaborative way is a promising solution.

In well-known folksonomy applications, such as del.icio.us (http://del.icio.us/), Flickr (http://www.flickr.com), etc., folksonomies are organized into a flat structure, which consists of several categories named by user-defined tags. This flat organization is suitable for users to manage their preferences. However, when the size of repositories gets larger and larger, a hierarchical organization becomes a better choice to organize the contents. To bridge the gap of the two structures, we formulate this index creation problem and propose a bottom-up approach to constructing an index from existing folksonomies according to the similarity between tags, which considers metadata and structural information of the teaching materials annotated by the tags. Then, a maintenance mechanism is designed to efficiently update the index. The index creation method has been implemented, and a synthetic learning object repository has been built to evaluate the proposed approach. Experimental results show that this method can increase precision of retrieval. In addition, impacts of different similarity functions on precision are discussed.

The contributions can be summarized as follows. First of all, we propose a folksonomy-based method for index creation, which can reduce the effort required in subsequent work of location-aware content retrieval. With this method, the heavy burden of experts for manually developing concept hierarchy can be significantly alleviated. Second, a similarity function is proposed to increase the precision of folksonomy fusion, which considers more characteristics of learning contents, including metadata and structural information of the teaching materials annotated by the tags. Next, a self-organizing mechanism was designed to balance the number of documents annotated by the tags, which can increase the performance of indexing structures. Finally, the proposed method is implemented and the built index is evaluated. Experimental results reveal that this method can improve the performance of retrieval.

4.2 Folksonomy-based Index Creation Problem

Before the index creation problem is described, some concepts and models are introduced. First, a level-wise structural model of learning contents is presented, which is intended to model the semantic features of different levels of a structural learning content.

A folksonomy is a user generated taxonomy used to categorize and retrieve web content such as Web pages, photographs and Web links, using user-defined labels called tags. Typically, a folksonomy has a flat structure, which consists of several user-defined categories. The folksonomic tagging is intended to make a body of information increasingly easy to search, discover, and navigate over time. One well-known website using folksonomic tagging is del.icio.us. In this work, folksonomies are modeled as follows.

Definition 4.1. Folksonomy

A **folksonomy** defined by a user is a triple of (Tag, Item, Relation), where Tag is a set of tags, Item is a set of content packages and Relation is a relation on Tag×Item. We say that tag t is related to a content package i if i is annotated by t. Each tag is associated with two types of attributes, which are derived from content packages annotated by this tag.

- Level j feature vector, L_j, $(0 < j < \text{Height})$

where L_j = the average of the level j feature vectors of content packages annotated by the tag, and Height is the height of the rooted tree;

- Metadata, $\{M_k \mid k = 1 \text{ to } m, m \text{ is the number of metadata}\}$

where the value of M_k is defined by the most frequent M_k value of the content packages annotated by the tag.

■

The following example illustrates this definition. Figure 4.2 depicts two folksonomies defined by users A and B, respectively. The folksonomy of user A can be represented by

Tag = $\{t_1, t_2\}$

Item = $\{i_1, i_2, i_3, i_4\}$

Relation = $\{(t_1, i_1), (t_1, t_2), (t_1, i_3), (t_2, i_1), (t_2, i_4)\}$

where i_1, i_2, i_3 and i_4 are content packages.

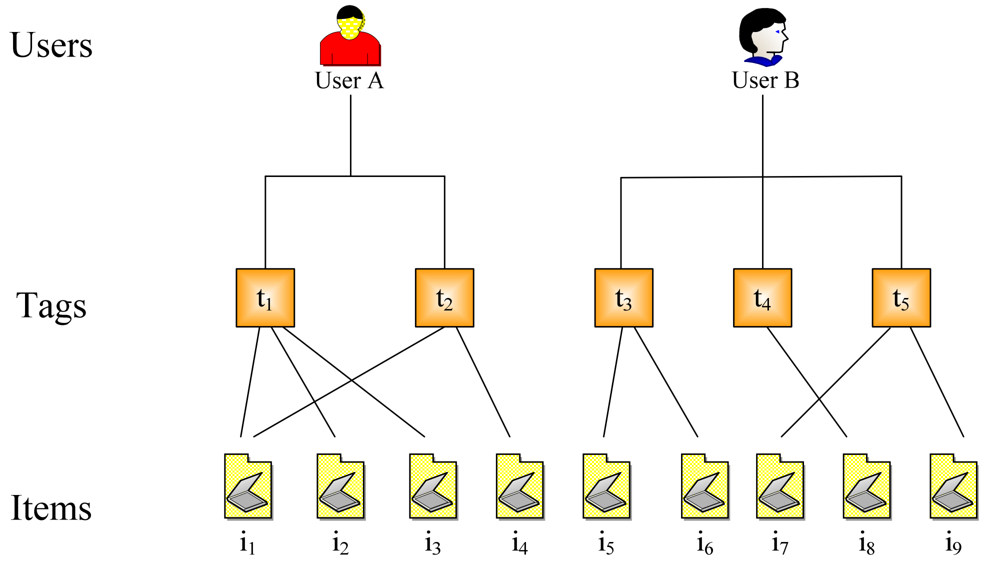


Figure 4.2 The folksonomy of user A

In the field of computer science, an ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to understand the objects within that domain. However, there has not been a commonly acceptable definition for ontologies. The following definitions describe ontology-like indices referred to in this paper. First, the relations, general and specific, are stated.

Definition 4.2. General

A tag *a* is more **general** than a tag *b* if the set of teaching materials annotated by *a* includes those annotated by *b*.

■

Next, we define the index.

Definition 4.3. Index

An **index** is considered to be a rooted tree. The nodes represent concepts in the domain, and the edges represent relations between nodes. A parent node p is more general than its child node c. Each node is associated with a feature vector, which characterizes the semantic meaning of this concept.

Based on the definitions above, the index creation problem is stated as follows.

Definition 4.4. The Folksonomy-based Index Creation Problem (FICP)

Given a collection of content packages and corresponding folksonomies, generate an index. The objective is to improve performance of learning content retrieval.

4.3 Folksonomy-based Index Creation

Our idea to solve this problem is based on the heuristic that existing folksonomies generated by users can be a good starting point from which to construct a location-aware index. While most folksonomies are organized into flat structures, we plan to build hierarchical indices to better organize the learning content. To bridge the gap of the two structures, folksonomies and indices, we propose a bottom-up approach to construct an index from existing folksonomies according to the similarity between tags, which considers metadata and structural information of the teaching materials annotated by the tags.

We design an algorithm to merge two folksonomies into one which is used for subsequent information retrieval. The idea is to make decisions of tag merging according to the similarity of the two tags. The main difficulty is how to choose a suitable similarity function for SCORM-compliant teaching materials, which are characterized by textual content, metadata and structural information. Here, a similarity measure for two tags, a and b, is proposed:

$$Sim(a,b) = (1-\beta)\sum_{i=0}^{Height} \alpha_i \times Sim_i(a,b) + \beta \times Sim_M(a,b) \tag{4-1}$$

where the sum of α_i is equal to one, and $0<\beta<1$. The parameter, α_i, is used to adjust the weighting of level-wise content vector. The parameter β is used to adjust the weighting of metadata similarity. The similarity function consists of two parts:

- Sim_i: level i similarity function, which is cosine function, $0< i <$ Height. The similarity between two vectors $v_k = <k_1, k_2, \ldots, k_{|V|}>$ and $v_p = <p_1, p_2, \ldots, p_{|V|}>$ is measured by the following formula:

$$sim_i = \frac{\sum_{i=1}^{|V|} k_i \times p_i}{\sqrt{\sum_{i=1}^{|V|} k_i^{\,2}} \times \sqrt{\sum_{i=1}^{|V|} p_i^{\,2}}} \tag{4-2}$$

- Sim_M: Metadata similarity function, which is (the number of matched attributes) / (the number of all attributes).

The Folksonomy-based Index Creation Algorithm consists of two steps: initializing the master folksonomy and iteratively merging tags of the other folksonomy into the master folksonomy, listed as follows.

Algorithm 4.1 Folksonomy-based Index Creation Algorithm (AlgFIC)

Input:

*F*1, *F*2: the two folksonomies to be merged

Th_sim: the threshold for comparing similarity

sim: the similarity function

Output:

F: the merged folksonomy

Step 1. Initialization

1.1 Each tag of F_1 and F_2 is represented by the average of its related teaching materials.

1.2 F_1 is assigned as the Master folksonomy, *M*.

Step 2. for each tag t in F_2

2.1 calculate the similarity of t and each tag of *M*.

2.2 let *t_close* be the closest tag in *M* to *t*.

2.3 if the *sim*(*t*, *t_close*) > *Th_sim* then

add the tag *t* into *t_close*

else

add tag *t* into *M* as a new tag

Step 1 of the AlgFIC algorithm evaluates attributes of the input folksonomies. Also, one of the input folksonomy is assigned as the Master folksonomy, which serves as the base for other tags to join. Next, tags of the other folksonomy are merged into the Master folksonomy one after another in Step 2. Finally, a merged folksonomy is generated.

In Step 2.3, a threshold, Th_sim, is set to decide whether to merge the tag or construct a new tag. When the similarity is greater than the threshold, the two tags are merged. Otherwise, a new tag is constructed in the master folksonomy for the dissimilar tag from the other folksonomy.

4.4 The Maintenance Mechanism

After the folksonomies are merged, a master folksonomy is generated. This flat-structured folksonomy can serve as a category or an index for information retrieval. However, when the number of tags in the master folksonomy gets larger and larger, it will be inefficient to access this folksonomy. The purpose of this mechanism is to reduce the search time by maintaining a balanced folksonomy. Therefore, the Hierarchy Organizing problem is to organize the flat folksonomy into a concept hierarchy which is used for subsequent information retrieval. The idea is to setup two thresholds: Th_split and Th_general, to control the process of hierarchy organizing. The former is used to control the size of a tag. When the number of teaching materials annotated by a tag is larger than Th_split, this tag is split into two tags. When the number of tags in a level exceeds Th_general, two tags which have the largest similarity are merged into a tag in the upper level. The main difficulty is how to setup suitable threshold values for Th_split and Th_general. In this work, the two thresholds are setup according to experiments. Different values of the two thresholds are setup, and values with best results are chosen as default values.

A self-organizing mechanism for maintaining the merged folksonomy is designed to attain load balance, which is listed as follows.

Algorithm 4.2 Hierarchy Organizing Algorithm (AlgHO)

Input:

F: the input folksonomy

Th_split: the threshold for splitting a tag

Th_general: the threshold for generalizing tags

Output:

O: the output index

Step 1. (Splitting) for each tag *t* in *F*

1.1 if *t.tm_num* > *Th_ split* then

create a new tag *n_tag*

move half of teaching materials from *t* to *n_tag*

re-compute the attributes of *t* and *n_tag*

Step 2. (Generalization)

2.1 calculate the number of tags in *F*

2.2 if the number is greater than *Th_ general* then

find the most similar two tags

create a new tag *n_tag*
copy teaching materials from the two tags to *n_tag*
assign *n_tag* as a parent node of the two tag
re-compute the attributes of the two tags and *n_tag*

In Step 1.1, t.tm_num means the number of teaching materials annotated by the tag t. Also, a threshold, Th_split, is set to decide whether to split the tag. In Step 2.2, a threshold, Th_general, is set to decide whether to generalize two tags.

4.5 Evaluation

To evaluate the performance of the proposed approach for information retrieval, two experiments are conducted. Two synthetic learning object repositories (LOR) are used in this experiment. The first LOR contains 1,200,000 SCORM-compliant documents [91], which are converted from Web pages related to educational domains. After stop-word cleansing, there remains 2,570,623 distinct index terms. The size of this LOR is around 20 GB. The other LOR contains 2,400,000 SCORM-compliant documents, which are converted from technical papers related to computer science domains. After stop-word elimination, there remains 4,730,384 distinct index terms. The size of this LOR is around 40 GB. A set of 20 queries was prepared for the performance evaluation. For example, one of these queries is "teaching materials about fern plants."

We compare the performance of the proposed method with that of the keyword-based method. The values of α_1 and α_2 are both 0.5. The value of β is 0.3. As shown in Figure 4.3, the proposed method can significantly improve the performance with respect to precision and recall.

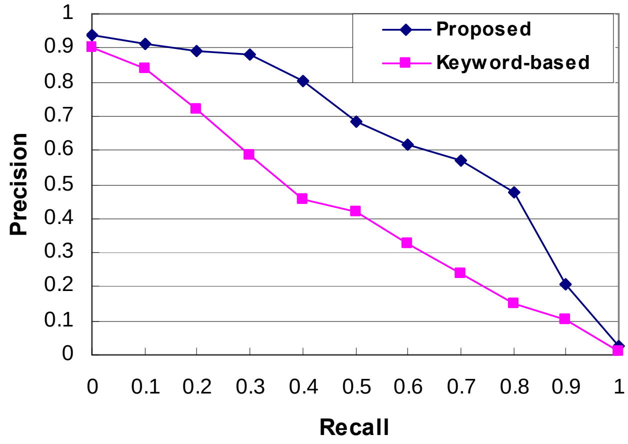


Figure 4.3 Comparison with the keyword-based method

We compare the performance of different similarity functions. As shown in Figure 4.4, the proposed similarity (All) got the best performance. Also, the level-wise similarities have larger impact on performance than metadata. Finally, the level_1 similarity has larger impact than Level_2.

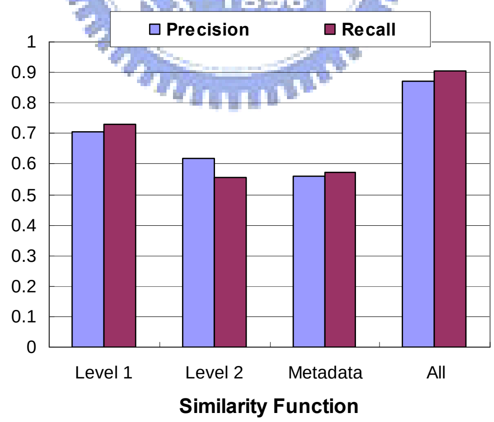


Figure 4.4 Comparison of Different Similarity Functions

The results show that the proposed approach attains better performance than the traditional keyword-based search. The primary reason is that the proposed method uses the location-aware index to conduct semantic search, trying to find various semantic meanings of a given query. For example, a query for “fern plants” would return relevant results about plants which are present nearby, in a location-aware manner. Therefore, the proposed method can get better precision.

Chapter 5 Learning Content Retrieval on Sensor Networks

5.1 Location-aware Retrieval

As described in Chapter 4, the fast development of wireless communication and sensor technologies has made ubiquitous learning (u-learning) emerge as a promising learning paradigm, which can sense the situation of learners and provide adaptive supports to students. Context-awareness is one major characteristic of u-learning, where the situation or environment of a learner can be sensed. Advantages of context-aware learning are two-folded. In the passive aspect, it can alleviate environmental limitations. In the active aspect, it can utilize available resources to facilitate learning.

Learning activities in ubiquitous environments are directed by instructional strategies, which are general approaches, instead of specific methods. As shown in Figure 5.1, instructional activities depend on instructional strategies, which are based on pedagogic theories. Designers of learning activities should utilize the advantages of u-learning environments to realize pedagogic goals.

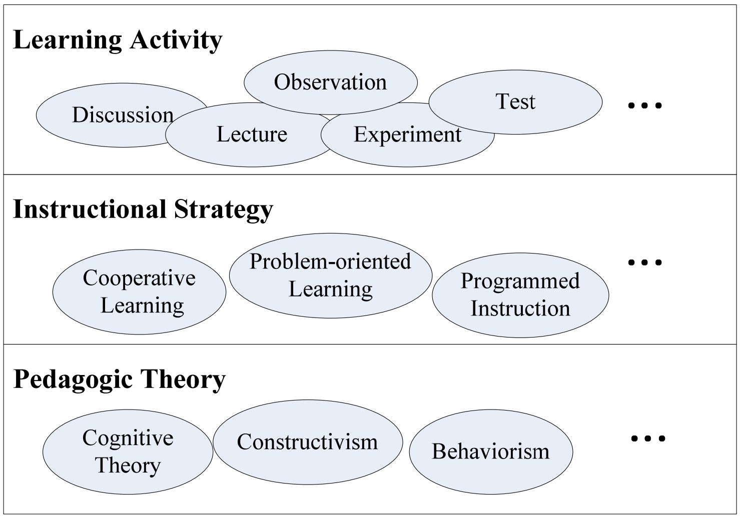


Figure 5.1 Layered relation of instructional activities, strategies and theories

Retrieval of learning content, hereafter named Content Retrieval (CR), is an important activity in u-learning, especially for on-line data searching and cooperative problem solving. Furthermore, both teachers and students need to retrieve learning content for teaching and learning, respectively. Conventional keyword-based content retrieval schemes do not take context information into consideration, so they cannot satisfy the basic requirement of u-learning, which is to provide users with adaptive results. To support context-aware learning, learning content needs to be provided according to learners' contexts. For example, when a student can not identify an insect in the u-learning course, s/he can access a learning object repository for more information by submitting a query. As we can image, queries are most likely ambiguous and need refinement. If context information can be applied to refine the original query, it will be easier for learners to retrieve relevant content.

As shown in Figure 5.2, we classify the schemes of content retrieval into static and dynamic ones according to the adaptability of the retrieved results. For static CR, the retrieved result only depends on the query, independent of users and contexts.

Dynamic CR can be further divided into personalized, context-aware and other schemes, according to the factors that are considered by the adaptive mechanisms of CR. The former is adapted to subjective factors of learners, such as user profile, preference, etc. In other words, the same query submitted by different persons could result in different results retrieved. The latter, context-aware CR, is adapted to objective factors of learners, like time, place, device, activity, peers, etc. Hence, the same query issued in different contexts could get different results.

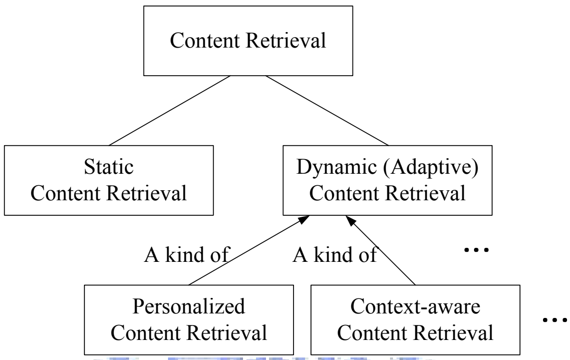


Figure 5.2 Classification of Content Retrieval

Retrieval of learning content is a universal requirement for many learning scenarios, such as Intelligent Tutoring Systems, Zone of Proximal Development, etc. However, each scenario has its own needs for content retrieval. In particular, one important characteristic of context-aware ubiquitous learning is to provide right content to learners at right place and right time. That is, it is desirable for a retrieval system to find the content which is adapted to the learners' context.

In this chapter, we investigate the context-aware learning content retrieval problem, which is to retrieve relevant learning contents from a repository for a given query and context information, improving precision and recall of retrieval. The difficulties are described as follows. First, context information needs to be taken into account for context-aware retrieval. Therefore, traditional information retrieval schemes have to be enhanced to be context-aware. Second, needs for teaching materials are related to pedagogic factors, such as instructional strategies and goals. It is desirable to design a retrieval scheme which is flexible enough to be able to cooperate with various instructional strategies. Third, characteristics of standardized learning content must be considered to improve the accuracy of similarity comparison, such as metadata and structural information. By the way, the acquisition of context information requires extensive deployment of sensors. In this paper, we assume that the module of context acquisition is available, and focus on the first three issues.

To overcome the aforementioned difficulties, our idea is a strategy-driven approach enabled by a knowledge-based query expansion method. First, we intend to expand the original query by acquired context information, in order to retrieve content which is adapted to learners' contextual environments. We adopt the technique of query expansion because most queries in web searching are short and ambiguous, thus needing refinement. Second, we propose a knowledge-based approach to expanding queries based on instructional strategies. According to our observation on ontologies, such as WordNet, basic strategies of query expansion include generalization, specialization, association, their combination, etc. For example, when the educator aims to encourage the learners to think in a higher level, it may be appropriate to adopt an expansion technique of generalization, which offers more general keywords for content retrieval. In this study, we assume that the content about entities near to the learner is more relevant than that far from the learner. For example, when walking by a fern plant, we may want to find some content introducing the fern. Also, this work assumes the instructional strategy and the strategy mapping are defined by experts in advance, which will be our future work. However, we focus on applying retrieval strategies to realize context-aware content retrieval.

Based on this idea, we designed a system consisting of four components: knowledge transformation, query expansion, content retrieval and user interface. In the knowledge transformation component, algorithms of ontology building and rule generation are proposed for teachers to easily construct an ontology from course outlines. The purposes of the ontology are to generate rules of query expansion and to construct taxonomic index of learning object repository. The other three components work as follows. In user interface, the user submits a query, and the context information is extracted by sensors. Next, in query expansion component, candidate keywords are inferred for query expansion, and keywords with entities nearer to the learner are selected. Finally, in content retrieval component, results are retrieved according to the expanded query, and are ranked by a similarity function considering characteristics of learning content. To speed up the searching process, we use a taxonomic index, which is generated by reorganizing the existing documents based on a bottom-up approach [92].

We think the proposed context-aware retrieval method can benefit the ubiquitous learning scenario by providing suitable content adapted to learners' context and instructors' strategies. Experiments have been conducted to show evidence of this claim. First, an experiment involving 30 elementary school students is conducted to show the learning performance affected by the proposed retrieval method. Next, a survey involving 12 elementary school teachers is performed to understand their degree of satisfaction for this system. The results show that this system can speed up the retrieval process, thus facilitating the learning activity. Also, the comments of teachers indicate that this system can effectively find suitable contents adapted to context and instructional strategies.

The contributions of this paper can be summarized as follows. First, we propose a strategy-driven approach to solving the context-aware learning content retrieval problem. This new approach integrates pedagogic requirements and technical solutions, thus benefiting both the parties. Second, a knowledge-based system is designed to support query expansion, which can increase maintainability. Also, the flexibility of the knowledge-based approach facilitates future integration with educational strategies. In addition, the distance-based keyword selection can achieve context-awareness. Third, knowledge transformation algorithms are designed for automatic derivation of ontology and query expansion rules, thus avoiding the difficulty for teachers to manually construct ontology and code rules. Finally, we have built a prototype and experimental results show that this approach can attain accuracy and is helpful to context-aware learning.

5.2 Problem Description

We assume that a context detection module is available, which can extract users' context information. Next, several definitions will be introduced, including teaching materials, learning object repository, a query, context and a similarity function.

The symbols in Table 5.1 are used throughout this paper.

Table 5.1 The notation used in this chapter

Symbol	Description
CP	Content Package
LOR	Learning Object Repository
w_i	Weighting element i of CP vector
V	Set of terms in the vocabulary
Q	Query
v_Q	Vector representing the query
LC_x	The x-coordinate of location context
LC_y	The y-coordinate of location context
$sim()$	Similarity function

We represent the query as a weight vector. Its formal definition is as follows.

Definition 5.1. Query.

A Query is used by a user to specify the TMs s/he wants. Users can express their queries in two forms: keyword-based and metadata-based. A keyword-based query is a vector of keyword weights, which mean the concepts about the desired contents. A metadata-based query is a list of (Attribute, Value) pairs, which describe the properties of TMs.

■

We will now define the notion of similarity between a query and a content package, which means the relevance of the content package to the query.

Definition 5.2. Similarity.

Let Q be a query with query vector v_Q, and TM be a content package. The similarity function is denoted by $sim(Q, TM)$.

■

In order to determine the degree of relevance of a query and a teaching material, the similarity function has to be defined. Conventional similarity functions, such as the cosine function, are not suitable for SCORM-compliant teaching materials which are characterized by textual content, metadata and structural information. Here, a similarity measure Sim between a query Q and a teaching material TM is proposed by combining a keyword-based similarity and a metadata-based similarity. The keyword similarity $Sim_{Keyword}$ adopts a cosine function to measure the text similarity between a query and a TM. The metadata similarity $Sim_{Metadata}$ is defined to be the number of matched attributes divided by the number of all attributes. Therefore, the range of these two similarity terms, $Sim_{Keyword}$ and $Sim_{Metadata}$, are both in [0, 1]. The similarity measure Sim is defined in (1).

$$Sim(Q, TM) = \alpha \times Sim_{Keyword}(Q, TM) + (1-\alpha) \times Sim_{Metadata}(Q, TM) \quad (5\text{-}1)$$

where the factor α, $0 < \alpha < 1$, is used to control the relative weighting of keyword similarity and metadata similarity.

This work focuses on context information related to location. We assume that context information of location can be acquired by extensively deployed sensors and built-in maps.

Definition 5.3. Location Context.

The Location Context is represented by a two-dimensional coordinate, (LC_x, LC_y), where LC_x is the x-coordinate, and LC_y is the y-coordinate. These coordinates correspond to a map of the campus.

■

Based on the definitions above, the Learning Content Retrieval Problem on Sensor Network (LCRP-Sensor) can be defined as follows.

Definition 5.4. Learning Content Retrieval Problem on Sensor Network (LCRP-Sensor).

Given a query and context information, retrieve relevant learning contents from a repository, ranking by a similarity function. The goal is to improve precision and recall of retrieval.

■

5.3 Ontology Building

Ontology building has been considered as a craft rather than an engineering activity. Traditionally, the process of ontology building requires the participation of domain experts and knowledge engineers. Although a number of automatic technologies of ontology construction have been proposed, it is still not easy for teachers, domain experts, to build ontology. Therefore, the ontology building algorithm is proposed for teachers to easily derive an ontology from course outline. In this algorithm, an "expert" means an educator who is also good at knowledge engineering.

Before describing the process of ontology building, we give a general definition of an ontology.

Definition 5.5. Ontology.

An Ontology is a conceptualization of a domain, which is defined as a quadruple $O = (C, A, R, X)$, where

- C is a set of concepts;
- A is a collection of attributes sets, one for each concept;
- R is a set of relations on $C \times C$;
- X is a set of axioms.

■

Example 5.1. A Campus_Plant_Course Ontology.

A Campus_Plant_Course Ontology $OCPC = (C, A, R, X)$ is an ontology where its components are endowed as follows.

C = {"Plant," "Structure," "Fern," …}

A = {Keyword, Type, Location, Level}

R = {"is_a," "related_to"}

X = { IF is_a("A", "B") and is_a("B", "C") THEN is_a("A", "C")}

■

In this study, the ontology is derived from a pre-defined course outline, which reflects the content of the course to be taught by the teacher. The course outline is usually organized by the teacher before the class begins. Without loss of generality, a course outline is defined as a two-level structure, chapters and sections.

Definition 5.6. Course Outline.

A Course Outline is a two-level tree-like representation of the table of content for a course. A course outline consists of a limited number of Chapters, which in turn consists of a limited number of Sections.

■

Example 5.2. A Campus_Plant Course Outline

A Campus_Plant Course Outline COCP can be represented as follows.

Course Name: Plants in the Campus

Chapter 1. Introduction to Plants

  Section 1.1. What is plants?

  Section 1.2. Classification of Plants

Chapter 2. Structures of Plants

  Section 2.1. Flowers

  Section 2.2. Leaves

  Section 2.3. Fruits

Chapter 3. Growth of Plants

  Section 3.1. Budding

  Section 3.2. The Growing Process

Chapter 4. Identification of Plants in the Campus

■

After the course outline is determined, the teacher can derive an ontology from the course outline, following the steps of the Ontology Building Algorithm. This is a special-purpose algorithm, which is designed for constructing the ontology of a course about plants in a campus. Teachers who teach this kind of courses can follow this algorithm to generate an ontology.

Algorithm 5.1 Campus_Plant_Course (CPC) Ontology Building Algorithm

Symbols Definition:

Course_Outline: a two-level course outline

CPC_Ont: an ontology for a course about campus plants

Input: *Course_Outline*

Output: *CPC_Ont*

Step 1: Build the skeleton CPC ontology.

Step 1.1: Extract the Course concept from the name of the course.

Step 1.2: Extract the related Topic concepts from the names of chapters.

Step 1.3: For each Topic concept, create a “related_to” relation to the Course concept.

Step 2: Develop concept hierarchy for the Course concept with “is_a” relations.

Step 2.1: Classify the plants of the campus into several Category concepts.

Step 2.2: For each Category concept, create an “is_a” relation to the Course concept.

Step 2.3: For each category, identify several Instance concepts in the campus.

Step 2.4: For each Instance concept, create an “is_a” relation to its corresponding Category concept.

Step 3: Develop concept hierarchies for Topic concepts.

Step 3.1: For each Topic concept, extract Sub-topic concepts from the names of sections.

Step 3.2: For each Sub-topic concept, create a “sub_topic” relation to the Topic concept.

Step 4: Experts verify the ontology.

The process of ontology building is shown in Figure 5.3. Two types of relationships in the Campus_Plant_Course ontology (CPC Ontology) are described as follows. (1) “is_a” is a generalization relationship, which could be used to describe the concept taxonomies in the class hierarchy. For example, either a wooden plant or a fern plant is a kind of plants. (2) “related_to” denotes that there exists a “related to” relationship between two concepts. For example, we could use the “related_to” relationship to denote that the “plant” subject is related to the “structure” topic.

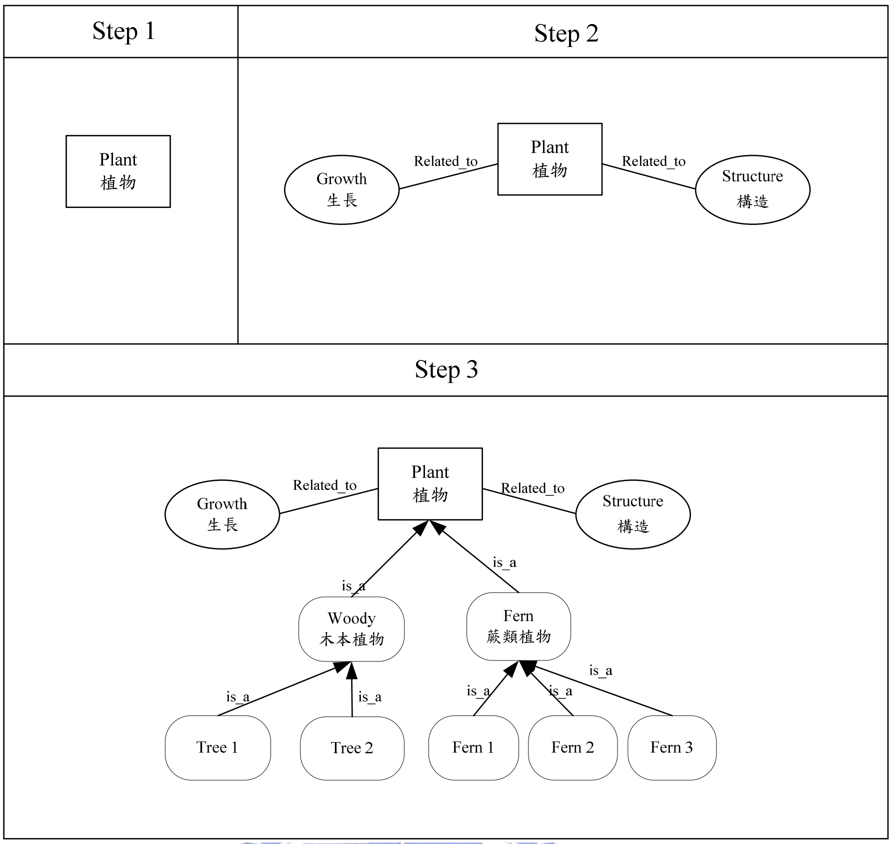


Figure 5.3 Process of ontology building

This section describes the process of constructing the knowledge base. Throughout this paper, we take the plants of a campus for example. Just like the concept of object orientation, we could view all the entities in the campus as concepts and it is natural for us to model the campus using concept hierarchies. For example, a "woody plant" is a concept, and it contains attributes or slots: Leaf_Shape, Leaf_Color, Leaf_Size, etc. Furthermore, people tend to group the knowledge and build structural information when they learn new concepts. The grouped knowledge could be viewed as a bigger concept as well. For example, both woody plants and fern plants are typical types of plants. Hence, the "woody plant" concept inherits the "plant" concept, and there exists a relationship between them. In essence, ontological representation is suitable for communications and natural for human thinking, meanwhile rule-based representation is powerful for machine to manipulate the concepts. As described above, ontology could be used to model the concept hierarchy and relationships between concepts. However, it is not easy to model the behavior of concepts using ontology only. When the problem domain is described clearly and well modeled, it is much easier to build a rule-based expert system because many tools (called expert system shells) can offer assistances. Hence, in practice, rule-based representation is more suitable for building applications. On the other hand, since most real-world applications need complex rules to model, the meaning captured in an ontology for the problem domain becomes very helpful for rule extractions when building complex systems.

5.4 System Overview

We have proposed a context-aware learning content retrieval system based on the knowledge transformation model. As shown in Figure 5.4, the overall system consists of four components:

- User Interface: query input and context detection
- Query Expansion: expanding a query by rule inference
- Content Retrieval: searching and results ranking
- Knowledge transformation: ontology building and rules generation

The flow of the system can be summarized as follows. First, the query and context information are transferred to the Query Expansion component to generate an expanded query. Next, the expanded query is sent to the Content Retrieval component for query processing and retrieving relevant content. Finally, the retrieved results are returned to the user. The search process is carried out by the Query Processor component, which receives expanded queries, processes the queries, and presents results to the users.

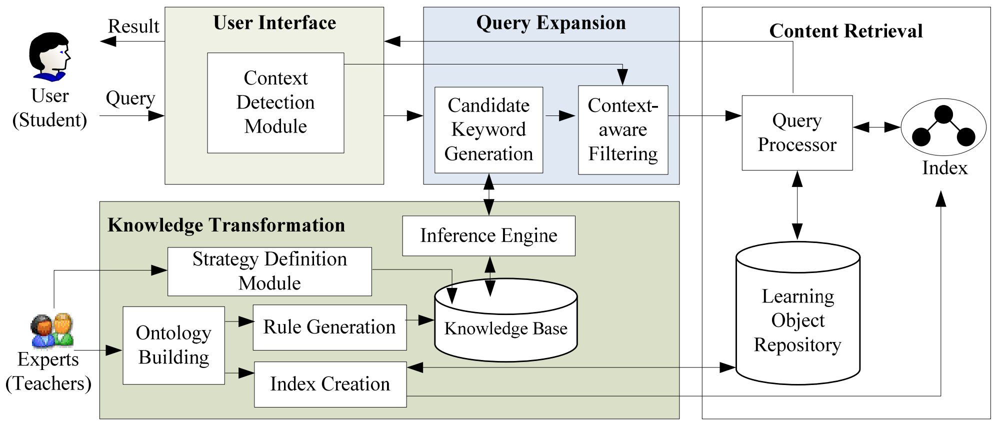


Figure 5.4 Overview of the system

In this work, we assume that the instructional strategies and corresponding retrieval strategies have been defined in the knowledge base by experts. Strategies of query expansion include:

- Specialization: Giving keywords belonging to subordinate concepts
- Generalization: Giving keywords belonging to super-ordinate concepts
- Association: Giving keywords belonging to related topics

Knowledge is represented by rules, describing actions of query expansion. After the generation of ontology, we derive rules for query expansion, using an ontology-driven method. Rules are automatically transformed from the ontology, extracted from two kinds of relations:

- "is_a" relation: For generalization/specialization strategies

For example, IF (Strategy="Specialization") and (term="Fern") THEN expand("Fern 1").

- "related_to" relation: For association strategies

For example, IF (Strategy="Association") and (subject="Plant") THEN expand("Structure").

The algorithm is listed as follows.

Algorithm 5.2 Ontology_to_Rule Algorithm

Input:

CPC ontology

Output:

Rules for query expansion

Step 1: Extract rules by "is_a" relations.

1.1: for each "A 'is_a' B" relation, generate:

IF (Strategy='Generalization') and (term="A") THEN expand("B")

1.2: for each "A 'is_a' B" relation, generate:

IF (Strategy='Specialization') and (term="B") THEN expand("A")

Step 2: Extract rules by "related_to" relations.

2.1: for each "A 'rel' B" relation, generate four rules:

IF (Strategy='Association') and (Subject="A") THEN expand("B")

IF (Strategy='Association') and (Subject="B") THEN expand("A")

IF (Strategy='Association') and (Topic="A") THEN expand("B")

IF (Strategy='Association') and (Topic="B") THEN expand("A")

Step 3: Verify the generated rules by domain experts and ask the knowledge engineers to modify the rules if needed.

To support the pre-defined instructional strategy and enable context-awareness, we propose to divide the process of query expansion into two phases:

- Phase 1: Candidate keywords generation. Based on the retrieval strategy derived from the instructional strategy, candidate keywords are recommended. This phase extends the instructional strategy.
- Phase 2: Context-aware filtering. This phase focuses on filtering the candidate keywords to realize context-awareness. Among a variety of methods, we adopt a distance-based method to determine relevance of keywords.

The main advantage of dividing the process of query expansion is the separation of pedagogic design and technical implementation. While the first phase generates keywords based on instructional consideration, the second phase is related to technical factors. Various technologies can be applied if appropriate. Actions of query expansion can be based on various ideas. The point is how to determine the correlation of two keywords. Conventional methods of calculating keyword correlation include: thesaurus-based, co-occurrence in the corpus and top-ranked in the returned set. Our idea is based on geographical proximity. The expanded keyword is mainly related to the learner's location, instead of the original query.

The distance of two entities is defined as follows.

$$Dist(C, L) = \sqrt{(x_C - x_L)^2 + (y_C - y_L)^2} \tag{5-2}$$

$Dist(C, L)$ means the geographical distance between C and L, where

- C: a concept;
- L: a learner;
- (x_C, y_C), (x_L, y_L): coordinates of C and L.

As shown in Figure 5.5 and 5.6, for specializing "Fern," there are three choices: Fern 1, Fern 2 and Fern 3. We choose the nearest one to the learner.

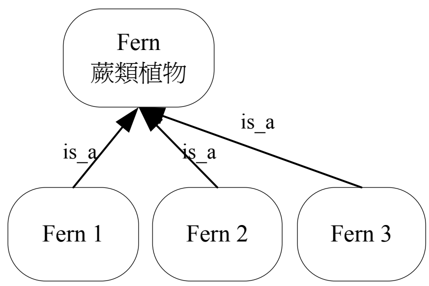


Figure 5.5 The Campus_Plant_Course Ontology (partial)

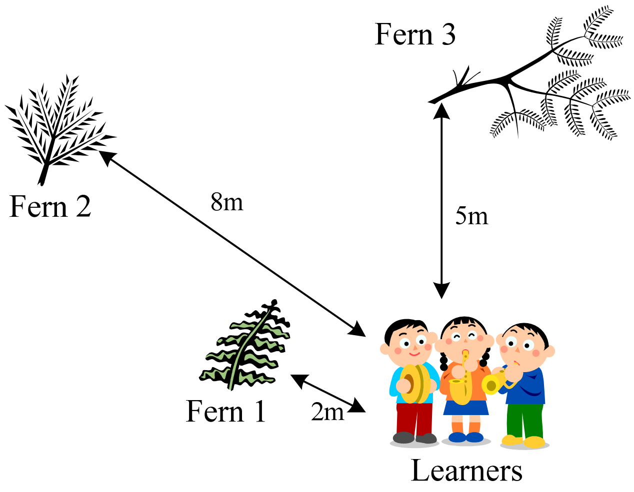


Figure 5.6 Geographical proximity of plants and learners

The User Interface receives users' queries and context information extracted by the Context Detection Module. Another task of the User Interface component is initialization of facts for inference. For example, the fact of the strategy defined by teachers is initialized and loaded into the knowledge for inference. Another important task is to initialize the campus map for default reasoning. The campus map records coordinates of primary plants, buildings and other entities. Although we assume that a lot of sensors have been installed in the u-learning environment, it is probable that the user walks by an area without a sensor. In this situation, coordinates in the campus map can serve as default context information, preventing the inference process from being failed.

5.5 An Illustrative Scenario

This section presents a scenario of a context-aware ubiquitous learning environment, where some communication and context-sensing devices have been installed to enable context-aware retrieval of teaching materials. For an "identification of plants” class of an elementary school, the teacher sets up five learning corners in the campus. The students in the class are divided into five groups, and the teacher arranges an on-line tour guidance for these students. When a student equipped with a PDA goes to the first corner, the system interacts with the student:

- System: Can you identify the plant in front of you?
- Student: Yes.
- System: Can you identify the name of this plant?
- Student: Yes.
- System: What is the name of this plant?
- Student: Fern.
- System: Can you describe the characteristics of a fern?
- Student: No.
- System: Do you need the Search service?
- Student: Yes.
- System: Please input the query:
- Student: fern, characteristic

After the student submits the query, the system conducts the context-aware retrieval, and presents relevant content to the student.

In this scenario, the system retrieves relevant teaching material according to the location context of the student. For example, assume the fern plant in the first corner is some kind of fern, say Fern 1. Assume that the contents in the repository include teaching materials about Fern 1, Fern 2 and Fern 3. The system will present content about Fern 1 to the student in accordance with the context information. The content describing the plant seen by the student will reinforce the impression of the learner, thus increasing the learning effect.

Also, we think the similarity can raise the learning performance. The query reflects the things the student wants to learn. When the learner is provided with more relevant content, the more questions of the learner can be solved, thus increasing the learning effect. Context-aware retrieval can provide students with relevant teaching materials, and shorten the learning process.

5.6 Evaluation

The proposed approach has been implemented and the prototype was provided to several teachers of an elementary school for evaluation. The corpus is composed of SCORM-compliant teaching materials adapted from those of repositories built by Ministry of Education, Taiwan (http://nature.edu.tw). The experiments investigate accuracy of searching. Also, questionnaire is used to understand the degree of satisfaction of users. First, we introduce the prototype and its implementation. Next, experimental results about precision and recall are presented. Finally, results of questionnaires are reported.

To evaluate the proposed approach, we have implemented a web-based prototype, as shown in Figure 5.7. Users can submit queries in this web page. Then, the original query is expanded by the knowledge-based system. After that, the desired contents are retrieved from repositories and returned to the user. The retrieved content packages are ranked by their similarities to the query. All programs are implemented in the Java language.

In this paper, we use DRAMA [93, 94] as an expert system shell because of its client–server architecture and the object-oriented knowledge base structure. The purpose of the DRAMA's server is to load, manage and use the knowledge bases according to the knowledge service that users need. DRAMA's server contains many different rulebases, and provides different APIs for the application servers to connect. The application server employs DRAMA's APIs to provide user-friendly web pages for users to use expert systems. Based on the client-server architecture, it thus becomes very easy for us to develop a KBS for supporting context-aware u-learning. The Lucene search engine (http://lucene.apache.org) was adapted to perform basic keyword indexing, search and retrieval in the prototype. This IR engine was open-source software developed by the Apache Lucene project. Lucene is characterized by the ease to enable applications to index and search documents.

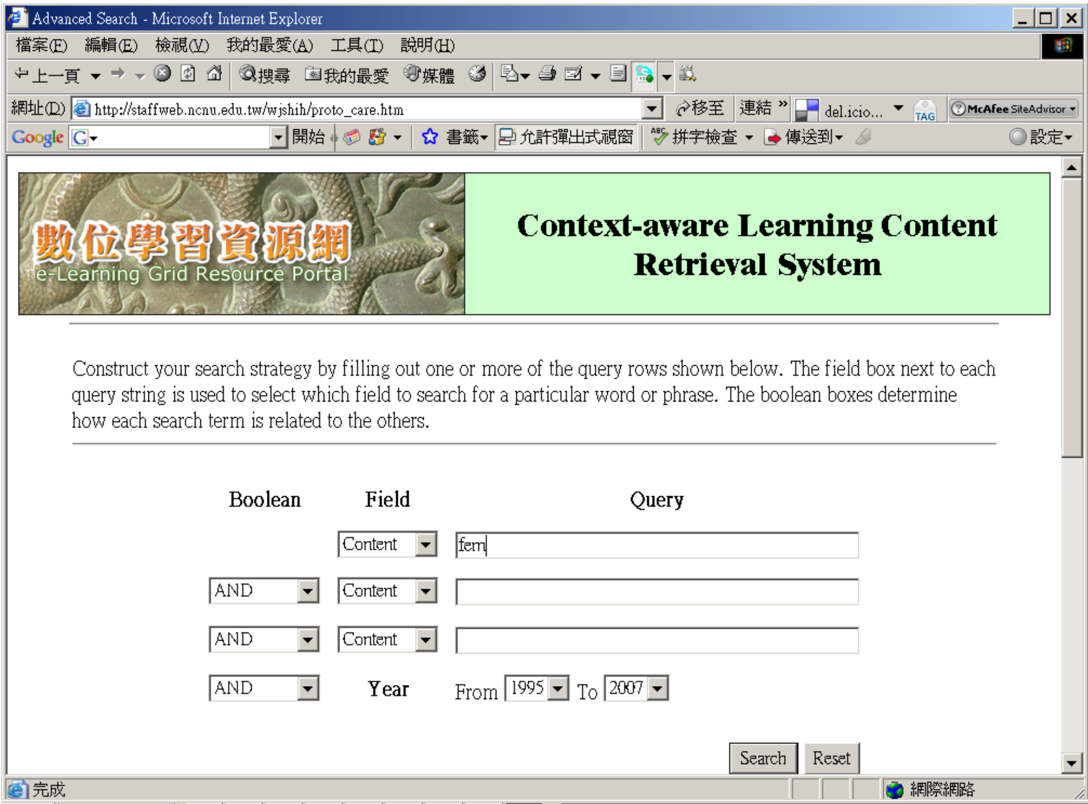


Figure 5.7 Query submission of the prototype

There are nearly one hundred Content Packages in the LOR, which are retrieved and adapted from existing repositories on the Internet, such as http://learning.edu.tw/mainpage.php. Currently, the LOR is only available to elementary school teachers who participate in this evaluation. However, in the near future, we plan to place the prototype and the LOR on the web for public access and large-scale evaluation.

The concept of “relevance” of a TM to a query has to be defined before the retrieval method can be evaluated. It is somewhat subjective, depending on the users, to judge whether a document is relevant to a query or not, especially in a dynamic changing environment. For example, a document about fern is usually thought lowly relevant to a query of “tree.” However, if the user is standing in front a fern plant, this document will be highly relevant to a query. In this experiment, we try to define an objective, reliable and fair measure of relevance. First, we define that a document and a query is relevant if their similarity value, calculated by (1), exceeds the threshold value assigned in advance in the system. Next, for 5 places in the campus, we manually generate a query for each of them. Then, the documents in the repository are manually judged its relevance to the 5 queries by teachers and experts.

In this experiment, we use two well-known metrics of information retrieval, precision and recall, to measure performance of the proposed approach. We define precision and recall as follows.

$$Precision = R\_ret \big/ Ret \tag{5-3}$$

$$Recall = R\_ret \big/ R\_LOR \tag{5-4}$$

where

- R_ret is the number of relevant documents in the retrieved documents;
- Ret is the number of retrieved documents;
- R_LOR is the number of all relevant documents in all repositories.

To optimally set up the parameters of the similarity function, α, is not easy. In this experiment, we set α= 0.5. This setting means equal weighting for keyword similarity and metadata similarity. Other setting of the parameter will be considered in future work.

In this section, three experiments/surveys are conducted to address, respectively,

- the performance of knowledge-based query expansion;
- the learning effect of using context-aware retrieval; and
- the loading of teachers for participating in ontology building.

The purpose of this first experiment is to evaluate the performance of knowledge-based query expansion, with respect to the precision and recall of context-aware content retrieval. The participants are 15 fourth-grade students invited from the Ai-Lan elementary school, Nantou, Taiwan. After usage training for one week, they use this system to retrieve relevant teaching material. At each learning corner in the campus, every student submits a query. Then, the original query is transformed by the system into a specialized query and a generalized query, respectively. Next, the three queries are used to retrieve teaching materials, and the precision and recall values are calculated. Figure 5.8 illustrates the average precision and recall values for the original query, the specialized query and the generalized query. Results show that the expanded queries perform better than the original one. The main reason may be that the original queries are usually short and ambiguous, thus insufficient to represent the intention of users. In addition, we found that generalization can improve recall, and specialization can improve precision. This is consistent with the cognition of precision and recall.

At first glance, the results shown in Figure 5.8 seem to conflict with conventional information retrieval practice which indicates the trend of decreasing precision along with the rise of recall. In fact, there is no conflict. To generate a typical precision-recall plotting for a given query, the set of retrieved documents are listed. Next, the precision and recall are calculated accumulatively from the first document to the last document. Finally, these pairs of precision and recall values are plotted in a 2-dimentional coordinate figure, with the precision against the recall. In this kind of figures, the following trend usually holds: as the recall rises, the precision decreases. However, Figure 5.8 is not obtained in this manner. Given a collection and a query, the precision and recall values are calculated by (5) and (6). That is, while we focus on one query, conventional experiments illustrate results of multiple queries. Therefore, it is possible for the expanded query to outperform the original query in both precision and recall.

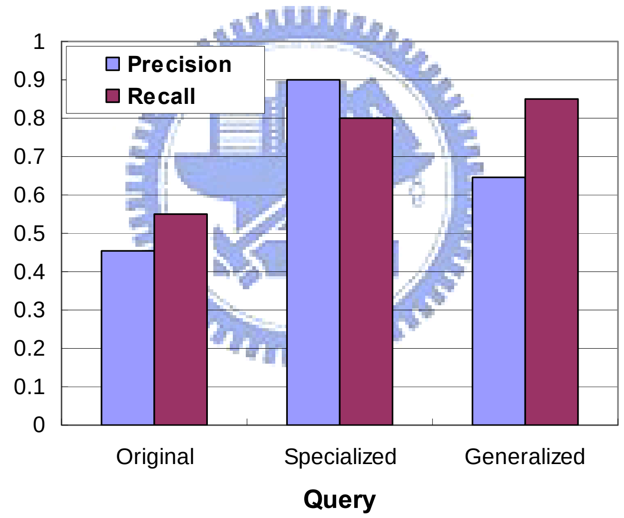


Figure 5.8 Comparison of Precision and Recall

Next, a survey was conducted with respect to the 15 students of the first experiment. The questionnaire contains questions of four categories, each of which has five questions. The purpose is to obtain their comments on the retrieved contents and learning effect. A five-point Likert scale with anchors ranging from strongly disagree (1) to strongly agree (5) is used for this survey. The mean value and standard deviation (SD) are calculated for each category, as shown in Table 5.2.

Table 5.2 The result of the survey for students

Category No.	Questions	Mean	SD
1	Satisfaction of the user interface of the system	2.9	1.06
2	Satisfaction of the expanded query	4.2	0.94
3	Satisfaction of the retrieved contents and learning effect	4.3	0.89
4	Willingness to use this system for learning	3.7	1.03

For questions of Categories 1 and 4, the deviation of user satisfaction is slightly larger than other categories. The reason may be that the participants are not all familiar with the usage of the system and some English interfaces. Some participants comment that they are not used to study on computers. However, some participants appreciate this idea and like to retrieve relevant TMs.

The results of Categories 2 and 3 show that most participants are satisfied with the expanded queries and the retrieved contents. Most students agree that the expanded queries can enhance their original queries, and the retrieved contents are helpful for their learning. In summary, the system can help students efficiently find relevant teaching materials for learning. However, the user interface has to be improved in order to attract more users.

Finally, a survey was conducted to address the loading and benefits of teachers for participating in the ontology building process. In this study, 5 teachers are solicited from the Ai-Lan elementary school to collaboratively construct the knowledge base. Each teacher was asked five questions to obtain their comments on teacher-involved ontology building. A five-point Likert scale with anchors ranging from strongly disagree (1) to strongly agree (5) is used for this survey. The mean value and standard deviation (SD) are calculated for each question, as shown in Table 5.3.

Table 5.3 The result of the survey for teachers

No.	Questions	Mean	SD
1	Do you think it suitable for teachers to build ontology?	4.2	0.84
2	Do you think teachers are capable of building ontology?	4.6	0.55
3	Do you think it takes too much time for teachers to build ontology?	3.8	0.84
4	Do you think the collaborative approach to ontology building can save time?	4.2	0.84
5	Do you think it beneficial for teachers to build ontology?	4.6	0.55

Table 5.3 shows that most teachers think the ontology building task time-consuming. However, they are willing to participate in this task for the purpose of improving learning effects of students. Furthermore, these teachers think that the task of ontology building is beneficial. Their comments indicate that they have a deeper insight into the course content through the process of ontology building, thus improving the quality of instruction. In addition, ontology facilitates the sharing and reuse of instructional expertise. One ontology can be easily adapted to similar courses for other teachers. This convenience results from the flexibility and maintainability of knowledge-based technologies. Also, they agree that the collaborative approach can alleviate the loading and speedup the building process.

The findings of experimental results are interpreted in terms of related literatures. First, we address the application of query expansion to context-aware retrieval. Query expansion has been investigated to improve recall of information retrieval and disambiguate the meaning of queries. A large number of researches have been devoted to this topic [95], but query expansion has not been widely applied to the e-learning domain, not to mention the ubiquitous learning. In [92], the authors indicate that efficient retrieval of teaching material can facilitate the learning process, thus fostering the learning effects. In this study, the result of the first experiment implies that the proposed query expansion can improve the performance of context-aware retrieval, which can support the ubiquitous learning scenario.

Currently, researches of context-aware ubiquitous learning focus on the acquisition and modeling of context information, such as location, temperature, humidity, etc. In this study, we find that the students agree that the context-aware retrieval tool is helpful for their learning. They comments that the system can retrieval context-related contents, which saves their time in finding relevant references.

The knowledge engineering such as ontology building has been thought as tough work, which can only be dealt with by domain experts and knowledge engineers. Therefore, researches of ontology building focus on automatic or semi-automatic approaches [93], in order to alleviate the burden of the builders. This study proposes a teacher-guided approach to building simple ontology for educational usage. The result of survey shows that it is feasible for teachers to provide their expertise and help the system generate a simple ontology based on a pre-defined course outline.

In addition, the contribution of this paper with respect to pedagogical feasibility and keyword association is clarified. First, to derive the important components of the proposed approach, the ontology and the knowledge base, an automatic approach is adopted, in order to alleviate the burden of teachers and domain experts. For the proposed setting, teachers guide the automatic construction of the ontology by providing a course outline and instances of concepts existing in the campus. Next, domain experts verify the ontology generated by the proposed algorithm. We do not intend to require teachers to manually build the rules and ontology. Instead, teachers provide their knowledge about course outline and campus context, which is not difficult, and the system will transform them into ontology and rules. In this way, the proposed approach will be pedagogically feasible. Second, the proposed approach is distinct from existing searching engines. In particular, the latter, such as Google, can not adaptively find results according to users' context. Although collaborative filtering techniques have succeeded in suggesting contents according to keyword association mining from users' query logs, the context-awareness has not been integrated into this technology.

Chapter 6 Learning Content Retrieval on Centralized Grids

6.1 Retrieval on e-Learning Grids

Due to the advances in information technologies and the requirements of courseware, more and more teachers are able and willing to design their own teaching materials and make them accessible on the Web. In addition, a growing number of large-scale projects aim to construct learning content repositories. For example, in 2002, the National Science Council of Taiwan approved a resolution on the "National Science and Technology Program for e-Learning," planning to spend $120 million within a 5-year period. These educational contents are mainly based on Sharable Content Object Reference Model (SCORM), which has become a popular standard for creating sharable and reusable teaching materials for e-Learning. With the popularization of e-Learning, how to find and reuse these existing materials becomes an important issue.

Grid computing systems [2, 3] are transparent resource-sharing infrastructures, which can overcome the limitations in traditional e-Learning platforms, such as scalability, interoperability, availability, etc. Also, grid computing technologies provide possibilities for supporting innovative applications of e-Learning. For example, a medical college can provide students with three-dimensional simulation of human body anatomy using high performance grid computing systems, which is beyond the ability of traditional e-Learning platforms. Therefore, more and more effort has gone into the field of e-Learning grid, using grid technologies in the context of e-Learning. Among these, ELeGI (European Learning Grid Infrastructure, 2004-2008) is the most representative project with respect to e-Learning Grid [4]. Its goal is to address and advance current e-Learning solutions through use of geographically distributed resources as a single e-Learning environment.

With the promising development of e-Learning grid, there will be a great demand to find desired teaching materials from multiple repositories in the e-Learning grid. A traditional approach is to implement a meta-search engine on top of these distributed repositories [17]. When the meta-search engine receives a query, it distributes the query to the local repositories, and then collects the returned results and presents them to users. However, this traditional approach doesn't consider characteristics of SCORM and grid computing to speed up the retrieval process. For example, SCORM-compliant documents are associated with nine categories of metadata. When processing a query with a restrictive filter, such as "documents about insects," the meta-searching approach will search the whole databases. In fact, a better approach is to search only the "insects" category, which will obviously reduce the searching time. For another example, because common grid middleware is implemented with a tightly-coupled master-slave model, it is efficient to use centralized management schemes. The traditional meta-searching approach is not designed for this model, and is inefficient in the process of query dispatching and results collecting, which involves synchronization and communication overhead. Hence, there is an urgent need to design a tailor-made approach to be suitable for e-Learning grids.

The goal of this work is to minimize the time of query processing and content transmission when retrieving SCORM-compliant documents in a grid. To speed up the searching process, our idea is to use a centralized index, which is generated by reorganizing the existing documents based on a bottom-up approach, because this approach is suitable for the master-slave grid model and can effectively collect the information of existing documents from all sites in the grid. Furthermore, the indexing structure stores metadata and structural information, which increases the efficiency and precision of searching. To speed up the transmission process, the other idea is to present the ranked results with estimated transmission time, which is derived from grid monitoring tools. In this way, the document which has high ranking score but has a long estimated transmission times (maybe due to a low-bandwidth link) can be avoided by users.

In this chapter, we address the problem of retrieving SCORM-compliant documents on e-Learning grids. To efficiently manage documents in the grid, we have designed an indexing structure named Taxonomic Indexing Trees (TI-trees). A TI-tree is based on an existing taxonomic schema and has two novel features: 1) reorganizing documents according to the Classification metadata such that queries by classes can be processed efficiently and 2) representing each document by a term-weighting vector, where the term-weighting includes structural information. In this way, an appropriate weighting scheme can be used to utilize the structural information in the SCORM-compliant documents. In addition, the cost of constructing, merging and maintaining TI-trees is not expensive, but the benefits are significant. The overall process of this approach is composed of a Construction phase and a Search phase. In the former, a local TI-tree is built from each Learning Object Repository. Then, all local TI-trees are merged into a global TI-tree. In the latter, a Grid Portal processes queries and presents results to users. After the user specifies the desired documents, the Portal retrieves this document from the target site for the user.

The primary contribution of this chapter is the design of an efficient approach to retrieving SCORM-compliant documents on grid environments. To the best of our knowledge, this topic has not been addressed in previous work. Second, real-time information of network bandwidth is taken into consideration for estimation of transmission time on grid environments. Finally, we have built a prototype on a grid test-bed, and experimental results show that this approach is scalable with respect to storage requirements and time complexity.

6.2 Problem Description

Typically, grid infrastructure is built with a suite of middleware. Common middleware platforms, such as Globus Toolkits [3] and Condor (http://www.cs.wisc.edu/condor/), are based on a master-slave paradigm. Hence, we represent the grid by a master-slave model. Also, real-time information is included to model the dynamic grid.

Definition 6.1. Grid.

A grid is a star graph $G = <N_G, E_G>$ that consists of a finite set of node N_G, and a finite set of edges E_G. N_G represents the set of sites in the grid. One node $P_0 \in N_G$ is specified as the master node, and other nodes are slave nodes. Each edge in E_G connects the master node and a slave node. The real-time information of network bandwidth between the Master and Site i at time t is denoted by $BW_i(t)$

■

An example grid is shown in Figure 6.1. In this graph, P_0 is the master node and the other 3 nodes, P_1, P_2, and P_3, are slave nodes. In addition, there is a virtual communication link L_i connecting the master node and the slave node P_i. The Grid Monitoring Tool can provide real-time network bandwidth information.

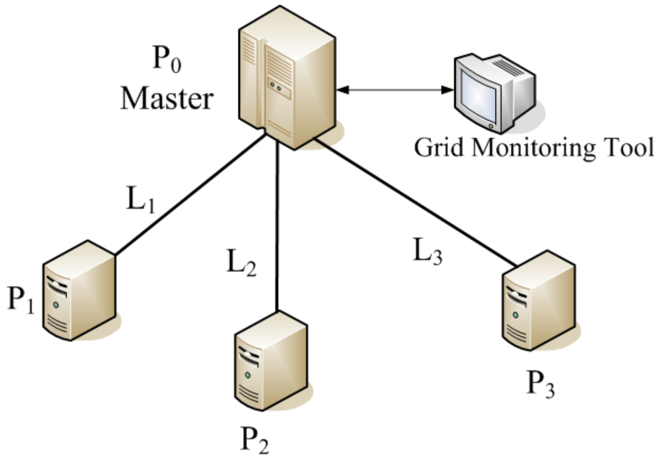


Figure 6.1 An example grid

In this paper, the metadata contains only one piece of information: Classification. An existing taxonomic schema is assumed, and the value of metadata means the classification ID of the CP in this taxonomic schema.

The Dewey Decimal Classification (DDC) system, devised by library pioneer Melvil Dewey in the 1870s and owned by OCLC since 1988, provides a structure for the organization of library collections. Now in its 22nd edition, the DDC is the world's most widely used library classification system [96]. In this paper, we simplify the DDC for the underlying taxonomic schema which the index tree is based on.

We represent the query as a two-tuple: a weight vector and a class id. Its formal definition is as follows.

Definition 6.2. Query.

A Query is defined by $Q= <v_Q, C>$, where

$$v_Q = \{q_1, q_2, \ldots, q_{|V|}\} \tag{6-1}$$

and C is a class ID.

■

We will now define the notion of similarity between a query and a content package, which means the relevance of the content package to the query.

Definition 6.3. Similarity.

Let Q be query with query vector vQ and class C, and CP be a content package. The Similarity sim(Q, CP) is defined by:

$$sim(Q,\ CP) = v_Q \cdot v_{CP} \qquad (6\text{-}2)$$

where the operation is inner product of vectors.

■

Based on the definitions above, the Grid SCORM Document Search Problem (GSDSP) can be defined as follows.

Definition 6.4. Learning Content Retrieval Problem on Grid (LCRP-Grid).

Given a query Q = <v_Q, C>, $|N_G|$ learning object repositories, a similarity function *sim*, and an integer $k > 0$, the Learning Content Retrieval Problem on Grid is to find the top k relevant *CP*s, whose class is *C*, with respect to the query. The objective is to minimize the query processing time.

■

6.3 Bottom-Up Index Creation

Taxonomic Indexing Trees are data structures that are designed to support SCORM documents management on grid environments. The main idea is to reorganize SCORM documents according to their associated Metadata, and to utilize the centralized indexing structure for grid environments. The attributes and operations of Taxonomic Indexing Trees are described as follows.

The Dewey Decimal Classification (DDC) system, devised by library pioneer Melvil Dewey in the 1870s and owned by OCLC since 1988, provides a structure for the organization of library collections. Now in its 22nd edition, the DDC is the world's most widely used library classification system. In this paper, we simplify the DDC for the underlying taxonomic schema which the index tree is based on.

Definition 6.5. Indexing Tree.

An Indexing Tree (I-tree) is a rooted tree having the following properties:

- Every node x has the following fields:
  - *x.name*: Class Name
  - *x.id*: Class ID
  - *x.num*: Number of Documents
  - *x.inv*: pointer to the Inverted Index
- Let x be a node in an I-tree. If y is a child of x, then there exists a relation IS-A between x and y. That is, y IS-A x, which implies that y is a specialization of x.

An example I-tree is shown in Figure 6.2. The hierarchical relation is based on a taxonomic schema. The root class "Natural Ecology" is divided into two classes: "Terrain" and "Insect." In turn, the two classes are also divided into two sub-classes, respectively. The right child of the root node is "Insect," and its Class ID is ".3." We can see that this node has 2 CPs, and its right child has no CP.

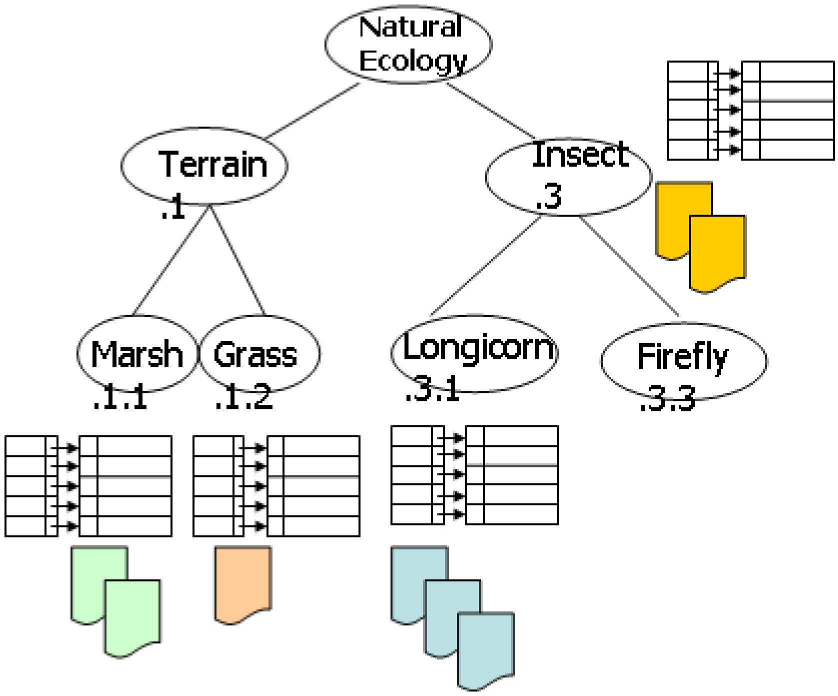


Figure 6.2 An example index tree

The operations supported by I-trees include Searching, Construction, Merging and Insertion, which are described as follows.

- Search (T_0, C)

Given an I-tree T_0 and a class name C, this function returns a pointer x to a class node in T_0 such that $x.id = C$, or NIL if no such class belongs to T_0. A common operation performed on an I-tree is searching for a query in a sub-tree. If the class of the sub-tree is not specified by users, the searching process will start from the root.

- Construct (*LOR*)

Given a Learning Object Repository LOR, a Construct operation returns an I-tree. The resulting I-tree represents the content packages in the LOR.

- Merge (T_1, T_2)

Given two I-trees T_1 and T_2, a Merge operation returns a new I-tree which includes CPs of T_1 and T_2.

- Insert (T_0, CP)

Given an I-tree T_0 and a content package CP, an Insert operation inserts the CP into I-tree T_0.

The query processing is described in Algorithm 6.1.

Algorithm 6.1 Query Processing

Input:

T_0: the index to be searched

Q: the query

C: the Class ID associated with the query

k: the number of desired CPs

Output: A sorted list of pointers to the top k relevant CP

Step 1. Search (T_0, C)

Step 2. Scan the term index for each query term.

Step 3. Get the posting lists corresponding to each query term.

Step 4. Scan the posting lists to accumulate similarity scores for each document.

Step 5. Extract the top k documents.

Step 6. Sort the results.

Due to the dynamic grid environments, the network bandwidth might fluctuate. The results are ranked by similarities, and each result is associated with its estimated transmission time. Therefore, the search engine presents the searching results to users with estimated transmission time. The transmission time is estimated by the following formula:

$$T_{trans} = \frac{S_j}{B_i} \tag{6-3}$$

where T_{trans} is the estimated transmission time, S_j is the storage size of TM j, and B_i is the network bandwidth of Site *i*.

The Construction phase is carried out in backend, and is transparent to users. First, each site in the grid constructs its index from the local LOR. Next, this local index is transferred to the search engine by GridFTP middleware. Then, these indices gathered from all sites are merged into a global index at the search engine.

The operation Construct (LOR) is implemented by Algorithm 6.2.

Algorithm 6.2 Index Construction

Input:

LOR: the LOR to be converted to an index

Output: An index

Step 1. Initialize class nodes of a new index.

Step 2. For each class node, allocate a term index.

Step 3. Scan the LOR. For each TM, update the attributes, then scan each term and update the postings.

After all the local indices are built and transferred to the search engine, the global index is generated as follows, which indices all the documents in the grid. The operation Merge (T_1, T_2) is implemented by Algorithm 6.3.

Algorithm 6.3 Index Merging

Input:

$\boldsymbol{T_1}$, $\boldsymbol{T_2}$: the two indices to be merged

Output: An index

Step 1. Initialize class nodes of a new index.

Step 2. For each class node, allocate a term index.

Step 3. Scan each local index. Update the attributes of nodes and the postings of the global index.

The process above is for creating a complete index on a LOR. However, when a new document is added into a local LOR, it is inefficient to rebuild the whole index. The operation Insert (T_0, CP) is implemented by Algorithm 6.4.

Algorithm 6.4 Index Insertion

Input:

$\boldsymbol{T_0}$: the index to be inserted to

CP: the CP to be inserted

Output: An index containing CP

Step 1. Search the local index for the target class node, and update the posting.

Step 2. Transfer the attributes to the search engine.

Step 3. Search the global index for the target class node, and update the posting.

The Search phase is carried out by the search engine component, which receives queries from users, processes the queries, and presents results to the users. After users specify the desired documents from the returned results, the search engine accesses the site where the documents are stored and retrieves these documents for the users.

When a user submits a query which contains terms don't belong to the Vocabulary, the search engine will suggest some synonymous terms in the Vocabulary. The suggestion is based on a synonym dictionary.

The purpose of this phase is to minimize the time of query processing and content transmission when retrieving SCORM-compliant documents in a grid. To speed up the searching process, our idea is to use a centralized index, which is generated by reorganizing the existing documents based on a bottom-up approach, because this approach is suitable for the master-slave grid model and can effectively collect the information of existing documents from all sites in the grid. Furthermore, the indexing structure stores metadata and structural information, which increases the efficiency and precision of searching. To speed up the transmission process, the other idea is to present the ranked results with estimated transmission time, which is derived from grid monitoring tools. In this way, the document which has high ranking score but has a long estimated transmission times (maybe due to a low-bandwidth link) can be avoided by users.

6.4 Analysis

Analysis of storage requirements and time complexity can help us understand the performance of TI-trees. Because analytical results heavily depend on methods of implementation, fix-length attributes of nodes and index terms are represented by static array, and Posting Lists are implemented with Linked Lists, as shown in Figure 6.3.

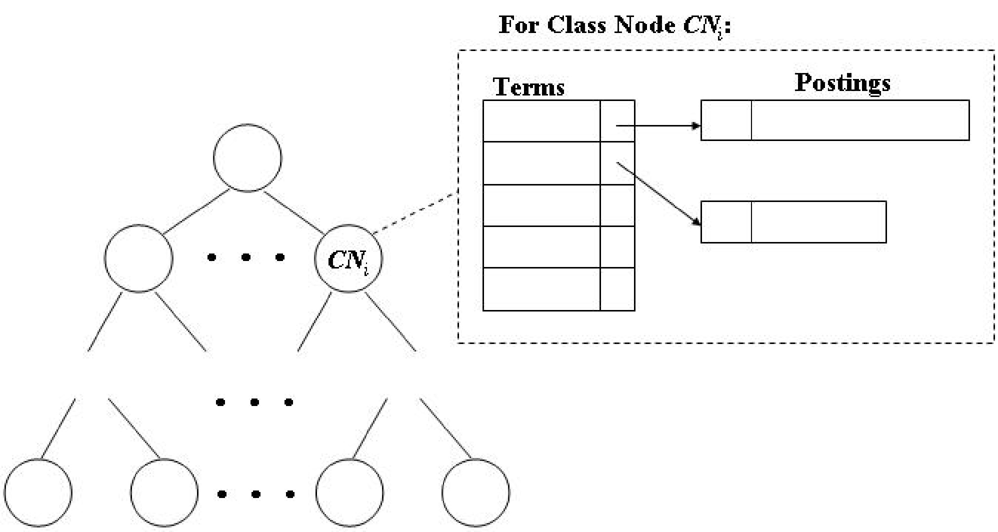


Figure 6.3 TI-Tree Indexing implemented with array

To simplify our discussion, we assume that the whole structure of a TI-tree is stored in main memory, without disk accesses.

- **Storage Requirement**

The storage requirement of a TI-tree structure, STI-tree, is the product of the number of class nodes and the sum of storage required for a class node, term index, and posting lists.

$$S_{TI\text{-}tree} = N_C \times (S_C + S_{inv} + S_{post}) \tag{6-4}$$

The number of class nodes, N_C, depends on the assumed taxonomic schema. The storage requirement of fix-length attributes in a class node is represented by S_C. Hence, S_C is a constant.

Term index is straightforwardly implemented with static array.

$$S_{inv} = |V| \times S_{term} \tag{6-5}$$

where $|V|$ is the number of terms and S_{term} is the space for a term entry.

Posting lists are also implemented with static array.

$$S_{post} = N_p \times S_{ent} \tag{6-6}$$

where N_p is the number of postings and S_{ent} is the space for a posting entry.

The storage requirement of this implementation is $O(N_C*(S_C+|V|+N_P))$. Assume N_C is a constant, $N_C << |V|$, and $N_C << N_P$. Then, $S_{TI\text{-}tree} = O(|V|+N_P)$, which is asymptotically equal to the storage requirement of the traditional inverted index scheme.

- **Time Complexity of Search Phase**

We denote the time cost of step i as Ti. Costs for query processing are analyzed as follows.

Step 1. A query contains a class ID, which specifies a class node in the TI-tree. To search the global TI-tree for the target class node, a pointer is pointed to the root node, and this pointer traverses the path down to the target node. Let D denote the depth of the target node. $T_1 = O(D)$.

Step 2. Term index is implemented with a static array. To find a term, this array is accessed sequentially from the first element to the next element. In worst case, $T_2 = O(|V|)$.

Step 3. Each term q_j in the query is considered in turn to get the corresponding posting lists. $T_3 = O(|Q|)$.

Step 4. Scan the posting lists to accumulate similarity scores for each document. This step involves N_P reading and writing. $T_4 = O(N_P)$.

Step 5. Extract the top k documents. $T_5 = O(N_{CP})$.

Step 6. Sort the results with array implementation. $T_6 = O(k^2)$.

The running time of this implementation is $O(D + |V| + |Q| + N_P + N_{CP} + k^2) = O(|V| + N_P + N_{CP})$. The time complexity of query processing for the TI-tree approach is $O(|V| + N_P + N_{CP})$, which is linear.

- **Time Complexity of Construction Phase**

Costs for constructing a local TI-tree are as follows.

Step 1. Initialize class nodes of a new TI-tree. Hence, $T_1 = O(N_C)$.

Step 2. For each class node, allocate a term index and posting lists. $T_2 = O(N_C * (|V| + N_P))$.

Step 3. Scan the LOR. For each CP, scan each term and update the postings. $T_3 = O(N_{CP})$.

The running time of this implementation is $O(|V| + N_P + N_{CP})$, which is linear.

Costs for generating a global TI-tree are as follows.

Step 1. Initialize class nodes of a new TI-tree. $T_1 = O(N_C)$.

Step 2. For each class node, allocate a term index and posting lists. $T_2 = O(N_C * (|V| + N_P))$.

Step 3. Scan each local TI-tree. Update the postings of the global TI-tree. $T_3 = O(|N_G| * N_P)$.

The running time of this implementation is $O(|V| + N_P + |N_G| * N_P) = O(|V| + |N_G| * N_P)$. Assume that $|N_G| << N_P$. Then $O(|V| + |N_G| * N_P) = O(|V| + N_P)$, which is linear.

Costs for inserting a new CP are as follows.

Step 1. Search the local TI-tree for the target class node, and update the posting. $T_1 = O(D+N_P)$.

Step 2. Transfer the attributes to the Portal. Assume $T_2 = O(1)$.

Step 3. Search the global TI-tree for the target class node, and update the posting. $T_3 = O(D+N_P)$.

The running time of this implementation is $O(D+N_P)$, which is linear.

6.5 Implementation

A metropolitan-scale Grid computing platform named TIGER Grid (standing for

Taichung Integrating Grid Environment and Resource) has been built in a project leaded by Tunghai University. The TIGER grid interconnects computing resources of universities and high schools and shares available resources among them, for investigations in system technologies and high performance applications. This novel project shows the viability of implementation of such a project in a metropolitan city. The TIGER Grid computing platform consists of three universities and two high schools, all located in Taichung, Taiwan. The project of constructing such a grid infrastructure was to share computational resources of each institution.

A grid test-bed based on part of the TIGER Grid has been built, using the following middleware:

- Globus Toolkit 4.0.2
- Mpich library 1.2.6

The master node is at Tunghai University (THU), and the slave nodes are located at Tunghai University (THU), Providence University (PU), Li-Zen High School (LZ), and Hsiuping Institute of Technology School (HIT). Figure 6.4 shows our grid test-bed, and the specifications of the grid test-bed are shown in Table 6.1. Figure 6.5 shows the real-time status of the grid test-bed acquired by the monitoring tool.

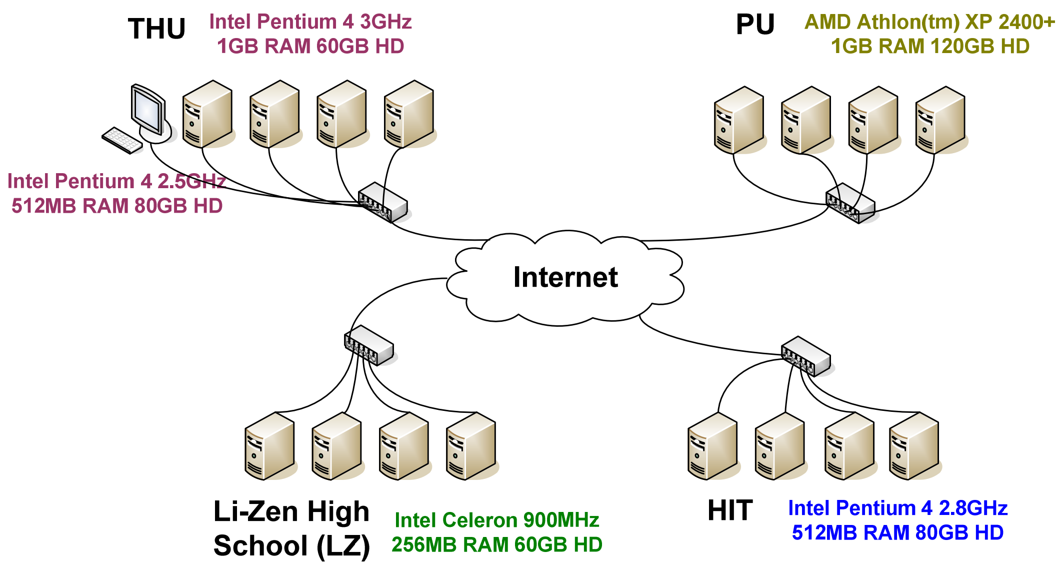


Figure 6.4 The logical diagram of our grid test-bed

Table 6.1 Specifications of computing resources on the test-bed

Site	Host	CPU Type	Clock (Mhz)	RAM	NIC	Linux Kernel	Globus Version
THU	delta1	Intel Pentium 4	3001	1GB	1G	2.6.12	4.0.1
	delta2	Intel Pentium 4	3001	1GB	1G	2.6.12	4.0.1
	delta3	Intel Pentium 4	3001	1GB	1G	2.6.12	4.0.1
	delta4	Intel Pentium 4	3001	1GB	1G	2.6.12	4.0.1
LZ	lz01	Intel Celeron	898	256MB	10/100	2.4.20	4.0.1
	lz02	Intel Celeron	898	256MB	10/100	2.4.20	4.0.1
	lz03	Intel Celeron	898	384MB	10/100	2.4.20	4.0.1
	lz04	Intel Celeron	898	256MB	10/100	2.4.20	4.0.1
HIT	gridhit0	Intel Pentium 4	2800	512MB	10/100	2.6.12	4.0.1
	gridhit1	Intel Pentium 4	2800	512MB	10/100	2.6.12	4.0.1
	gridhit2	Intel Pentium 4	2800	512MB	10/100	2.6.12	4.0.1
	gridhit3	Intel Pentium 4	2800	512MB	10/100	2.6.12	4.0.1
PU	hpc09	AMD Athlon XP	1991	1GB	1G	2.4.22	4.0.1
	hpc10	AMD Athlon XP	1991	1GB	1G	2.4.22	4.0.1
	hpc11	AMD Athlon XP	1991	1GB	1G	2.4.22	4.0.1
	hpc12	AMD Athlon XP	1991	1GB	1G	2.4.22	4.0.1

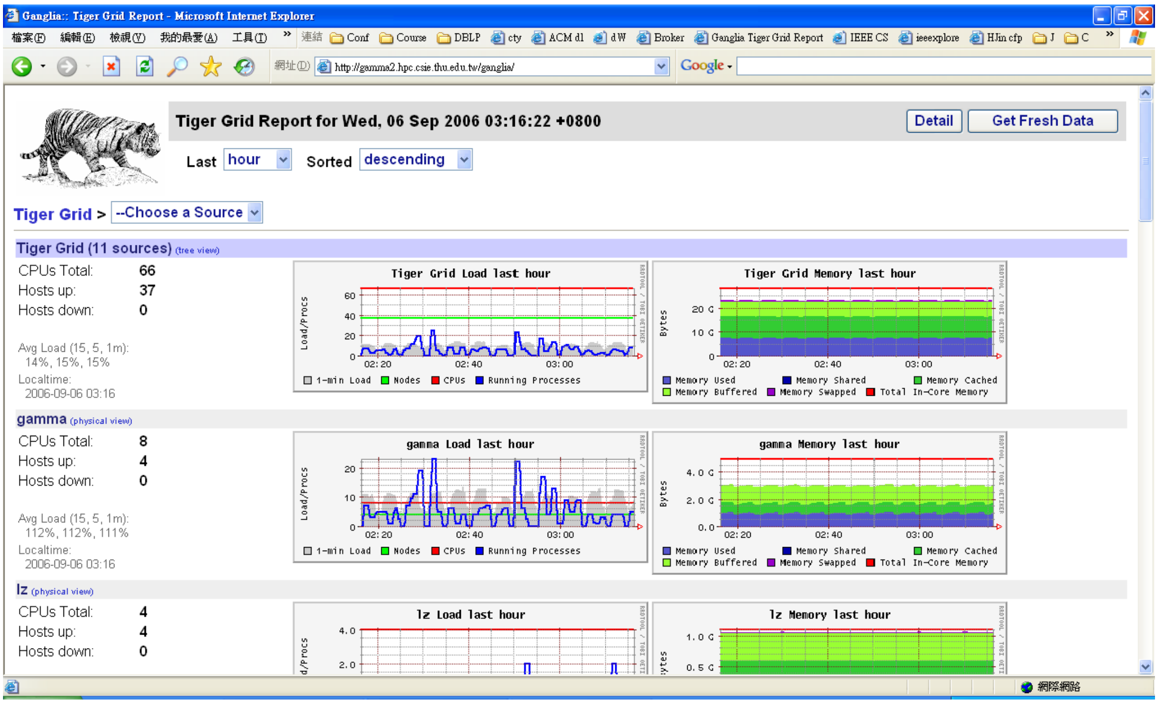


Figure 6.5 The snapshot of the monitoring tool on the TIGER Grid

6.6 Evaluation

To evaluate the proposed approach, we have implemented a web-based prototype, named "e-Learning Grid Resource Portal," for the e-Learning grid test-bed. As shown in Figure 6.6, users can submit queries in this web page. Then, the TI-tree is accessed to find related teaching materials. After that, the desired contents are retrieved from local sites and returned to the user. The retrieved content packages are ranked by their similarities to the query. Also, the estimated transmission time of each content package is listed. All programs are implemented in the Java language. Because of the dynamic nature of grid environments, experiments are repeated 5 times, and the average values are reported. All experiments are conducted in dedicated mode.

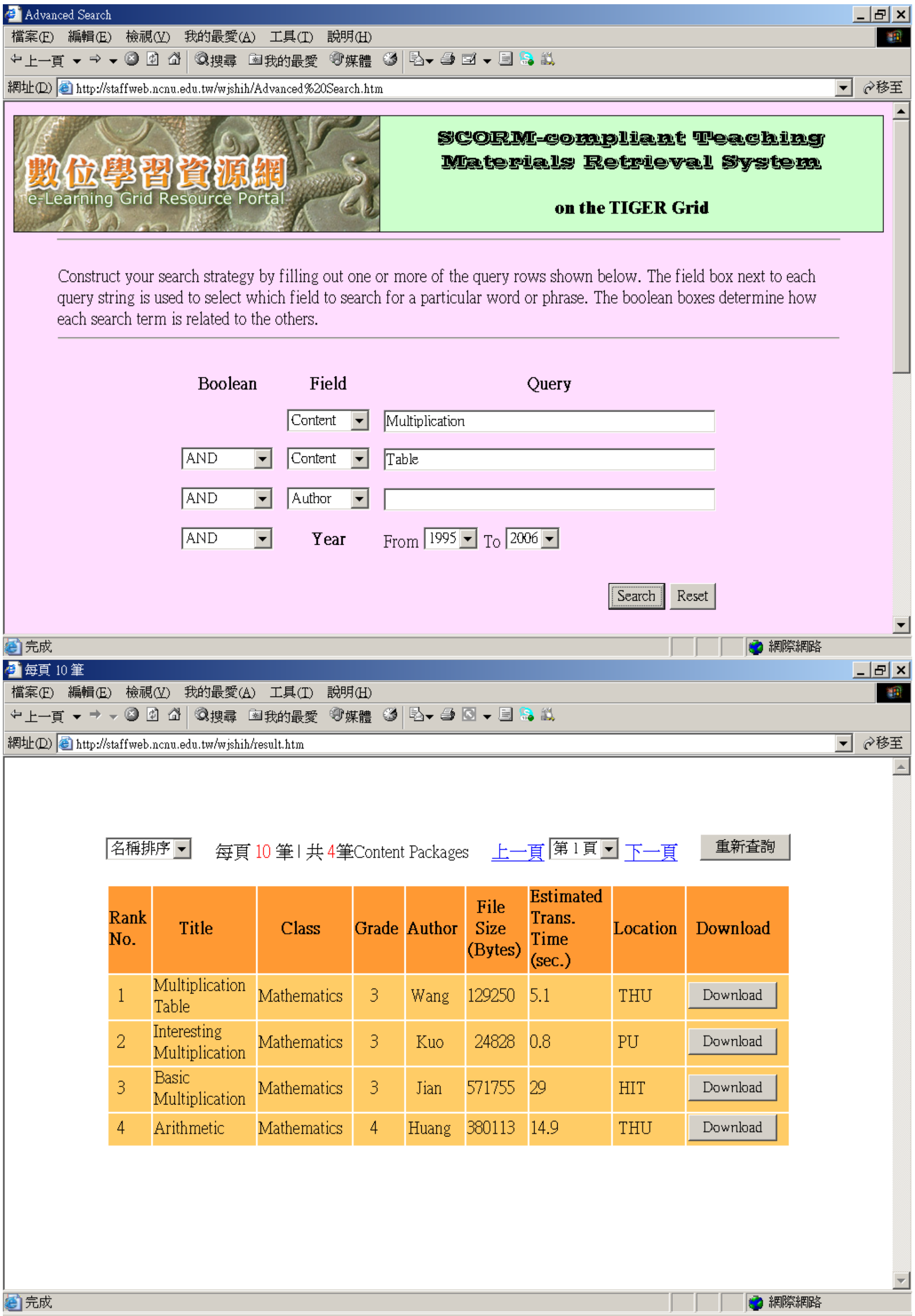


Figure 6.6 Query submission and results presentation of the prototype

We adopt the method in [88] to generate synthetic learning contents. All the synthetic teaching materials were generated by setting three parameters: 1) V: The dimension of the feature vectors of the teaching materials (TM); 2) D: the depth of the content structure of the TM; 3) B: the upper and lower bounds of the included sub-section for each section of the TM. Four synthetic LORs are built, with V = 15, D

= 3, and B = [5, 10], and stored in the grid, which are listed in Table 6.2. The four LORs contain 2,400,000 SCORM-compliant documents, which are converted from Web pages related to educational domains. After stop-word cleansing, there remains 4,730,384 distinct index terms. The total size of these LORs is around 40 GB.

Some of the Content Packages in the LORs are retrieved and adapted from existing repositories on the Internet, such as http://learning.edu.tw/mainpage.php. Currently, these LORs are only available to primary-school teachers who participate in this evaluation. However, in the near future, we plan to place the prototype and the LORs on the web for public access and large-scale evaluation.

Table 6.2 Specifications of learning object repositories

Site	No. of Documents	Size (GB)
THU	900,000	15
LZ	300,000	5
HIT	600,000	10
PU	600,000	10

We simplify the Dewey Decimal Classification (DDC) system to serve as the taxonomy. After refinement, there were 10, 50, and 250 class nodes in levels 1, 2 and 3 of the TI-trees, respectively.

We have implemented two algorithms for this experiment. The first algorithm is a traditional meta-searching approach, named DIR (Distributed Information Retrieval). The other is the approach using TI-trees, named GIR (Grid Information Retrieval).

- **Experiments on Execution Time**

The purpose of this experiment is to show that the proposed GIR is faster than the traditional approach DIR, by comparing the average time of query processing. 20

queries generated randomly were used to compare the performance of DIR and GIR. Figure 6.7 illustrates the processing time of queries for implementations of DIR and GIR. The average query processing time of GIR is 1.1 second less than that of DIR. The main reason may be that GIR using the TI-tree approach can effectively speed the searching processing. However, DIR dispatches the query to four sites, and merges the results returned from these sites, thus slowing down the process.

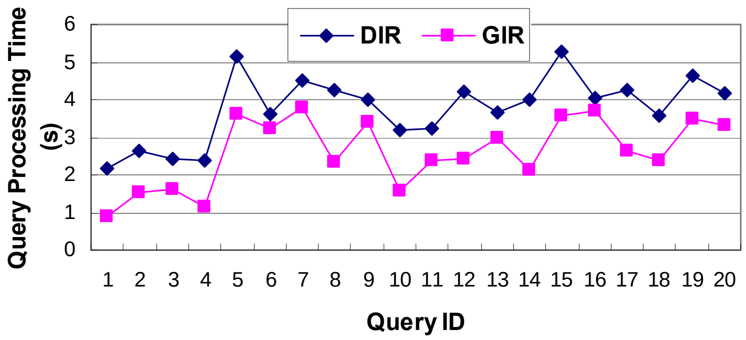


Figure 6.7 The query processing time for 20 queries

The difference of query processing time obtained in Figure 6.7 is 1.1 second, which seems not significant. One reason may be duo to the short length of queries. In the following experiment, we compare the average query processing time of queries with different length (number of terms). We generated three kinds of queries with lengths of 5, 10 and 15 terms, respectively. As shown in Figure 6.8, when the query length is 15, the difference of query processing time for DIR and GIR is 7.1 second. Although further experiments are needed to justify the real reason, we conjecture that the difference of performance will become more significant when the scale gets larger. This conjecture is further verified in the next experiment.

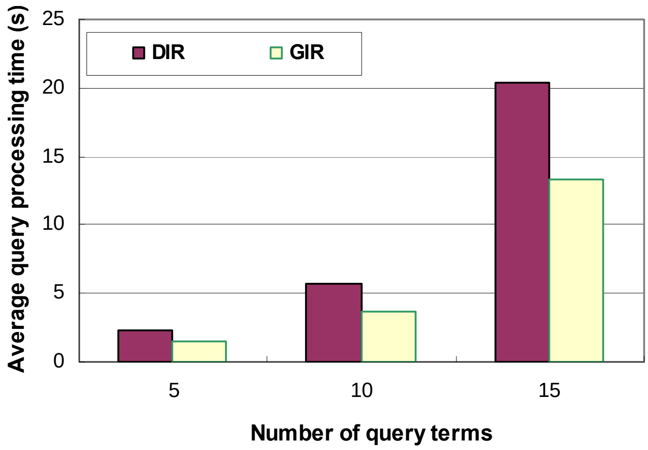


Figure 6.8 The average query processing time for different numbers of query terms

- ## Experiments on Scalability

This experiment aims to address the influence of the scale of LOR sizes. We generate four combinations of LORs with doubled sizes, as shown in Table 6.3. The number of query terms is 10. Figure 6.9 shows the average query processing time for collections of different numbers of documents. According to this figure, there is almost no scalability problem for GIR. However, scalability begins to become an issue when we further increase the size of document collection for DIR. In general, the proposed approach is more scalable. This is mainly because the global TI-tree decreases the searching time.

Table 6.3 Combinations of learning object repositories

Combination of Sites	Original No. of Documents	Double No. of Documents
LZ	300,000	600,000
LZ+HIT	900,000	1,800,000
LZ+HIT +PU	1,500,000	3,000,000
LZ+HIT +PU+THU	2,400,000	4,800,000

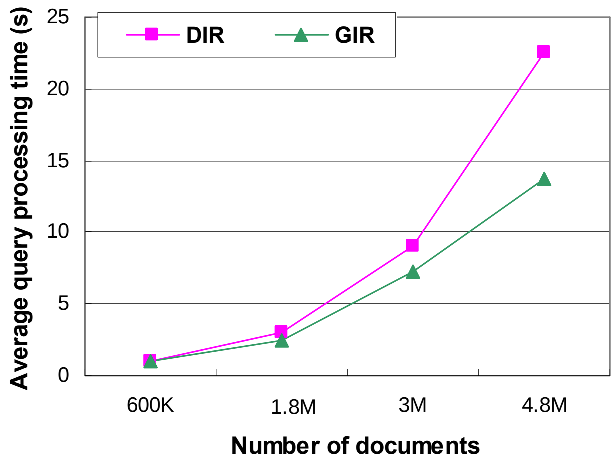


Figure 6.9 Scalability for increasing numbers of documents

In the experiment, 12 participants are invited to use our prototype system for retrieving documents. These users include 12 primary school teachers in Taichung and Nantou, in Taiwan. They are asked to answer two questions after they use this prototype. The first question is concerning the precision of the retrieved results, and the second question is regarding the perceived performance of transmission. The results are illustrated in Figure 6.10. In this figure, the score given by a user ranges from one to five. “One” means “Very Unsatisfactory” and “Five” means “Very Satisfactory”. The average of satisfaction for “Search Time” is 4.6, and that for “Precision” is 3.9. This experiment shows that our system could be efficient and helpful to users. After reading comments from these teachers, we find that they are satisfied with the response time of the prototype.

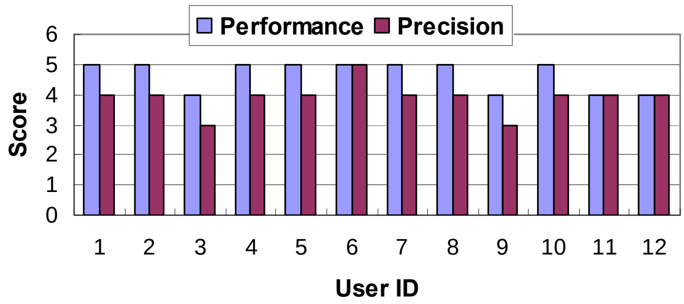


Figure 6.10 The running time for different numbers of classes

Chapter 7 Learning Content Retrieval on Peer-to-Peer (P2P) Networks

7.1 Retrieval on P2P Networks

With the flourishing development of information technologies, e-learning has become a promising learning paradigm. A number of e-learning researches aim to facilitate adaptive learning, which provides a customized environment according to students' requirements. To conduct adaptive instruction, teachers need to prepare customized TM (teaching material) for students with various learning styles, which is a heavy burden for teachers. TM sharing has been proposed to avoid redundant efforts on TM authoring. However, two limitations of current TM sharing platforms hinder the development of TM sharing. First, centralized management in these platforms can assure the quality of TM, but it can also discourage the passion for authoring. For example, not all submitted TMs can be published, resulting from collection policies and censorship. Second, the problems inherent in the client-server model, such as SPOF (single point of failure), service bottleneck, etc., can impede the expansion of the community.

With the concept of decentralization in mind, we propose a Blog-like P2P TM sharing environment. The Blog-like operation facilitates the publication of TM and the formation of communities, while the P2P nature can overcome the scalability problem. As shown in Figure 7.1, each peer in the P2P network represents a participating teacher, and an overlay is formed to represent the neighborhood relationship of participating peers. Also, a peer can join/leave the P2P network at any time, which characterizes the dynamic nature of P2P networks. To conduct TM sharing, a peer installs the tailor-made software, which supports Blog-like operations, such as publication, comment/reply, content organization, etc. In addition, the functions of TM search and download are incorporated.

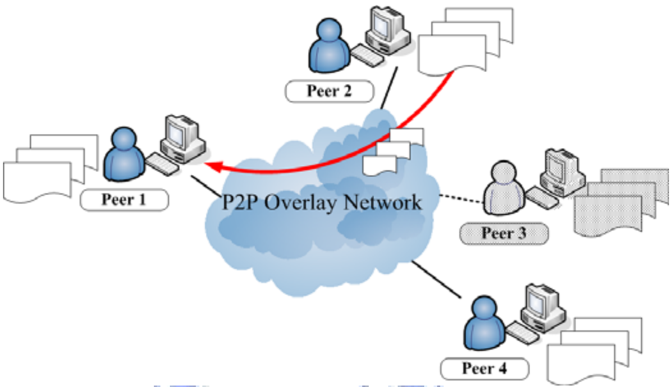


Figure 7.1 A Blog-like P2P TM sharing environment

Lacking a centralized index, search becomes a challenging issue in P2P networks. A number of researchers have been devoted to the study of P2P search. However, existing P2P search methods, which are mainly designed for file sharing applications, are not suitable for TM retrieval. The primary reason lies in fundamental differences between TM retrieval and file sharing in P2P networks, as shown in Table 7.1.

Table 7.1 Comparison of TM retrieval and file sharing in P2P networks

Difference	**TM retrieval**	**file sharing**
Target	TM	Movie, music, etc.
Alternative results	Many	Few
Searched by	Metadata, content	File names
Relevance to query	Ranking by similarity	Exact matching
Availability and Trust	Considered	Not considered

Furthermore, the attitude toward response time is different for TM retrieval and

P2P file search. Reducing response time is an important criterion for P2P search. However, this criterion is neglected by current P2P file sharing applications. For example, a typical P2P search process can be described as follows. The requesting peer sends a query message to some peers. Then, this peer waits for a period of time, called Due Time, to collect the response messages. In conventional P2P applications, due time is usually set a large value because of the all-or-nothing nature of exact matching. That is to say, users are willing and have to wait a long time for the desired file. For TM retrieval, however, we want to find a satisfiable TM, not a specific one. Therefore, we can pursue the goal of reducing response time.

Due time setting is one of important issues for TM retrieval in P2P networks. The primary reason is that a criterion is required to make decisions when a submitted query is not responded. Particularly, if the query is not replied in the expected time, which can be estimated by log mining, the user will be confused. In this work, the due time setting problem (DTS) is formulated as follows. Given a query, a positive integer k and a similarity function, determine the due time for each forwarded peer to retrieve k relevant teaching materials from a P2P network, where the peers might be unavailable or untrustworthy. The Goal is to maximize similarity and minimize due time. There are two main difficulties in the DTS problem. First, it is not totally a technical problem. TM retrieval involves subjective human factors. For example, the system can calculate the estimated due time according to availability information, and provide it to the user as a suggestion. It is fine when the requested peer answered in time. However, what should the user act if the requested peer does not respond in time? When unexpected situations happen, one flexible solution is to resort to users' wills. Second, too little information about peers for decision making is available. A mechanism of peer information acquisition is needed for human decision makers.

Our idea is to interactively set due time when an exception happens. Objectively, availability information can be for reference. Subjectively, it depends on users to make decisions in due time setting. For example, some user might want to wait longer for a quality peer, even with low availability. To realize this idea, a two-phased framework is proposed, which consists of a construction phase and a sharing phase. First, due time is set to a default value estimated by the system. When the query is due, the system asks the user whether to extend the due time or not. Other information is also provided for decision making.

The evaluation of the performance for the proposed approach consists of quantitative experiments and qualitative surveys. In the quantitative experiments, the interactive algorithm is compared with the two greedy algorithms in terms of similarity and waiting time. In the qualitative surveys, a satisfaction survey is conducted to understand the degree of user satisfaction for the design of interactive due time setting. The results imply that the interactive method is more flexible than the two greedy algorithms, and can be accepted by most of the users.

The contributions can be concluded as follows. First, we propose a Blog-like P2P TM sharing platform. Second, the Due Time Setting problem is formulated and solved. Particularly, a decentralized peer information management scheme is designed. Most importantly, an interactive due time setting algorithm is proposed.

7.2 Due-Time Setting Problem

In essence, information retrieval is a kind of human activities, which can be more efficient with the assistance of computer systems. The goal of traditional information retrieval problems is to improve such measures as precision, recall, etc. However, when considering TM retrieval in P2P networks, we should include more potential goals, such as response time. Therefore, we formulate the due time setting problem as a bi-objective optimization problem. Due to the dynamic nature inherent in P2P environments, we do not intend to formulate this problem in a strict mathematical form, which is used to be solved in an evolutional computing approach. Instead, we present a formulation which is suitable for interaction-based heuristic solutions. As explained in Section 1, the due time setting problem is to determine the due time for each peer to whom messages are forwarded. In this section, we introduce related definitions and then formulate this problem.

A *Query* is used by a user to specify the TMs s/he wants. Users can express their queries in two forms: keyword-based and metadata-based. A keyword-based query is a vector of keyword weights, which mean the concepts about the desired contents. A metadata-based query is a list of (Attribute, Value) pairs, which describe the properties of TMs.

In order to determine the degree of relevance of a query and a teaching material, the similarity function has to be defined. Conventional similarity functions, such as the cosine function, are not suitable for SCORM-compliant teaching materials which are characterized by textual content, metadata and structural information. Here, a similarity measure *Sim* between a query *Q* and a teaching material *TM* is proposed by combining a keyword-based similarity and a metadata-based similarity. The keyword similarity $Sim_{Keyword}$ adopts a cosine function to measure the text similarity between a query and a TM. The metadata similarity $Sim_{Metadata}$ is defined to be the number of matched attributes divided by the number of all attributes. Therefore, the range of these two similarity terms, $Sim_{Keyword}$ and $Sim_{Metadata}$, are both in [0, 1]. The similarity measure *Sim* is defined in (10).

$$Sim(Q,TM) = \alpha \times Sim_{Keyword}(Q,TM) + (1-\alpha) \times Sim_{Metadata}(Q,TM) \quad (7\text{-}1)$$

where the factor α, 0 ≤ α ≤ 1, is used to control the relative weighting of keyword similarity and metadata similarity. The setting of the α value is discussed in Section 7.7.

The P2P environment proposed in this work is an unstructured P2P network, where there is no centralized index structure. We model this P2P network as a graph. A node represents a peer who publishes TMs in local storage. An edge means a friendship relation between two peers. The network formed by the friendship relation is also called a P2P overlay network. To characterize the dynamic nature of the P2P network, where a peer can join/leave the network at any time, two important concepts are defined. *Availability* means the probability that a peer is on-line at some time instance. *Trust* is the probability that the information published by a peer is correct.

When a peer submits a query, it also assigns a due time, when the requesting peer will begin to merge results. After due time, responses from requested peers will not be accepted. The *Due Time Setting* (*DTS*) problem is described as follows. Given a query *Q*, a positive integer *k* and a similarity function *Sim*, determine the due time T_{Due} for retrieving *k* relevant teaching materials from a P2P network, where each peer has a repository, and the peers might be unavailable or untrustworthy. The goal is to optimize the following two objective functions:

$$\text{Max} \quad \frac{1}{k}\sum_{i=1}^{k} Sim(Q, TM_i) \quad (7\text{-}2)$$

$$\text{Min} \quad T_{\text{Due}} \quad (7\text{-}3)$$

where TM_i is the *i-th* TM retrieved from the P2P network.

7.3 Peer Node Architecture

As mentioned in Section 7.1, the main difficulty of the DTS problem lies in that it is not totally a technical problem. TM retrieval involves subjective human factors, and flexible setting is desirable to respond to unexpected situations in P2P networks. In addition, too little information about peers for decision making is available. Therefore, our idea is that users can interactively adjust the due time. The architecture and algorithms based on this idea are presented in this section.

To realize this idea, a two-phased architecture for each peer node is proposed, as shown in Figure 7.2. In Construction Phase, an overlay network is formed, and the peer information is collected and managed. In Sharing Phase, the peer node can publish and retrieve TMs.

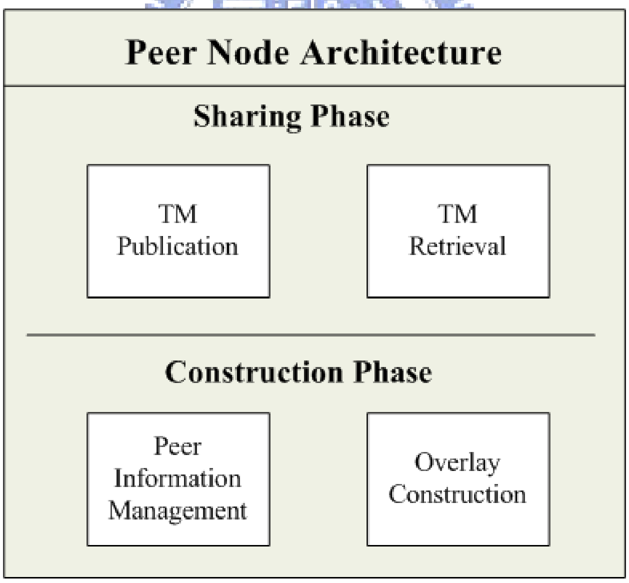


Figure 7.2 Peer node architecture

The four modules are described as follows.

- TM Publication Module. This module enables this peer to publish TMs in a Blog-like manner. The published TMs are posted on the "Blog", and are stored in the local repository. Other peers can browse and download the published TMs.

In addition, common Blog operations are supported by this module, such as comments.

- TM Retrieval Module. This module enables the peer to retrieve desired TMs from the P2P network by submitting a query.
- Peer Information Management Module. The purpose of this module is to support decision making. This module manages the peer's information, including private information and public information. The former is used by the peer, and the latter is for public access.
- Overlay Construction Module. This module enables the peer to develop a friendship overlay in the P2P network. A number of methods have been proposed to build P2P overlay, among which the method proposed by [70] is an effective method. In this work, we use existing methods to implement the overlay construction module.

Lacking global information, each peer maintains three data structures in this architecture. 1) A Friend table. This table is built by the overlay construction module. Each row represents a friend peer. The table size is limited by the capacity of system memories. The fields include Trust, Availability, Collection Summary (C_ Summary) and Collection Size (C_Size).

- Trust. A peer evaluates this value for each friend. The trust value is an accumulative score about the trustworthiness of the friend. The update of Trust is described in the next section.
- Availability. A peer maintains this value for each friend. The peer acquires the availability value which is published by the friend. Therefore, the accuracy of the value depends on the friend's trustworthiness. A trustworthy peer will probably provide correct availability.

- Collection Summary. Each peer publishes the summary information of its repository, in order that others can access this summary information and estimate the content of the repository. This summary information is also represented by a VSM-based vector. For example, assume that the vocabulary includes three keywords: K_1, K_2 and K_3. A peer can publish its collection summary by calculating the three weights, w_1, w_2 and w_3, from TMs in its repository. Then, the summary vector is obtained as $< w_1, w_2, w_3>$.
- Collection Size. This value is also provided by each peer for others to access. For example, if a peer has 20 TMs in its repository, this peer can publish its Collection Size as 20.

An example is shown in Table 7.2. 1) This peer maintains three friends: peer 2, peer 3 and peer 4. 2) Self-information array. This array contains such information as Availability, C_Summary and C_Size, which is for public access. 3) A Local index for its TM collection. With this index, the search in local repository can be sped up. The index is built by the scheme in [92].

Table 7.2 An example of a Friend Table

Information / Peer	Trust	Availability	C_Summary	C_Size
Peer 2	0.9	0.2	<0.7, 0.6, 0.4>	20
Peer 3	0.6	0.8	<0.5, 0.8, 0.6>	35
Peer 4	0.5	0.6	<0.3, 0.7, 0.9>	16

7.4 Peer Information Management

Peer information management is an important task in Construction Phase. To facilitate decision making for TM retrieval in the P2P network, each peer maintains two categories of information:

- Public information. This kind of information is about the peer itself, and is generated by itself for other peers to access. Availability, collection summary and collection size can be categorized to this class.
- Private information. Each peer privately collects this type of information from other peers, such as trust information. This information can be obtained by evaluating previous searching transactions.

Trust information represents the degree to which the peer has trust in the information provided by other peers. Trust information can be used to update friend tables and evaluate other peers' information. Policies for initializing trust information depend on peers. The optimistic policy can have 1 as the initial value. The pessimistic policy sets the initial value to 0. With a neutral policy, the initial value is 0.5. Updating of trust information occurs after each searching transaction. If the retrieved TM from peer *j* has a higher similarity value than the C_Summary claimed by peer *j*, the Trust for peer *j* will be increased. Otherwise, the trust score will be decreased. A normalization function, *Normal*, will be applied to the calculated score to generate a value ranging from 0 to 1. The definition of *Normal* is as follows. It returns:

- 1, if its input parameter is larger than 1;
- 0, if its input parameter is smaller than 0;
- its input parameter, otherwise.

We denote the original trust information for peer *j* by $Trust_j$, and the updated trust by $Trust_j'$. Also, *Sim* is the similarity function defined in (13). Then the update formula is listed as follows.

$$Trust_j' = Normal(Trust_j + Sim(\mathrm{Q}, \mathrm{V}_{\mathrm{Retrieved}}) - Sim(\mathrm{Q}, \mathrm{V}_{\mathrm{C\_Summary}})) \qquad (7\text{-}4)$$

where

- Q is the query;
- $V_{Retrieved}$ is the content vector for the retrieved TM;
- $V_{C\_Summary}$ is the collection summary information published by peer j.

Availability information, generated by a peer itself, means the probability that a peer is on-line. Methods for generating availability information depend on peers. Simple methods can estimate online probability by statistics on the past 24 hours. Advanced mechanism, such as Markov Process, can also be used to generate predictive online probability. Update frequencies also depend on peers.

Collection Size is the number of TMs in the peer's collection. Collection Summary is the summary information of the collection. The degree of detail depends on peers' choice. An average vector, level-wise vectors and Categorized vectors provide different details.

7.5 TM Retrieval

In contrast to traditional P2P search methods, such as flooding, we adopt a query forwarding mechanism based on a built overlay to improve the search performance. The algorithm for TM retrieval consists of the following main steps: local search, query forwarding and result merging. Query forwarding includes peer selection and due time setting. This section focuses on the former, and the latter is presented in the next section.

The query forwarding is based on the P2P overlay. That is, a peer forwards the query to its selected friends. The purpose is to select peers which possibly have more similar TM. The criterion for peer selection is as follows.

$$Trust_{\mathrm{j}} \times Sim(\mathrm{Q}, \mathrm{V}_{\mathrm{j}}) \tag{7-5}$$

where

- $Trust_j$ is trust information of peer j;
- V_j is the vector of collection summary for peer j.

In Result Merging, the requesting peer collects the responses from its friends. When the request is satisfied, the requesting peer can choose to finish the merge and omit the friends who are still in processing. Condition for stopping searching beforehand can be stated as follows.

- The number of retrieved TMs >= k, and
- The estimated similarity of the remaining peer < the smallest similarity of retrieved TMs

When a peer receives a searching request, it executes this algorithm, Peer_Retrieval. Step 2.2 calls the subroutine *Sub_DTS* to interactively set the due time, which is detailed in the next section.

Algorithm 7.1 Peer_Retrieval (ALG_PR)

Symbols Definition:

***Q*:** the query submitted by a user

***Num_TM*:** the number of TMs to be retrieved

***TM_set_size*:** the size of the returned set of TMs

***TM_set_summary*:** summary of the returned set of TMs

Input: *Q*, *Num_TM*

Output: *TM_set_size, TM_set_summary*

Step 1: Local searching

Step 2: Query forwarding

Step 2.1: Select peers

Step 2.2: call Sub_DTS // for due time setting

Step 2.3: Forward the query

Step 3: Result merging

Step 4: Trust information updating

Step 5: Return

7.6 The Heuristic Method for Due Time Setting

This section describe the subroutine called by Step 2.2 of ALG_PR for due time setting in detail. This method is based on an IRT (Item Response Theory)-like function, which characterizes the relationship between due time and availability. Let $\mathrm{Prob}_j(T_{due})$ denote the probability that a requester can access a peer j given the setting of due time, T_{due}.

$$Prob_j(T_{due}) = \frac{1}{1+e^{(T_{due}-T_{avail})}} \qquad (7\text{-}6)$$

where T_{due} is the time instance before which the requester will wait; T_{avail} is the estimated time instance before which the peer j will not be available. For example, assume that T_{avail} is 8:30 p.m. If T_{due} is set to 8:30 p.m., the probability of successful access to peer j is 0.5. When $T_{due} > T_{avail}$, the probability will be larger than 0.5. When $T_{due} < T_{avail}$, the probability will be smaller than 0.5.

For initialization, system default values are calculated according to the formula (15). During the search process, users can interactively extend the default due times. If a quality peer is not on-line, the system will ask the user whether to extend the due time or not. If an on-line peer exceeds the due time, the system will also ask the user whether to extend the due time or not. For example, “Peer 2 is not on line. It has quality TMs you want, and might get on line soon. Would you extend the due time?” The procedure is listed as follows.

Algorithm 7.2 Subroutine: Interactive_Due_Time_Setting (Sub_DTS)

Subroutine: Interactive_Due_Time_Setting (*Sub_DTS*)

Symbols Definition:

T_{due}: the time instance before which the requester will wait

T_{avail}: the estimated time instance before which the peer *j* will not be available

$Prob_j(T_{due})$: the probability that a requester can access a peer *j* given T_{due}

p: the desired probability that a requester can access a peer *j*

Input: p, T_{avail}

Step 1: Set T_{due} according to the formula of $Prob_j(T_{due})$ in (6), given p.

Step 2: Wait until time is due.

Step 3: For all peers who have not responded

Step 3.1: Show the unresponsive peer’s information, and ask the request whether to wait or not.

Step 3.2: Get the requester’s answer.

Step 4: If the requester will not wait, return.

Step 5: Go to Step 1.

7.7 Evaluation on P2P Networks

In this section, the implementation and evaluation design are described. Then, experimental results are presented and discussed.

- **Test Collections and P2P Configurations**

We have collected two TM sets for the experiments, SLN and LFS. SLN is a collection of TM transferred from the Six Learning Nets (http://learning.edu.tw/sixnet/) built by the Ministry of Education, Taiwan. LFS is a collection of TM transferred from the Learning Fueling Station (http://content1.edu.tw/) built also by the Ministry of Education, Taiwan. Characteristics of the two test collections are presented in Table 7.3.

Table 7.3 Characteristics of the three test collections

Test Collection	**No. of TMs**	**Average Length of a TM (word)**	**Subject**	**Metadata**	**Description**
SLN	1200	451.3	Mathematics	Yes	For elementary schools
LFS	1200	1050.2	Mathematics	Yes	For junior high schools

We have simulated four P2P environments with different trust and availability. In P2P configuration with low trust, the probability that a peer honestly publishes its information is 0.25, while the probability is 0.75 in a high-trust P2P configuration. Similarly, In P2P configuration with low availability, the probability that a peer is on-line is 0.25, while the probability is 0.75 in a high- availability P2P configuration. Characteristics of the four configurations are presented in Table 7.4.

Table 7.4 Characteristics of the four P2P configurations

P2P Configuration	**Trust**	**Availability**
1	High	High
2	Low	High
3	High	Low
4	Low	Low

(low = 0.25, high = 0.75)

We have implemented a prototype to conduct the experiments. This prototype is developed based on open-source software, myJXTA (https://jxta.dev.java.net/), and the Java language. We have also setup a small-scale P2P community, which consists of twelve elementary-school teachers.

- **Quantitative analysis**

We have designed another two greedy DTS methods based on precision and response time respectively, for the purpose of comparison with the interaction-based method. The precision-based method always waits for all friends to respond the query, while the time-based method waits for a pre-defined time period. The measure of precision is the same as that defined in conventional information retrieval literatures.

$$Precision = {N\_Relevant}\big/{N\_Retrieved} \tag{7-7}$$

where

- *N_Relevant* is the number of relevant TMs in the retrieved TMs;
- *N_Retrieved* is the number of retrieved TMs.

The two figures in Figure 7.3 show the results with respect to SLN and LFS respectively. First, we observe that the precision-based method performs well in all configurations. This is because it broadcasts to all peers, and waits a long time for the results. This will incur a large cost. Second, the time-based method degrades significantly when the P2P configuration becomes low-trustworthy or low-available. The main reason may be that it statically set a small due time, and some relevant results are too late to be received. The interactive method is almost as good as the performance-based method in terms of precision.

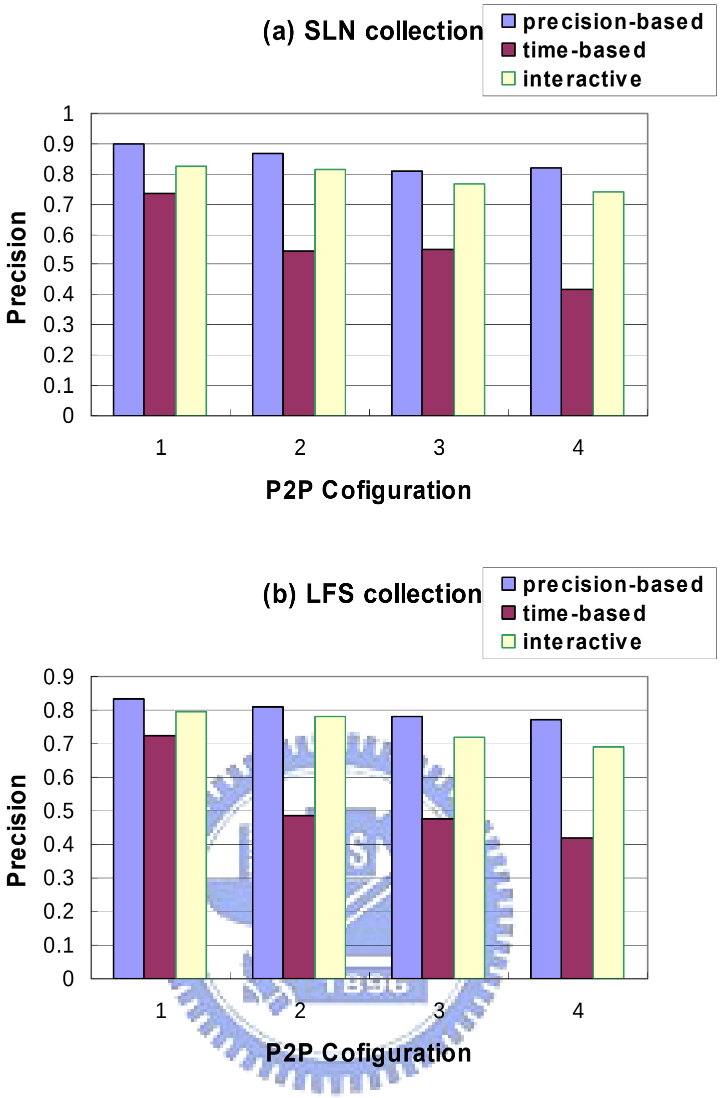


Figure 7.3 Comparison of Precision for the three DTS methods: (a) SLN; (b) LFS.

The two figures in Figure 7.4 show the results in terms of response time. First, we observe that the time-based method performs well in all configurations. This is because it sets a fixed small due time. However, this will incur poor precision as mentioned above. Second, the precision-based method degrades significantly when the P2P configuration becomes low-trustworthy or low-available. The main reason may be that it always set a large due time. When the environment becomes dynamic, the response time will be long. The interactive method is almost as good as the performance-based method in terms of response time.

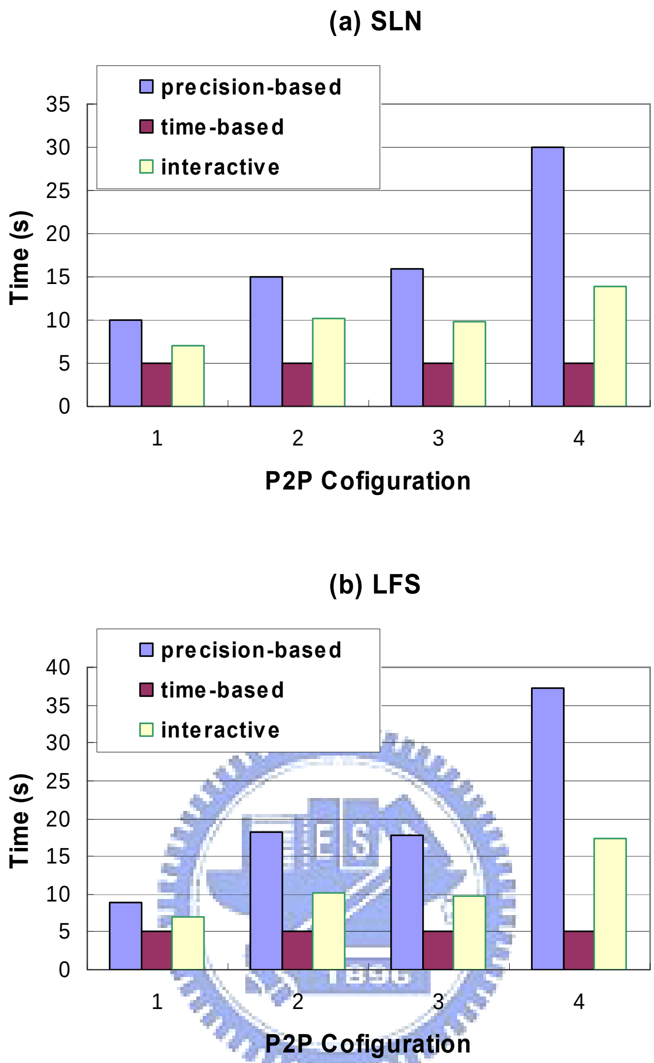


Figure 7.4 Comparison of response time for the three DTS methods: (a) SLN; (b) LFS.

The two experiments show that the interactive algorithm is both effective in terms of precision and response time. Although it is not the optimal method, the interactive method can adapt to low-availability and low-trust P2P networks.

- **Qualitative analysis**

One month after the users began to use this application, a survey was conducted with respect to the twelve teachers. Each teacher was asked four categories of questions to obtain their comments on the prototype system. Each category has five questions, and a five-point Likert scale with anchors ranging from strongly disagree

(1) to strongly agree (5) is used for this survey. The mean value and standard deviation (SD) is calculated for each category, as shown in Table 7.5.

For Category 1 questions, the deviation of user satisfaction is slightly larger than other categories. The reason may be that the participants are not all familiar with the concept of Blog. Some participants comment that they are not used to publish their articles to others. However, some participants appreciate this idea and like to frequently update and publish their TMs.

The results of Category 2 and 3 show that participants are willing to publish their private information for efficient retrieval. However, the published information is not highly satisfiable. This implies that we can reconsider to include more useful information as public information to improve the user satisfaction, such as peers' professional information.

In summary, the interactive due time setting is user-friendly and satisfactory, according to the results of Category 4.

Table 7.5 The result of the survey

Category No.	Questions	Mean	SD
1	Satisfaction of the blog-like TM sharing platform	3.46	1.23
2	Satisfaction of peer information provided by others	3.69	0.91
3	Willingness to provide self information	4.62	1.18
4	Satisfaction of interactive due time setting	4.33	0.55

- **Discussion**

This study adopts the similarity measure which combines keyword similarity and metadata similarity through a controlling factor, α. In this section, we investigate how the α value is determined to improve the precision of TM retrieval. Twelve queries are submitted to SLN and LFS collections, respectively. Also, the α value is set to be 0, 0.25, 0.5, 0.75, 1, respectively. For each different setting, the experiment is repeated five times, and the average precision is obtained. To avoid the dynamic effect of P2P networks, all participating peer are always on-line.

Figure 7.5 shows the precision value for different α values, which range from 0 to 1. The observations can be summarized as follows. First, we obtain the best precision when $\alpha = 0.75$, among the five α values. This implies that the combination of keyword similarity and metadata similarity is useful. Second, the precision on $\alpha = 0$ (metadata similarity only) is slightly smaller than that on $\alpha = 1$ (keyword similarity only). This probably results from the characteristics of the two TM collections. The metadata do not include detailed subject information. Therefore, using merely metadata can not exactly retrieve the desired TM.

Based on the results, we set $\alpha = 0.75$ for experiments in this paper.

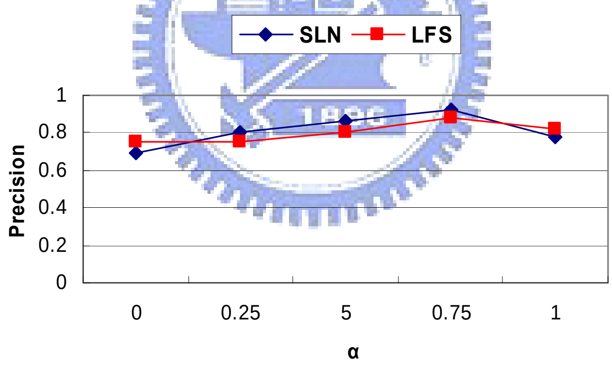


Figure 7.5 The effect of the α value on precision

Chapter 8 Application: Wiki-based Teaching Material Design

With the trend of individualized and adaptive learning, there will be a great demand to various teaching-materials. A typical approach to content design is ADDIE [97], which consists of five stages: Analysis, Design, Develop, Implement, and Evaluate. The primary disadvantage is its time-consuming development process. In addition, it requires expensive human resources. Furthermore, redundant efforts could happen when different sites develop teaching materials for the same course units simultaneously. To solve the problem, a new method is needed for teachers to rapidly develop their own course materials.

Our idea is to design teaching materials by a rapid prototyping approach based on automatic draft generation and Wiki-based revision. Rapid prototyping is the process of quickly building and evaluating a series of prototypes of a system, which has been widely applied to manufacturing, software engineering [98], etc. First, a draft is automatically generated by combining relevant teaching materials in e-Learning grid. However, the main challenges result from verification of user intention and finding useful teaching materials from existing ones. To address this issue, we plan to enhance searching performance by using expertise acquired by a powerful knowledge acquisition tool to speed up the development process. Next, we adopt a Wiki-based authoring environment to revise the automatically generated draft. Wiki is an accessible markup language for people to edit a site together. Wikipedia is the most successful Wiki-based project. Our method is to utilize the collaborative intelligence and labor to accelerate the revision process. The primary difficulties are the storage requirements and overheads of maintaining historical revisions. This issue is alleviated in grid computing environments because of its abundant resources of storage and computation.

Based on the aforementioned ideas, we propose an approach named WARP (Wiki-based Authoring by Rapid Prototyping), which is composed of five phases: 1) requirement verification, 2) query expansion, 3) teaching-material retrieval, 4) draft generation and 5) Wiki-based revision. The goal is to reduce the development time of teaching materials. Firstly, the system attempts to clarify users' intention by interactive ways, such as asking questions, requesting more query terms, etc. In the second phase, users' queries are expanded by using domain expertise to retrieve more relevant documents. Next, the system searches for existing teaching-materials related to the expanded query in the grid. Then, the retrieved documents are combined into a draft automatically in the fourth phase. Finally, the draft is placed in a Wiki-based authoring environment for collaborative revision.

The advantages of WARP are twofold: time-saving and low-cost, which result from effective sharing and reusing of resources. Meanwhile, our primary contribution is the idea of a rapid prototyping approach to teaching-material design for e-Learning grids. In addition, we deployed a prototype system in a grid environment, implementing each phase of the WARP approach. Twenty four randomly selected teachers from elementary schools participated in an experiment based on a two-group t-test design. Experimental results show that teachers in the experiment group can generate high-quality teaching materials more rapidly than those in the control group.

8.1 Teaching Material Design Problem

Typically, grid infrastructure is built with a suite of middleware. Common middleware platforms, such as Globus Toolkits and Condor [99], are based on a master-slave paradigm. Hence, we represent the grid by a master-slave model. Also, real-time information is included to model the dynamic grid. A grid is a star graph $G = <N_G, E_G>$ that consists of a finite set of node N_G, and a finite set of edges E_G. N_G represents the set of sites in the grid. One node $P_0 \in N_G$ is specified as the master node, and other nodes are slave nodes. Each edge in E_G connects the master node and a slave node.

A query Q is modeled as a set of keywords. The feature vector of a query Q is denoted by v_Q,

$$v_Q = < q_1, q_2, \ldots, q_{|V|} > \tag{8-1}$$

where V means the set of vocabulary and $|V|$ is its size. The term-weighting q_i is 1 if the i-th keyword in the vocabulary, V, is a term in the query. Otherwise, q_i is 0.

We will now define the notion of similarity between a query and a content package, which means the relevance of the content package to the query. Let Q be a query with feature vector v_Q, and CP be a content package with feature vector v_{CP}. The Similarity $sim(Q, CP)$ is defined by:

$$sim(Q, CP) = v_Q \cdot v_{CP} \tag{8-2}$$

where the operation is inner product of vectors.

The editor, who submits the query to develop a teaching material, is usually a teacher who is not necessarily an expert in courseware design. Domain taxonomy and a thesaurus are assumed to be available for the material development process. Also, the designer can reuse any existing teaching materials in LORs of the e-Learning grid.

We assume that the course ontology, built by educational experts, is available, as shown in Figure 8.1. The subject matter is mathematics for nine-year coherence curriculum at low-grade elementary-school level, according to Ministry of Education in Taiwan. The course ontology is modeled as a rooted tree $O = <N_O, E_O, root>$ that consists of a finite set of node N_O, a finite set of edges E_O, and a root node in N_O. N_O represents the set of nodes in the tree. Each node in N_O represents a concept in this ontology and is associated with a set of keywords, which describe the concept. For example, the set of keywords of the node "Arithmetic Operations" is {"addition", "subtraction", "multiplication", "division"}. An edge in E_O connects a node and its child node, which expresses the hierarchical relation of the two nodes. For instance, the edge connecting "Geometry" and "Shapes" means the former is a more general concept than the latter. Finally, the root node in this example is the "Mathematics" node.

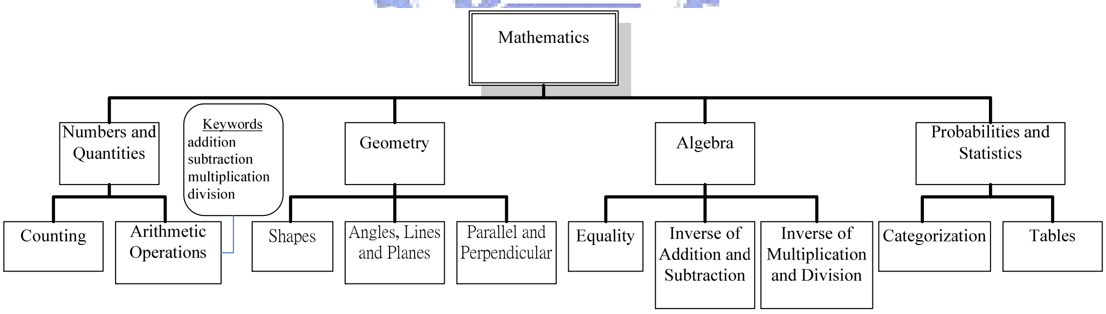


Figure 8.1 The ontology of mathematics at elementary-school level

Based on the definitions above, the Teaching-Material Designing Problem (TMDP) can be described as follows. For a query given by a teaching-material editor, design a teaching-material, where the designer can interactively consult the editor to elicit the meaning of the query; the existing materials in m LORs can be reused; also, a course ontology is available. The objective is to minimize the total development time.

8.2 Wiki-based Rapid Prototyping

The Search phase is carried out by the search engine component, which receives queries from users, processes the queries, and presents results to the users. After users specify the desired documents from the returned results, the search engine accesses the site where the documents are stored and retrieves these documents for the users.

When a user submits a query which contains terms don't belong to the Vocabulary, the search engine will suggest some synonymous terms in the Vocabulary. The suggestion is based on a synonym dictionary.

The purpose of this phase is to minimize the time of query processing and content transmission when retrieving SCORM-compliant documents in a grid. To speed up the searching process, our idea is to use a centralized index, which is generated by reorganizing the existing documents based on a bottom-up approach, because this approach is suitable for the master-slave grid model and can effectively collect the information of existing documents from all sites in the grid. Furthermore, the indexing structure stores metadata and structural information, which increases the efficiency and precision of searching. To speed up the transmission process, the other idea is to present the ranked results with estimated transmission time, which is derived from grid monitoring tools. In this way, the document which has high ranking score but has a long estimated transmission times (maybe due to a low-bandwidth link) can be avoided by users.

First, the problem of retrieving SCORM-compliant documents on e-Learning grids is addressed. To efficiently manage documents in the grid, an indexing structure named Indexing Trees (I-trees) have been designed. A I-tree is based on an existing taxonomic schema and has two novel features: 1) reorganizing documents according to the Classification metadata such that queries by classes can be processed efficiently and 2) representing each document by a term-weighting vector, where the term-weighting includes structural information. In this way, an appropriate weighting scheme can be used to utilize the structural information in the SCORM-compliant documents. In addition, the cost of constructing, merging and maintaining I-trees is not expensive, but the benefits are significant. The overall process of this approach is composed of a Construction phase and a Search phase. In the former, a local I-tree is built from each Learning Object Repository. Then, all local I-trees are merged into a global I-tree. In the latter, a Grid Portal processes queries and presents results to users. After the user specifies the desired documents, the Portal retrieves this document from the target site for the user.

This phase is mainly composed of three steps. In the following paragraphs, these steps are described in detail.

- **Step 1: requirement verification**

This phase aims to clarify users' information need specified by query terms. Because the initial queries submitted by novice editors are usually vague and ambiguous, documents found by the searching engine might not answer to their expectations. Lee and Liu modeled query intention with a goal-driven approach [100]. In spite of its powerful functionality, it is not suitable for our rapid prototyping approach in terms of runtime overhead. Therefore, a lightweight method is designed to clarify users' intention. To begin with, a single-concept query is defined as follows:

Definition 8.1. Single-concept queries.

A query is said single-concept with respect to a ontology if its terms appear in the set of keywords of only one node in the ontology.

■

The following example is presented to illustrate this definition.

Example 8.1. Single-concept queries

Let a query Q be {"addition", "multiplication"}. As shown in Figure 8.2, the keyword set of the node "Arithmetic Operations" is {"addition", "subtraction", "multiplication", "division"}. Q is a subset of the keyword set, and its terms do not appear in other ontology nodes. Therefore, Q is a single-concept query with respect to the ontology.

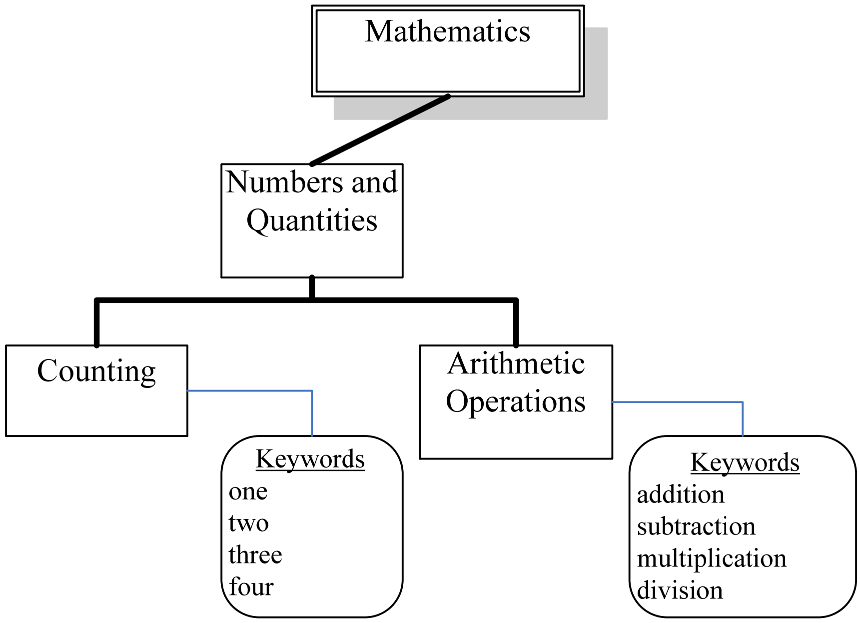


Figure 8.2 The partial ontology of mathematics at primary school level

This method uses the ontology of courses to identify users' intentions. By means of finding the ontology nodes whose keywords include the query terms, the system can validate the search scope of the query. Users' requirements are verified through dialogues between users and computers. The algorithm is presented as follows.

Algorithm 8.1 Requirement Verification Algorithm (QVAlg)

Symbols Definition:

***Q*:** a set of keywords; the query submitted by a user

O*:** a tree <N***, ***E***>, where ***N*** is the set of nodes and ***E*** is the set of edges;

representing an ontology

L**:** the depth of the ontology tree ***O***

Root**:** the root node of the ontology tree ***O***

S**:** an ontology node in ***N***, representing the scope of the query, returned by the algorithm

Input: *Q*, ***O***

Output: *S*

Step 1: *count* := 0

Step 2: for each node ***I*** in the ontology ***O***

If (***Q*** and ***I*** have some keywords in common) **Then**

S* := *I

count := *count* + 1

ENDIf

Step 3: If (*count* $\neq$ 1) **Then**

Ptr **:= Root**

For *i* **:= 1 to *L***

Ask the user to choose a node from the children of *Ptr*

let *Ptr* point to the node chosen by the user

S := *Ptr*

ENDIf

Step 4: return *S*

The purpose of Step 2 is to check whether the query is a single-concept one with respect to the ontology. When a query involving multiple concepts is detected or query terms do not belong to the ontology keyword set, the interactive interview process is activated, as shown in Step 3. In the past, interactive interviewing is an effective way to acquisition what users think in knowledge engineering domain. The system uses a top-down traversal on the ontology to ask questions. Requirements are verified through dialogues between users and computers. For example, the following dialog, trigged by Step 3, can help the system clarify the subject matters of users' intentions.

Example 8.2. Interactive interview

Q: "Please select a topic: "

1) Number 2) Geometry 3) Algebra 4) Statistics

A: 1

Q: "Please select a sub-topic: "

1) Counting 2) Addition 3) Subtraction 4) Multiplication 5) Division

A: 4

According to users' answers, the system can confirm that the user want to find documents related to "Multiplication" in the "Number" topic. Therefore, the system returns a modified query containing predefined keywords in the topic.

- **Step 2: query expansion**

Searching strategies used by experts and novices are different. If the domain expertise of searching experts can be acquired, the system can assist novices to search like an expert. This thesis is focused on the strategies of query expansion. The main reason is that the query expansion technique can be well integrated with existing information retrieval technologies, which are primarily based on the Vector-Space Model.

To acquire searching heuristics of experts, the Repertory-Grid method [101] is used, which has been widely applied to knowledge acquisition. After the analysis of the Repertory-Grid, rules representing the knowledge of searching experts can be generated. The purpose of these rules is to choose an appropriate strategy according to the input values of attributes. The inference process can be illustrated in Figure 8.3.

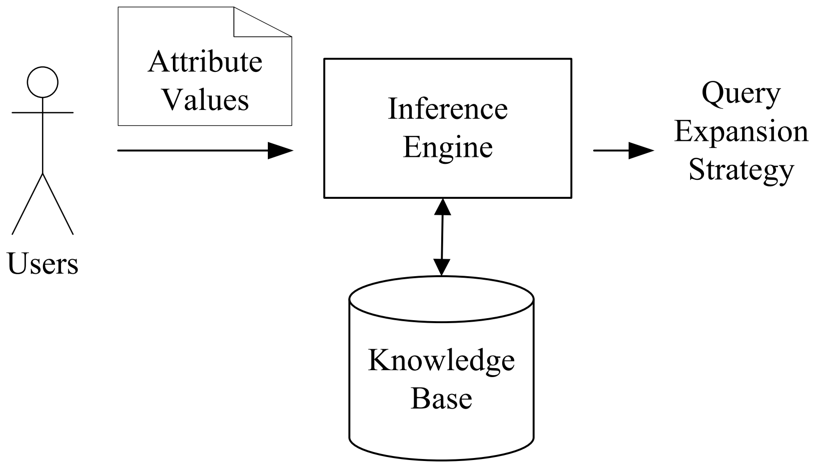


Figure 8.3 The architecture of inference for searching strategies

In this work, four strategies of query expansion are adopted. These operations are guided by the given ontology. When the information need is focused on some node in the ontology, the four operations can be used to reformulate the query, described as follows.

- Generalization: append keywords of the parent node in the ontology.
- Specialization: append keywords of the children nodes in the ontology.
- Expansion: append other keywords of the same nodes in the ontology.
- Shrink: remove terms of the query.

The choice of a suitable strategy depends on the values of several attributes, which could include:

- The number of query terms
- The number of the previous searching results
- The level of the targeted students

The following example shows a rule representing one of the expert heuristics.

Example 8.3. Knowledge of query expansion

IF (Number_Query_Term < 2) THEN Expansion

This rule means that the "Expansion" strategy is triggered when the number of query terms is less than two. Therefore, The system selects keywords from the current node in the ontology to expand the original query. The number of selected terms depends on the system configuration.

The following example illustrates the difference between a query and its expanded version.

Example 8.4. Query Expansion

Q = {"multiplication"}

The original query contains only one term. Assume that the current node is "Arithmetic Operations". Based on the rule in Example 3, the "Expansion" action is triggered. Then, the system selects another two keywords in the current ontology-node, addition and subtraction. After expansion, the query contains three terms.

Q' = {"Multiplication", "addition", "subtraction"}

- **Step 3: teaching-material retrieval**

In this phase, the expanded query is used to search for relevant teaching materials in the grid. Learning object repositories are located at different sites, which are connected with wide area networks. Our method is to reorganize the documents in each learning object repositories into a local index. Next, all local indexes are merged into a global index. The global index is then utilized to find the relevant documents. The primary advantage of building a global index is its suitability for centralized management and implementation on e-Learning grids, thus increasing the searching performance.

The searching algorithm is shown below.

Algorithm 8.2 Teaching-Material Searching Algorithm (TMSAlg)

Symbols Definition:

$\boldsymbol{V_Q}$**:** the feature vector of the query ***Q*** submitted by a user

***k*:** the number of documents to be returned

$\boldsymbol{V_D}$**:** the feature vector of a document ***D*** or teaching material ***D***

***Sim*:** a similarity function, cosine function

***I*:** the global index of teaching materials

R*:** the ***k teaching materials found by the algorithm

Input: $\boldsymbol{V_Q}$, ***k***

Output: ***R***

Step 1: For each document ***D*** indexed by ***I***

Compute $\boldsymbol{Sim}(\boldsymbol{V_Q}, \boldsymbol{V_D})$

Step 2: Rank all documents by the similarity values

Step 3: return the first ***k*** documents as ***R***

Example 8.5. Teaching-Material Searching

Let the feature vector of the query be <1, 2, 1>. Assume that the repositories contain 5 documents, and their feature vectors are :

Document 1: <1, 2, 1>

Document 2: <3, 2, 1>

Document 3: <2, 2, 1>

Document 4: <2, 2, 3>

Document 5: <3, 2, 3>

In this example, the similarity function is vector inner product. Then, the similarity values are 6, 8, 7, 9 and 10, respectively. Consequently, the most relevant document is Document 5, which has the highest similarity value.

8.3 An Illustrative Example

An example is presented to illustrate how teachers of an elementary school use the WARP method to collaboratively design a teaching material for the "Area" unit in the third-grade Mathematics course. The ontology of the "Shape" unit is shown in Figure 8.4. The keyword "Area" is associated with the "Shape" node. The overall process is summarized as follows.

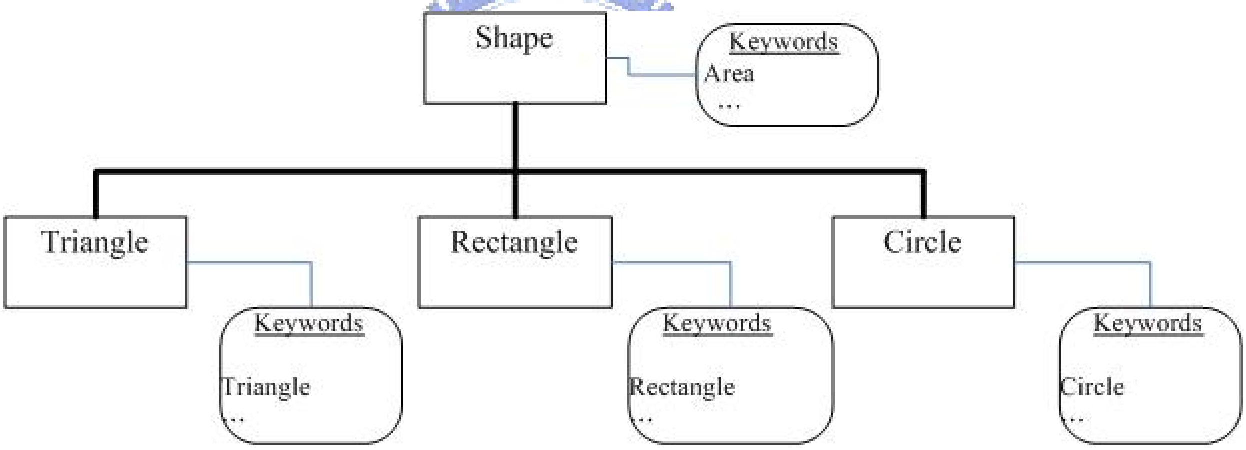


Figure 8.4 The ontology of the "Shape" unit

- Phase 1: requirement verification

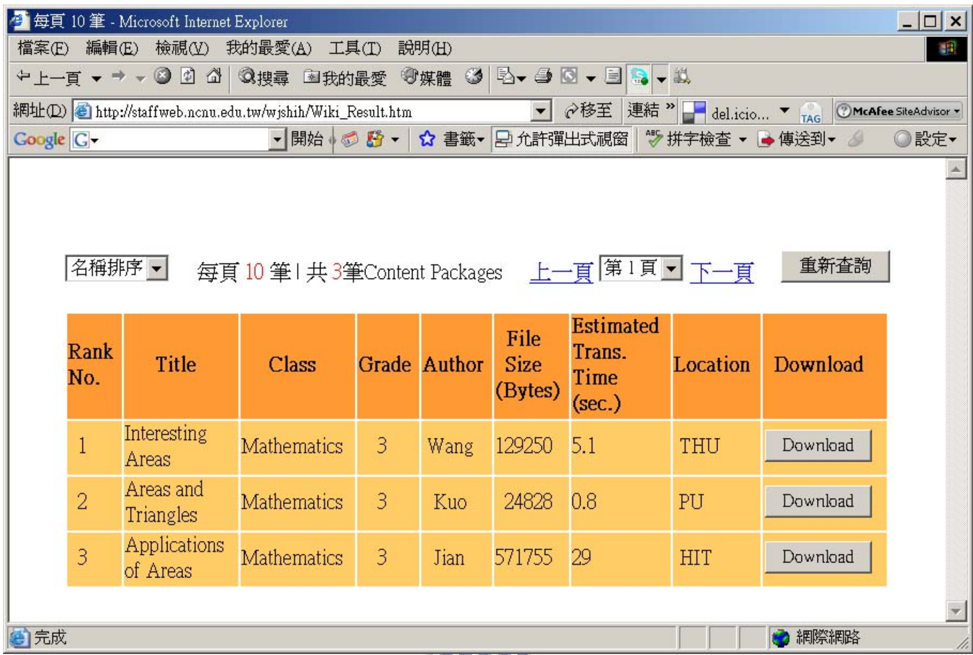
The teachers express their requirement by specifying the keyword "area" and metadata "grade" (its value = 3). The system verifies this query is a single-concept query, and then sends the original query to the next phase.

- Phase 2: query expansion

In order to increase the precision of searching, the teachers define the strategy of query expansion as "Specialization." After inference, the system recommends another three keywords to refine the original query: “Triangle,” “Rectangle,” and “Circle.” The teachers adopt the “Triangle” as an expanded keyword. Consequently, the expanded query, “area and triangle,” is sent to the next phase for searching.

- Phase 3: teaching-material retrieval

According to the expanded query and the specified metadata, three teaching materials are found in the repositories, as shown in Figure 8.5.



Rank No.	Title	Class	Grade	Author	File Size (Bytes)	Estimated Trans. Time (sec.)	Location	Download
1	Interesting Areas	Mathematics	3	Wang	129250	5.1	THU	Download
2	Areas and Triangles	Mathematics	3	Kuo	24828	0.8	PU	Download
3	Applications of Areas	Mathematics	3	Jian	571755	29	HIT	Download

Figure 8.5 The Screenshot of search results

The top two relevant teaching materials are retrieved for draft generation in the next phase. The outlines of the two teaching materials are shown in Figure 8.6. The first teaching material, with the name “Interesting Areas,” consists of five lessons. The second one, with the name “Areas and Triangles,” has six lessons.

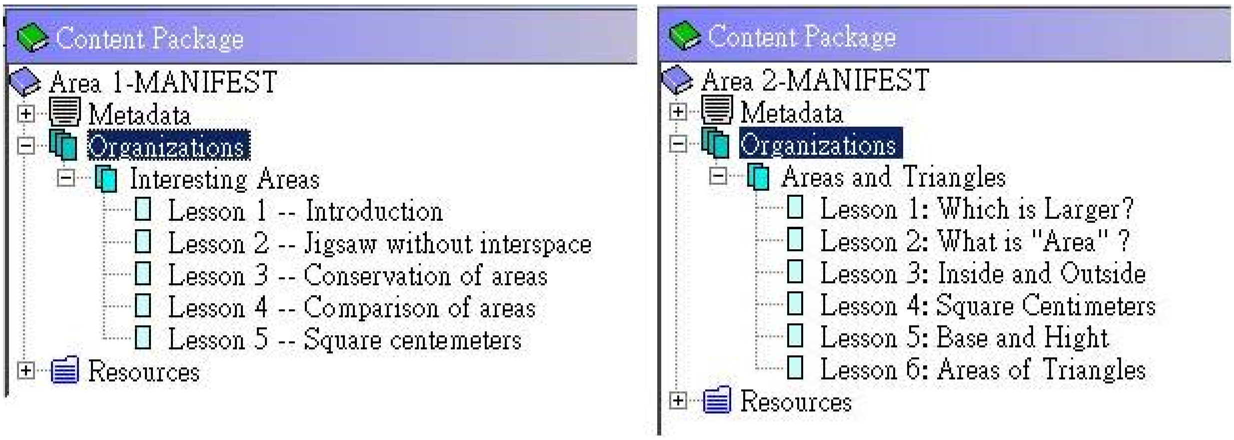


Figure 8.6 The outlines of the two teaching materials

- Phase 4: draft generation

The first version of the draft is automatically generated by merging the teaching materials found in the previous phase. In this phase, redundant modules are removed. For example, both teaching materials have a lesson about “square centimeters.” The two lessons are clustered into one group, and one of them is removed from the draft. Similarly, lesson 2 of the second teaching material is removed after the clustering process. The resultant draft consists of nine lessons. The outline of the draft is shown in Figure 8.7.

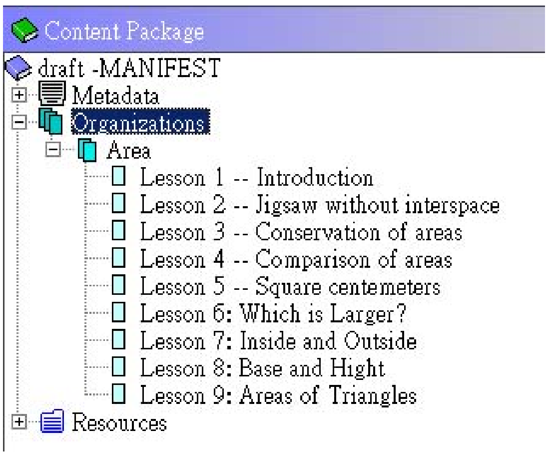


Figure 8.7 The outline of the draft

- Phase 5: Wiki-based revision

The teachers use a Wiki-based authoring tool to facilitate collaborative revision for the draft. Through the Talk page, the revision work is coordinated. In this phase, inappropriate content is modified, and the presentation of content is adjusted. Finally, the teaching material is composed of six lessons, organized into two modules, as shown in Figure 8.8.

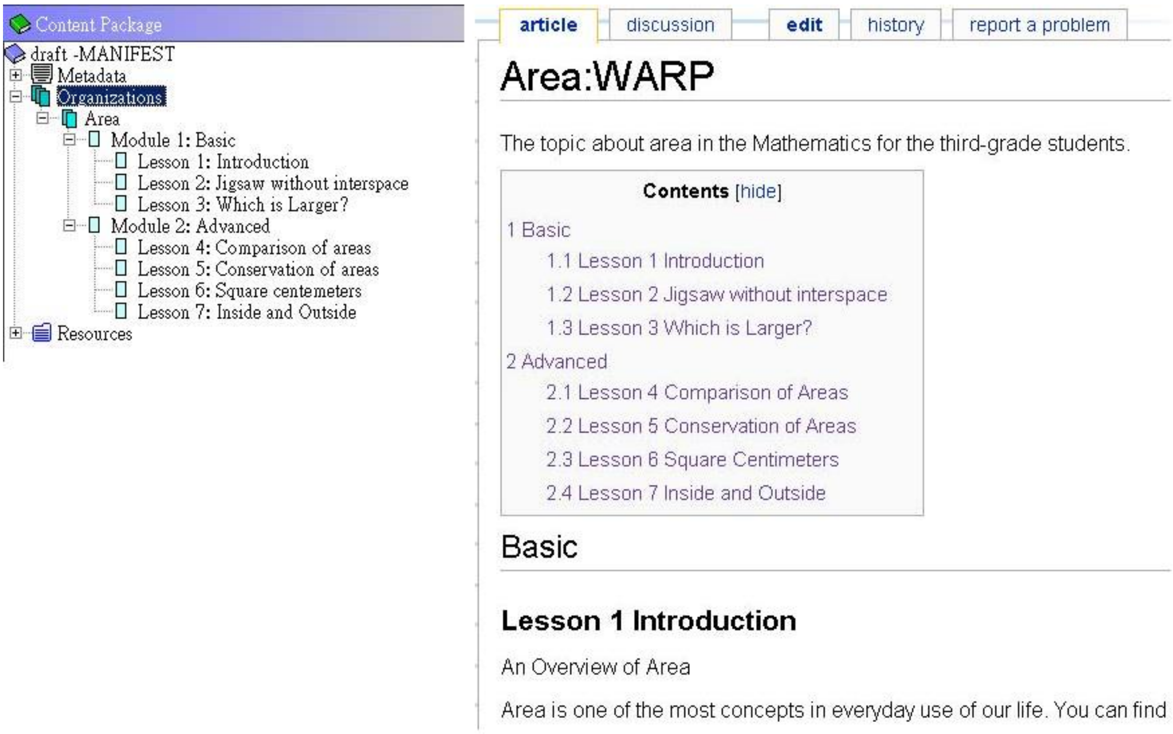


Figure 8.8 The final version (a) outline; (b) Wiki page

8.4 Evaluation on Wiki-based Content design

In order to evaluate the proposed approach, the aforementioned algorithms are to be implemented, and build a prototype for wiki-based authoring. The wiki-based authoring interface is shown in Figure 8.9.

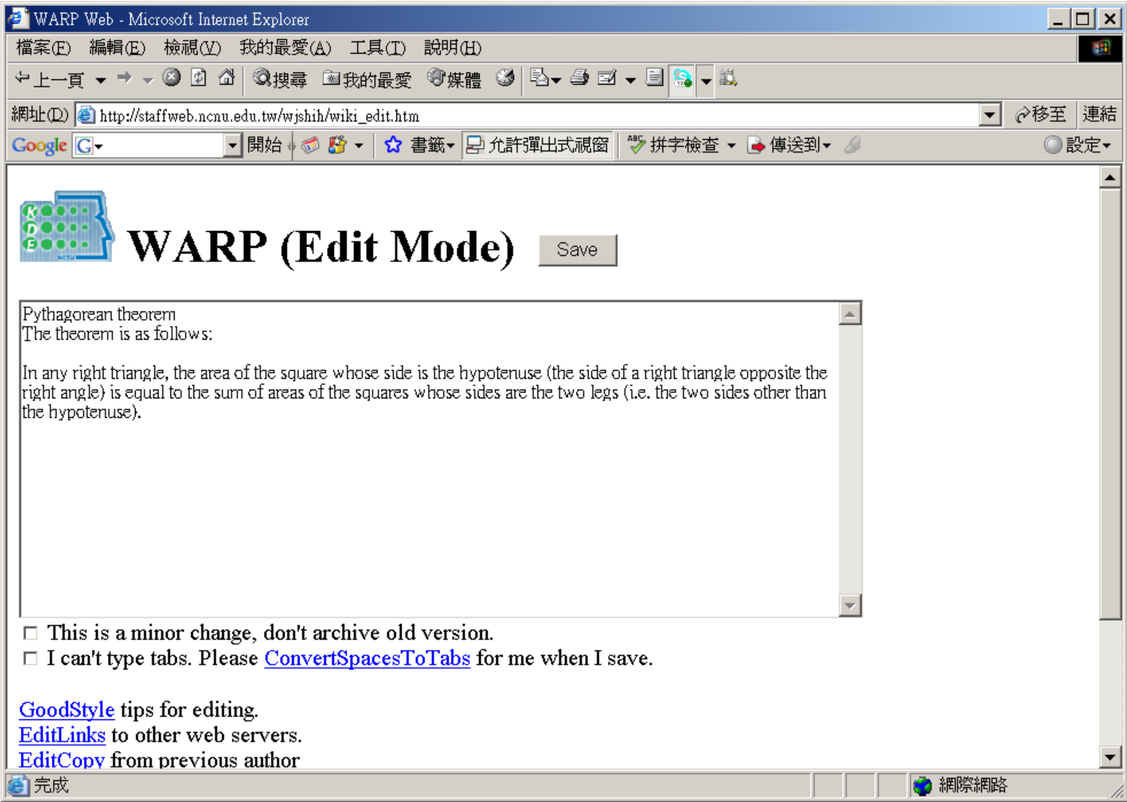


Figure 8.9 Interface of a Wiki-based authoring environment

To elicit the expertise of searching experts, the DRAMA tool [93] is used, which is a suite of toolkits for knowledge engineering developed by KDE Lab. of NCTU. This tool is used for rapid acquisition of searching rules.

The WARP approach is applied to a primary school mathematics Course. Participants are 24 teachers from three primary schools in Nantou, Taiwan. The course is mathematics for the third grade. The existing teaching materials are retrieved from repositories built by Minister of Education, Taiwan.

- Evaluation of WARP

The objective of this evaluation is to answer the question: is the teaching-material development time using WARP significantly shorter than one using a traditional approach?

(1) Experimental Design

A two-group t-test was employed. It is a widely used method to test whether the difference between two means is significant. It can measure the difference of two groups.

(2) Tools

The participants are provided with an internet-enabled environment. That is, they can access information and content available on the web.

(3) Sample

Twenty-four teachers from three primary schools in Nantou, Taiwan, are selected as participants. They are randomly divided into two groups, each with twelve teachers. One is named the experimental group, and the other is named the control group.

(4) Hypothesis

A null hypothesis was set up, which is that no significant difference exists between the development times of the two groups.

(5) Treatment

The experimental group was provided with the WARP environment while the control one was not. That is, teachers of the control group can only search for existing teaching materials and revise them manually. Furthermore, teachers of the experimental group formed a wiki community, and participated in wiki-based revision.

A total of 45 items about feedback from teachers were collected during the Wiki-based authoring process by examining postings on Talk pages, and were classified along the following four dimensions:

- Postings for comments. 26 postings are related to coordination of editing activities. For example, "I would like to suggest pruning lesson 9. The content of evaluating areas seems too difficult for the third-grade students." (Talk page for lesson 9)

- Replies to comments. 14 postings are responses to comments of others. For example, “I agree with that the calculation of area for a triangle is too hard for this stage.” (Talk page for the article on lesson 9, as shown in Figure 8.10).
- Polls. 2 voting sessions were organized by users to decide on controversial editing actions. For example, “The vote is this: Should the above paragraph be included in the lesson? The three possible answers are: Yes, No and Abstain” (Talk page for lesson 3).
- Off-topic remarks. 3 postings are unrelated to the content. For example, “I will suggest my colleagues to try this interesting tool” (Talk page for Area:WARP).

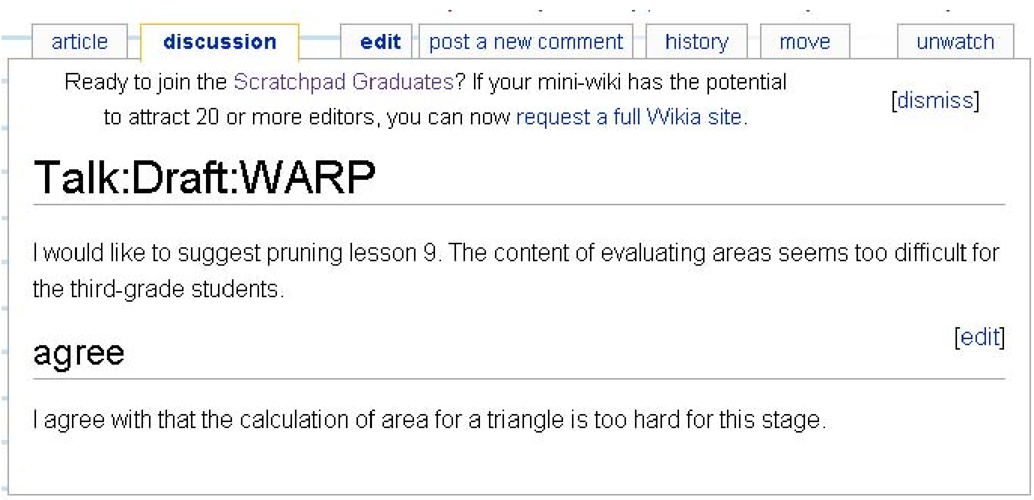


Figure 8.10 Screenshot of Wiki’s Talk pages

- Evaluation of Query Expansion

This experiment investigated whether the proposed intelligent query expansion could enhance the performance of the original query. One experiment shows the precision value from the twelve teachers of the experimental group who used the WARP tool to search for relevant teaching materials. The other experiment will measure the recall value, to see whether the WARP approach could improve the performance of the original query.

This experiment investigated whether the proposed intelligent query expansion could enhance the performance of the original query. Figure 8.11 shows the precision value from the twelve teachers of the experimental group who used the WARP tool to search for relevant teaching materials. The precision values ranged from 0.7 to 1.0 with the WARP and from 0.2 to 0.7 with the original query. The next experiment measured the recall value, as shown in Figure 8.12. Similarly, the WARP approach could improve the performance of the original query.

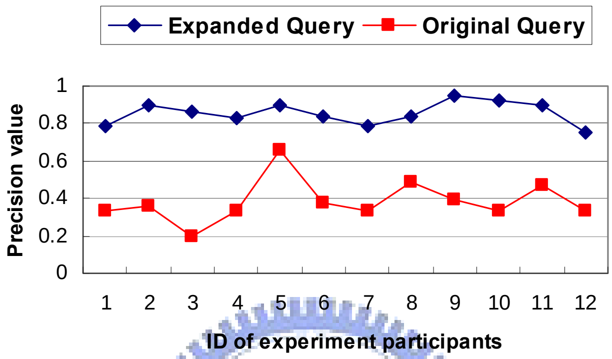


Figure 8.11 Comparison of precision

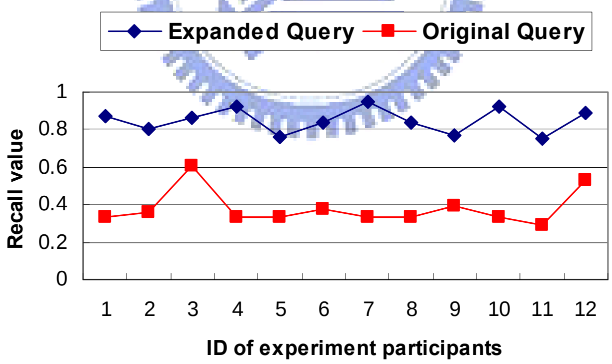


Figure 8.12 Comparison of recall

- Discussion

The Wiki-based revision for teaching material produces both individual and collective benefits. The individual who makes a knowledge contribution can see it immediately published, thus observing the contribution outcome without delay and with pride of authorship. This immediacy between action and positive outcome may very well create a positive reinforcement effect for the author. Immediacy of results has social impacts as well. First, any published result is visible and therefore potentially beneficial to others right away. As others see useful contributions being made, they can use these contributions, as well as build upon them and add their own associated knowledge.

The teaching material produced by the proposed approach has two advantages: variety and innovation. On the one hand, the draft is generated from several relevant teaching materials, which results in its variety of content. On the other hand, the draft is revised by many authors. In this process, different ideas are added in the draft, thus resulting in its innovation.

Wikis enable instant publication of content. As soon as an author saves the new content, it becomes immediately visible to all readers viewing the page. No coordinator is involved in the publication process. Nevertheless, there are safeguards. For instance, Wikis maintain a temporal database of earlier page versions, and roll-back to an earlier version requires only a few clicks. However, from the viewpoint of Wiki designers and administrators, the storage and management for temporal revision are challenges when the Wiki system scales up. Grid platform is a suitable solution to these problems, which can provide resource for the storage and operation of temporal revision.

We discuss the quality of teaching materials produced by WARP in two aspects: content and presentation. First, the quality of the final version heavily depends on the effort of involved authors. The proposed merge algorithm can help to automatically collect relevant learning objects. However, it depends on human authors to refine the draft, such as course sequence, content selection, etc. For example, the draft has evolved from a flat course structure to a two-level hierarchy, which is more organized and understandable for students. Second, currently available Wiki platforms are mostly text-based, and allow users to upload image files. However, multimedia learning objects can not be easily edited on current Wiki platforms. Therefore, in this work, most of the multimedia learning objects in original teaching materials are skipped because of the limitation of Wiki platforms. Consequently, the final version is mainly composed of texts and figures, as shown in Figure 8.13.

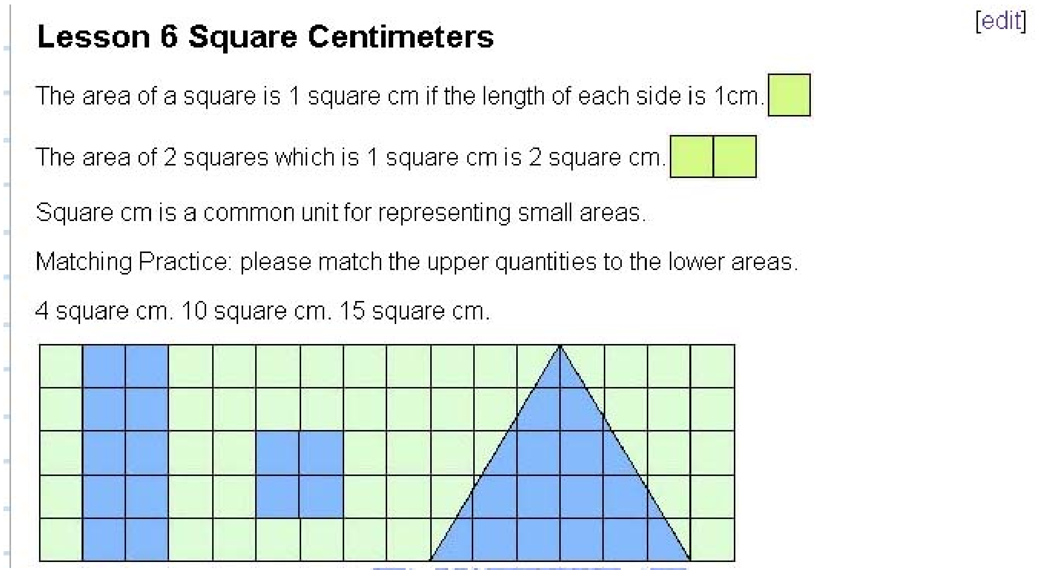


Figure 8.13 Screenshot of Wiki pages

Chapter 9 Conclusion and Future Work

This thesis describes an ontology-based approach to managing teaching materials on grid environments. Three issues related to content management are addressed: Information retrieval, ontology building and content development. To efficiently retrieve learning content on grids, our idea is a bottom-up approach to organize local repositories and generate a global index, which is based on an ontology built from user-defined tagging systems, folksonomies. Furthermore, to facilitate rapidly develop individualized teaching materials, a wiki-based rapid prototyping approach is proposed. The research roadmap is shown as Figure 9.1.

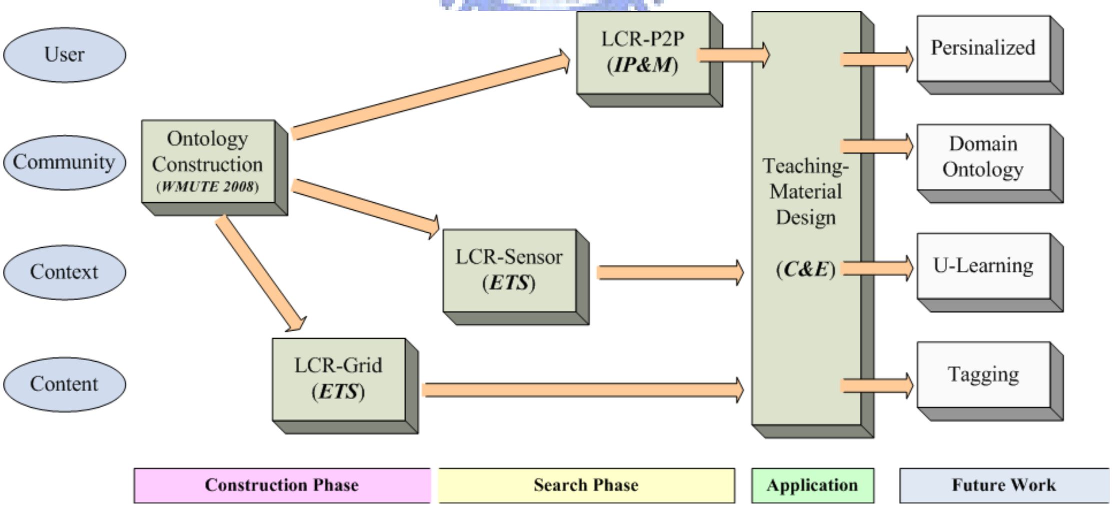


Figure 9.1 Research roadmap

We propose an ontology-based framework to manage learning content on e-Learning grids, aiming at fast and precise content retrieval. The main idea is to increase precision by ontology-based semantic search, and to reduce search time by ontology-based indexing. The idea of ontology building is based on folksonomy, in order to alleviate the heavy burden of experts and knowledge engineers. This framework consists of three phases. In the ontology building phase, users' folksonomies are clustered into a hierarchical ontology, which can be referenced by the index creation phase, where a bottom-up method is designed to organize learning contents located in different sites on grids, according to the built ontology. An ontology-based global index is then created to facilitate semantic search. Finally, users' queries are interactively verified in the search phase, and desired content is retrieved fast and precisely. The ontology-based approach is characterized by a time-saving development process, minimal involvement of experts, reducing redundant effort and high-quality teaching materials and high performance of information retrieval. The evaluation will be carried out on a grid platform.

The contributions can be summarized as follows. First of all, an ontology-based approach is proposed to solve the learning content management problem on e-Learning grids, which is not completely investigated by other existing methods. Also, experimental results reveal that this approach can improve the performance of information retrieval. Second, a novel folksonomy-based method for ontology building is proposed. With this method, the heavy burden of experts for manually developing ontologies can be partially alleviated. Third, a bottom-up method for learning content organization on a grid is proposed. Next, to illustrate the applicability of this approach, this framework is applied to wiki-based rapid prototyping for teaching material design. Finally, a prototype is implemented on a metropolitan-scale grid environment.

Several issues will be further investigated in the future work. In this thesis, the performance issue of location-aware indexing has not been addressed. When the size of the learning object repository grows rapidly, low-level indexing technologies can be adopted to alleviate this issue. Resource sharing and fault tolerance are interesting issues for Grid and P2P applications. The technology of replica management will be incorporated into this framework of learning content retrieval to discuss their effect on content access. In addition, social agreement is an important issue for Wiki-based applications. In recent years, researches on the convergence process of Wiki applications have attracted extensive attention, such as the ontology crystallization problem. These techniques can be applied in the Wiki-based teaching material design process.

In the near future, a maintenance phase to update the ontology and indices will be incorporated into the framework. Content management on e-Learning grids includes many important and challenging issues. After the study of this work, the future work will address Replica Management on grids to speed up the content retrieval process. In addition, it is a promising way to use Expert Systems technologies to facilitate the searching process. Also, Adaptive Information Retrieval will be another future research topic.

Reference

[1] T. O'Reilly, "What Is Web 2.0," 2005.

[2] I. Foster, "The Grid: A New Infrastructure for 21st Century Science," *Physics Today,* vol. 55, pp. 42-47, 2002.

[3] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputer Applications and High Performance Computing,* vol. 11, pp. 115-128, 1997.

[4] M. Gaeta, P. Ritrovato, and S. Salerno, "ELeGI: The European Learning Grid Infrastructure," in *Proceedings of 3rd International LeGE-WG Workshop: GRID Infrastructure to Support Future Technology Enhanced Learning*, 2003, pp. 1-9.

[5] H. Kim, H. Lee, and J. Seo, "A reliable FAQ retrieval system using a query log classification technique based on latent semantic analysis," *Information Processing and Management,* vol. 43, pp. 420-430, 2007.

[6] H. Kim and J. Seo, "High-performance FAQ retrieval using an automatic clustering method of query logs," *Information Processing and Management,* vol. 42, pp. 650-661, 2006.

[7] C.-H. Wu, J.-F. Yeh, and Y.-S. Lai, "Semantic Segment Extraction and Matching for Internet FAQ Retrieval," *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, pp. 930-940, 2006.

[8] S. Fujita, "Technology survey and invalidity search: A comparative study of different tasks for Japanese patent document retrieval," *Information Processing and Management,* vol. 43, pp. 1154-1172, 2007.

[9] I.-S. Kang, S.-H. Na, J. Kim, and J.-H. Lee, "Cluster-based patent retrieval," *Information Processing and Management,* vol. 43, pp. 1173-1182, 2007.

[10] Y. Li and J. Shawe-Taylor, "Advanced learning algorithms for cross-language patent retrieval and classification," *Information Processing and Management,* vol. 43, pp. 1183-1199, 2007.

[11] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. New York: ACM Press, 1999.

[12] A. Mittal, P. V. Krishnan, and E. Altman, "Content Classification and Context-Based Retrieval System for E-Learning," *Journal of Educational Technology & Society,* vol. 9, pp. 349-358, 2006.

[13] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw & Hill, 1983.

[14] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*, 2nd ed. ed. San Francisco, California: Morgan Kaufmann, 1999.

[15] Y. K. Lee, S.-J. Yoo, K. Yoon, and P. B. Berra, "Index structures for structured documents," in *Proceedings of the 1st ACM international conference on digital libraries*, 1996, pp. 91-99.

[16] B. B. Cambazoglu and C. Aykanat, "Performance of query processing implementations in ranking-based text retrieval systems using inverted indices," *Information Processing and Management,* vol. 42, pp. 875-898, 2006.

[17] C. Yu, K. L. Liu, W. Meng, Z. Wu, and N. Rishe, "A Methodology to Retrieve Text Documents from Multiple Databases," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, pp. 1347-1361, 2002.

[18] J. Joseph, M. Emest, and C. Fellenstein, "Evolution grid computing architecture and grid adoption model," *IBM System Journals,* vol. 43, pp. 624-645, 2004.

[19] Y. Li, S. Yang, J. Jiang, and M. Shi, "Building grid-enabled large-scale collaboration environment in e-Learning grid," *Expert Systems with Applications,* vol. 31, pp. 4-15, 2006.

[20] C. T. Yang and H. C. Ho, "An e-Learning Platform Based on Grid Architecture," *Journal of Information Science and Engineering,* vol. 21, pp. 911-928, 2005.

[21] G.-J. Hwang, "Criteria, Strategies and Research Issues of Context-Aware Ubiquitous Learning," *Journal of Educational Technology & Society,* vol. 11, pp. 81-91, 2006.

[22] J. L. Sierra, A. Fernández-Valmayor, M. Guinea, and H. Hernanz, "From Research Resources to Learning Objects: Process Model and Virtualization Experiences," *Journal of Educational Technology & Society,* vol. 9, pp. 56-68, 2006.

[23] R. Oppermann and M. Specht, "Adaptive mobile museum guide for information and learning on demand," in *Proceedings of the 8th International Conference on Human-Computer Interaction*, 1999, pp. 850-854.

[24] G. D. Abowd and C. G. Atkeson, "CyberGuide: A Mobile Context-aware Tour Guide," *Wireless Networks,* vol. 3, 1997.

[25] A. K. Dey and K. M. Futakawa, "The Conference Assistant: Combining context-aware with wearable computing," in *Proceedings of the 3rd International Symposium on Wearable Computers*, San Francisco, CA, USA, 1999, pp. 21-28.

[26] C. Yin, H. Ogata, and Y. Yano, "Ubiquitous-learning system for the Japanese polite expressions," in *Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education* 2005, pp. 269-273.

[27] M. M. El-Bishouty and H. Ogata, "Personalized Knowledge Awareness Map in Computer Supported Ubiquitous Learning," in *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies* Kerkrade, The Netherlands, 2006, pp. 817 - 821

[28] S. J. H. Yang, "Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning," *Educational Technology & Society,* vol. 9, pp. 188-201, 2006.

[29] Z. Cheng, Q. Han, S. Sun, M. Kansen, T. Hosokawa, T. Huang, and A. He, "A proposal on a learner's context-aware personalized education support method based on principles of behavior science," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, Vienna, Austria, 2006, pp. 341-345.

[30] E. D. Nitto, L. Mainetti, M. Monga, L. Sbattella, and R. Tedesco, "Supporting Interoperability and Reusability of Learning Objects: The Virtual Campus Approach," *Journal of Educational Technology & Society,* vol. 9, pp. 33-50, 2006.

[31] H.-C. Wang and C.-W. Hsu, "Teaching-Material Design Center: An ontology-based system for customizing reusable e-materials," *Computers & Education,* vol. 46, pp. 458-470, 2006.

[32] Wikipedia, "Wikipedia," Wikipedia Foundation Inc., 2004.

[33] M.-H. Abel, A. Benayache, D. Lenne, C. Moulin, C. Barry, and B. Chaput, "Ontology-based Organizational Memory for e-learning," *Journal of Educational Technology & Society,* vol. 7, pp. 98-111, 2004.

[34] R. Denaux, V. Dimitrova, and L. Aroyo, "Integrating Open User Modeling and Learning Content Management for the Semantic Web," in *Proceedings of UM2005*, 2005.

[35] D. Sampson, C. Karagiannidis, and F. Cardinali, "An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content " *Journal of Educational Technology & Society,* vol. 5, pp. 27-37, 2002.

[36] H. Funaoi, E. Yamaguchi, and S. Inagaki, "Collaborative concept mapping software to reconstruct learning processes," in *Proceedings of International Conference on Computers in Education*, 2002, pp. 306 - 310.

[37] J. H. McClellan, L. D. Harvel, R. Velmurugan, M. Borkar, and C. Scheibe, "CNT: concept-map based navigation and discovery in a repository of learning content," in *34th Annual of Frontiers in Education (FIE 2004)*, 2004, pp.

13-18.
[38] S. Dehors, C. Faron-Zucker, and R. Dieng-Kuntz, "QBLS:Semantic Web Technology for E-learning in Practice," in *15th International Conference on Knowledge Engineering and Knowledge Management* Czech Republic, 2006.
[39] G.-J. Hwang, "A conceptual map model for developing intelligent tutoring systems," in *Computers and Education*. vol. 40: Elsevier Science, 2003, pp. 217-235.
[40] G. J. Hwang, C. L. Hsiao, and C. R. Tseng, "A Computer-Assisted Approach to Diagnosing Student Learning Problem in Science Course," *Journal of Information Science & Engineering,* vol. 19, pp. 229-248, 2003.
[41] D. F. Salisbury, "Effect drill and practice strategies," in *Instructional Designs for Microcomputer Courseware*: D. H. Jonassen, ed., Lawrence Erlbaum Associates, 1998, pp. 103-124.
[42] S.-S. Weng, H.-J. Tsai, S.-C. Liu, and C.-H. Hsu, "Ontology construction for information classification," *Expert Systems with Applications,* vol. 31, pp. 1-12, 2006.
[43] L. Khan and F. Luo, "Ontology Construction for Information Selection," in *the 14th IEEE internation conference on tools with artificial intelligence*, Washington DC, 2002, pp. 122-127.
[44] A. Hotho, A. Maedche, and S. Staab, "Ontology-based Text Clustering," in *the IJCAI-2001 workshop text learning: Beyond supervision*, Seattle, 2001.
[45] A. Maedche and S. Staab, "Ontology learning for the Semantic Web," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications],* vol. 16, pp. 72-79, 2001.
[46] Protégé, "The Protégé Ontology Editor and Knowledge Acquisition System," 2007.
[47] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens, "OilEd: a Reason-able Ontology Editor for the Semantic Web," in *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI 2001)*, 2001, pp. 396-408.
[48] K. Mahalingam and M. N. Huhns, "Ontology tools for semantic reconciliation in distributed heterogeneous information environ-ments," *Intelligent Automation and Soft Computing,* 1999.
[49] A. Kalyanpur, B. Parsia, E. Sirin, B. C. Grau, and J. Hendler, "Swoop: A Web Ontology Editing Browser," in *Web Semantics: Science, Services and Agents on the World Wide Web*. vol. 4: Elsevier Science, 2006, pp. 144-153.
[50] A. G. Wright, "Semantic Web-Folksonomy," 2004.
[51] A. Farquhar, R. Fikes, and J. Rice, "The Ontolingua Server: a tool for

collaborative ontology construction," in *International Journal of Human-Computer Studies*. vol. 46: Elsevier Science, 1997, pp. 707-727.
[52] J. Bao, Z. Hu, D. Caragea, J. Reecy, and V. G. Honavar, "A Tool for Collaborative Construction of Large Biological Ontologies," in *the 17th International Conference on Database and Expert Systems Applications (DEXA'06)*, 2006, pp. 191-195.
[53] S. Auer, S. Dietzold, and T. Riechert, "OntoWiki - A Tool for Social, Semantic Collaboration," in *5th International Semantic Web Conference*, Athens, GA, USA, 2006, pp. 736-749.
[54] M. D¡browski and S. R. K. Szczecki, "Collaborative Ontology Development with MarcOnt Portal," in *Semantic Technique Conference*, 2007.
[55] A. Diaz and G. Baldo, "Co-Protégé: Collaborative Ontology Building with Divergences," in *Proceedings of the 17th International Con-ference on Database and Expert Systems Applications*, 2006.
[56] M. Li, D. Wang, X. Du, and S. Wang, "Ontology Construction for Semantic Web: A Role-Based Collaborative Development Method," in *Proceedings of the 7th Asia Pacific Web Conference (APWeb)*, 2005.
[57] G. Stumme and A. Madche, "FCA-Merge: Bottom-Up merging of ontologies," in *Proceedings of the 7th Intl. conf. on Artificial Intelligence (IJCAI'01)*, 2001, pp. 225-230.
[58] N. F. Noy and M. A. Musen, "PROMPT: algorithm and. tool for automated ontology merging and alignment," in *Proceedings of the National Conference on Artificial Intelligence*, 2000.
[59] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "The Chimaera Ontology Environment," *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000),* 2000.
[60] F. Ming, T. Yong, and A. B. Whinston, "Evaluation and design of online cooperative feedback mechanisms for reputation management," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, pp. 244-254, 2005.
[61] R. Zhou and K. Hwang, "Gossip-based Reputation Aggregation for Unstructured Peer-to-Peer Networks," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium, 2007. IPDPS 2007.* , 2007, pp. 1-10.
[62] H. Zhuge, X. Chen, X. Sun, and E. A. Y. E. Yao, "HRing: A Structured P2P Overlay Based on Harmonic Series," *Transactions on Parallel and Distributed Systems,* vol. 19, pp. 145-158, 2008.
[63] H. Zhuge and L. Feng, "Distributed Suffix Tree Overlay for Peer-to-Peer

Search," *Transactions on Knowledge and Data Engineering,* vol. 20, pp. 276-285, 2008.

[64] H. Zhuge and X. Li, "Peer-to-Peer in Metric Space and Semantic Space," *Transactions on Knowledge and Data Engineering,* vol. 19, pp. 759-771, 2007.

[65] Z. Yingwu and H. Yiming, "Enhancing Search Performance on Gnutella-Like P2P Systems," *Transactions on Parallel and Distributed Systems,* vol. 17, pp. 1482-1495, 2006.

[66] J. Yuh-Jzer and Y. Li-Wei, "Wildcard Search in Structured Peer-to-Peer Networks," *Transactions on Knowledge and Data Engineering,* vol. 19, pp. 1524-1540, 2007.

[67] H. T. Shen, Y. Shu, and B. Yu, "Efficient semantic-based content search in P2P network," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, pp. 813-826, 2004.

[68] D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "pFusion: A P2P Architecture for Internet-Scale Content-Based Search and Retrieval," *IEEE Transactions on Parallel and Distributed Systems,* vol. 18, pp. 804-817, 2007.

[69] H. Nottelmann and G. Fischer, "Search and browse services for heterogeneous collections with the peer-to-peer network Pepper," *Information Processing and Management,* vol. 43, pp. 624-642, 2007.

[70] J. X. Parreira, S. Michel, and G. Weikum, "p2pDating: Real life inspired semantic overlay networks for Web search," *Information Processing and Management,* vol. 43, pp. 643-664, 2007.

[71] X. Zhu, H. Cao, and Y. Yu, "SDQE: towards automatic semantic query optimization in P2P systems," *Information Processing and Management,* vol. 42, pp. 222-236, 2006.

[72] S. J. H. Yang and I. Y. L. Chen, "A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network," *International Journal of Human - Computer Studies,* vol. 66, pp. 36-50, 2008.

[73] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On unbiased sampling for unstructured peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, Rio de Janeriro, Brazil, 2006.

[74] E. Tsui, "Technologies for Personal and Peer-to-Peer (P2P) Knowledge Management," *Computer Sciences Corporation Leading Edge Forum,* pp. 1-53, 2001.

[75] V. S. Gordon and J. M. Bieman, "Rapid prototyping: lessons learned," *IEEE Software,* vol. 12, pp. 85-95, 1995.

[76] E. O. Kim and J. H. Park, "Study on the rapid prototyping methodology of the lecture contents for the IT SoC certificate program," in *IEEE Int. Conf. Microelectronic Systems Education (MSE'07)*, 2007.

[77] S. Tripp and B. Bichelmeyer, "Rapid prototyping: an alternative instructional design strategy," *Educational Technology Research and Development,* vol. 38, pp. 31-44, 1990.

[78] C. Yang, D. M. Moore, and J. K. Burton, "Managing courseware production: an instructional design model with a software engineering approach," *Educational Technology Research and Development,* vol. 43, pp. 60-70, 1995.

[79] T. S. Jones and R. C. Richey, "Rapid prototyping methodology in action: a development study," *Educational Technology Research and Development,* vol. 48, pp. 63-80, 2000.

[80] A. D. Iorio, A. A. Feliziani, S. Mirri, P. Salomoni, and F. Vitali, "Automatically Producing Accessible Learning Objects," *Journal of Educational Technology & Society,* vol. 9, pp. 3-16, 2006.

[81] R. Lanzilotti, C. Ardito, M. F. Costabile, and A. D. Angeli, "eLSE Methodology: a Systematic Approach to the e-Learning Systems Evaluation," *Journal of Educational Technology & Society,* vol. 9, pp. 42-53, 2006.

[82] A. Kassahun, A. Beulens, and R. Hartog, "Providing Author-Defined State Data Storage to Learning Objects," *Journal of Educational Technology & Society,* vol. 9, pp. 19-32, 2006.

[83] C.-C. Kiu and C.-S. Lee, "Ontology Mapping and Merging through OntoDNA for Learning Object Reusability," *Journal of Educational Technology and Society,* vol. 9, pp. 27-42, 2006.

[84] ELNP, "National Science and Technology Program for e-Learning." vol. 2007, 2002.

[85] R. Nkambou, C. Frasson, and G. Gauthier, "A new approach to ITS-curriculum and course authoring: the authoring environment," *Computers & Education,* vol. 31, pp. 105-130, 1998.

[86] J. W. Coffey, "A meta-cognitive tool for courseware development, maintenance, and reuse," *Computers & Education,* vol. 48, pp. 548-566, 2007.

[87] H.-C. Wang, "Performing a course material enhancement process with asynchronous interactive online system," *Computers & Education,* vol. 48, pp. 567-581, 2007.

[88] J. M. Su, S. S. Tseng, C. Y. Wang, Y. C. Lei, Y. C. Sung, and W. N. Tsai, "A Content Management Scheme in SCORM Compliant Learning Object Repository," *Journal of Information Science and Engineering,* vol. 21, pp. 1053-1075, 2005.

[89] F.-R. Kuo, G.-J. Hwang, Y.-J. Chen, and S.-L. Wang, "Standards and Tools for Context-Aware Ubiquitous Learning," in *Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies*, Niigata, Japan, 2007, pp. 704-705.

[90] C.-M. Chen, Y.-L. Li, and M.-C. Chen, "Personalized Context-Aware Ubiquitous Learning System for Supporting Effectively English Vocabulary Learning," in *Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies*, Niigata, Japan, 2007, pp. 628-630.

[91] SCORM, "Sharable Content Object Reference Model (SCORM)," Advanced Distributed Learning, 2004.

[92] W.-C. Shih, S.-S. Tseng, and C.-T. Yang, "Using Taxonomic Indexing Trees to Efficiently Retrieve SCORM-compliant Documents in e-Learning Grids," *Journal of Educational Technology & Society,* vol. 11, pp. 206-226, 2007.

[93] Y. T. Lin, S. S. Tseng, and C.-F. Tsai, "Design and implementation of new object-oriented rule base management system," *Expert Systems with Applications,* vol. 25, pp. 369-385, 2003.

[94] C.-L. Liu, S.-S. Tseng, and C.-S. Chen, "Design and implementation of an intelligent DNS management system," *Expert Systems with Applications,* vol. 27, pp. 223-236, 2004.

[95] J. Bhogal, A. Macfarlane, and P. Smith, "A review of ontology based query expansion," *Information Processing and Management,* vol. 43, pp. 866-886, 2007.

[96] Dewey, "Dewey Decimal Classification," 2004.

[97] ADDIE, "ADDIE design [On-line]," 2004.

[98] Luqi, "Software evolution through rapid prototyping," *Computer,* pp. 13-25, 1989.

[99] Condor, "Condor Project," 2004.

[100] C.-H. L. Lee and A. Liu, "Modeling the query intention with goals," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications* 2005, pp. 535-540.

[101] G. A. Kelly, *The Psychology of Personal Constructs* vol. 1. New York: W. W. Norton, 1955.