國立交通大學

資訊工程學系

博士論文

基於 SCORM 標準的智慧型學習內容管理系統之研製

Design and Implementation of an Intelligent Learning Content Management System based on SCORM Standard

研究生：蘇俊銘

指導教授：蔡文能　教授

指導教授：曾憲雄　教授

中華民國九十五年七月

基於 SCORM 標準的智慧型學習內容管理系統之研製

Design and Implementation of an Intelligent Learning Content Management System based on SCORM Standard

研 究 生：蘇俊銘　　Student：Jun-Ming Su

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

指導教授：曾憲雄　　Advisor：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

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

基於 SCORM 標準的智慧型學習內容管理系統之研製

學生：蘇俊銘　　　　　　　　　　　　　　　　　指導教授：蔡文能　教授

指導教授：曾憲雄　教授

國立交通大學資訊學院

資訊工程系

摘要

隨著網際網路的快速發展，網路學習(E-Learning)系統已廣為流行。為了解決教材無法在不同網路學習系統間分享與再利用之問題，國際組織已提出許多的國際標準格式，包含:ADL的SCORM、IMS的CP與QTI、IEEE LTSC的LOM、AICC的CMI等等。而SCORM在近幾年已成為最受廣泛使用的標準。SCORM為因應隨時隨地學習之需求，而提供可發展、包裝與傳遞高品質教育及訓練教材的教材標準。雖然SCORM具有分享、再利用、及重新組裝之優點，但對於製作、擷取與管理具個人化學習導引機制的SCORM教材來說，仍是相當困難。此外，如提供所有學習者，相同的學習課程與策略，則學習成效將無法有效提升。於是近幾年來，可根據不同學習者的學習能力與評量結果來提供不同學習課程的適性化學習環境便漸受重視。故對於基於SCORM標準的智慧型網路學習系統而言，如何有效地建立與管理具客製化學習導引與教學策略的SCORM課程、如何根據個人的學習歷程資訊、學習能力及教學策略，自動化提供學習者適當的學習活動、與如何評估及分析學習歷程資料來了解學習者的迷失概念等，便是本論文所關心的研究問題。目前IEEE LTSC組織提出一稱為學習科技系統架構(Learning Technology System Architecture, LTSA)的參考模型，用來定義學習科技系統中的關鍵互操作性介面。而為了支援分散式網路學習系統的互操作性(Interoperability)與延伸性(Scalability)，IMS 抽象架構(Abstract Framework, AF)與網路學習架構(E-Learning

Framework, ELF)規劃出具分層概念的網路學習系統模型，其每一層皆根據網路學習系統不同的需求來定義其不同的功能。

此外，基於知識管理的概念，如何有效管理適性化網路學習系統中的各種資源與資訊，就如同於有效的管理不同的知識。因此，基於知識管理與具分層概念的 LTSA 架構，在此論文中，提出了智慧型學習內容管理系統(**Intelligent Learning Content Management System, ILCMS)**，來智慧地管理大量的學習內容與提供學習者適性化的學習策略，並藉由有效地學習歷程分析，做進一步的策略精練。ILCMS 的分層架構具備 6 個知識模組: **(1)知識表示(KR):**使用 SCORM 標準、本論文提出的教學活動模型(IAM)與物件導向活動模型(OOLA)來表示與管理學習內容及活動、**(2)知識資源(KRes):**儲存學習活動、學習物件、試題、應用程式與學習歷程等學習資源於所屬之資源庫中、**(3)知識管理(KM):**應用叢集(Clustering)技術與負載平衡策略，提出階層式內容管理機制(Level-wise Content Management Scheme, LCMS)來有效管理大量的學習物件、**(4)知識擷取(KA):**提供教師有用的工具來製作 SCORM 與 OOLA 相容的課程與活動，其包含應用高階派翠網路(High Level Petri Nets, HLPN)來分析 SCORM 導引規則而提出的物件導向課程朔模(Object Oriented Course Modeling, OOCM)機制、**(5)知識控制(KC):**智慧地根據學生的學習成效來提供客製化的學習內容、服務、與試題，以進行適性學習、及**(6)知識探勘(KMin)**: 應用資料探勘技術來分析學習歷程資料以建構適性化學習課程與自動地建構學習概念圖。最後，為評估 ILCMS，針對每一知識模組，發展各個相對應的系統功能與實際進行實驗驗證。而藉由實驗結果可證實 ILCMS 所架構的知識模組確實是可行的，且有益於學習者與教師進行有效的學習與教學。

關鍵字: 共享內容物件參考模型(SCORM)、網路學習、知識管理、學習內容管理、適性化學習環境、資料探勘、學習物件、學習歷程分析、概念圖建立。

Design and Implementation of an Intelligent Learning Content Management System based on SCORM Standard

Student: Jun-Ming Su

Advisor: Prof. Wen-Nung Tsai

Advisor: Prof. Shian-Shyong Tseng

Department of Computer Science

College of Computer Science

National Chiao Tung University

Abstract

With the vigorous development of the Internet, e-learning systems have become more and more popular. Currently, in order to solve the issue of sharing and reusing teaching materials in different e-learning systems, several standard formats, including SCORM of ADL, CP and QTI of IMS, LOM of IEEE LTSC, CMI of AICC, etc., have been proposed by international organizations. Among these international standards, the Sharable Content Object Reference Model (SCORM) has become the most popular standard in recent years.

SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed. Although SCORM has many advantages of reusing, sharing, and recombining teaching materials among different standards, it is difficult to create, retrieve, and manage the SCORM compliant course with personalized learning sequences. Moreover, if the same teaching materials are provided to all learners based on predefined strategies, the leaning efficiency will be diminished. Thus, in recent years, adaptive learning

environments have been proposed to offer different teaching materials for different students in accordance with their aptitudes and evaluation results. Therefore, for the intelligent e-learning system based upon SCORM standard, how to efficiently create and manage the SCORM compliant learning contents with desired learning sequencing and teaching strategies, how to automatically generate appropriate learning activity for learners according to individual learning portfolio, personal aptitude, and teaching strategies, and how to evaluate the historical learning portfolio for understanding the mis-concept of learners are our concerns.

Currently, the IEEE's LTSC proposed a Learning Technology System Architecture (LTSA) which is as a reference model and identifies the critical interoperability interfaces for learning technology systems. In addition, in order to support the interoperability and scalability of distributed e-learning system, IMS Abstract Framework (AF) and E-Learning Framework (ELF) propose the e-learning system models with layering concept, each layer of which defines different functionalities according to the different requirements of an e-learning system.

Furthermore, based on the Knowledge Management concept, how to efficiently manage the different resources and information in an adaptive e-learning system is similar to efficiently manage diverse knowledge. Therefore, based on this concept and LTSA with layering concept, in this dissertation, an **Intelligent Learning Content Management System (ILCMS)** is proposed to intelligently manage a large number of learning contents and offer learners an adaptive learning strategy which can be refined by means of efficient learning portfolio analysis.

The layered architecture of ILCMS consists of six knowledge modules: **1) Knowledge Representation (KR)**, which uses SCORM standard, and proposed Instructional Activity Model (IAM) and Object Oriented Learning Activity (OOLA) model to represent and manage the learning content and activity, **2) Knowledge**

Resources (KRes), which stores related learning resources including Learning Activity, Learning Object, Test Item, Application Program, and Learning Portfolio in respective repositories, **3) Knowledge Manager (KM)**, which includes a Level-wise Content Management Scheme (LCMS), applying clustering approach and load balancing strategies, to efficiently manage a large number of learning resources, **4) Knowledge Acquirer (KA)**, which provides teachers with useful tools to create the SCORM and OOLA compliant learning content and activity by means of proposed Object Oriented Course Modeling approach based on High Level Petri Nets and OOLA model, **5) Knowledge Controller (KC)**, which intelligently delivers the desired learning contents, services, test sheet to learners according to her/his learning results and performance, and **6) Knowledge Miner (KMin)**, which applies data mining techniques to analyze the learning portfolio for constructing the adaptive learning course and the learning concept map automatically. Finally, in order to evaluate ILCMS, system implementations and experiments have been done for each knowledge module. Also, the experimental results shows that proposed knowledge modules of ILCMS are workable and beneficial for learners and teachers.

Keywords: Sharable Content Object Reference Model (SCORM), E-Learning, Knowledge Management, Learning Content Management, Adaptive Learning Environment, Data Mining, Learning Object, Learning Portfolio Analysis, Concept Map Construction.

誌 謝

本論文能順利完成，首先需要感謝的便是我的指導教授 蔡文能教授與 曾憲雄教授。如沒他們在我博士求學生涯中，遭遇最艱困的時候，適時的伸出援手，並給予我支持與鼓勵，則今日我無法順利完成此論文。

尤其更加感謝 曾憲雄教授，在曾教授不厭其煩的指導下，讓我對此博士論文的研究領域從陌生到熟悉，從疑惑到了解。曾教授對我的指導並非僅止於論文研究，他豐富的實務經驗與做人處事態度，著實令我獲益良多，因此，在這博士求學過程中，我學到的不僅僅是研究的方法，更獲得許多寶貴的經驗與生活的價值體認。雖然感謝二字不足以形容整個過程，但是，心中最真誠的感謝仍想要藉此機會表達。

此論文的完成，也非常感謝從校內口試到校外口試過程中，一路給予我許多論文修改建議的 李素瑛教授與 孫春在教授；以及在校外口試中，給予我寶貴意見的中央大學 陳國棟教授、台灣師範大學葉耀明教授、中央大學 楊鎮華教授與清華大學 張智星教授，讓此論文能夠更加完整與契合。

當然，也不能忘了在這論文研究其間，知識工程實驗室所有和我一起研究、打拼的研究夥伴們，亦讓我深切體認團隊合作的重要與價值。其中，尤其感謝瑞鋒在此論文研究上的辛苦幫忙與協助，讓此研究更加順利與扎實。

最後，家人的體諒與支持，是讓我能夠順利且安心完成博士學業的最大後盾，雖然此段求學過程，相當漫長與坎坷，遇到不少挫折與挑戰，如無父母親無怨無悔的支持與鼓勵，相信今日我難以順利完成此論文研究，順利取得博士學位。當然，更加感謝我那美麗賢慧、善良體貼的女友 蕙瑜，一路陪伴我走過這漫長與艱困的博士求學生涯，多虧她的付出與支持，讓本論文充滿了更多的愛與活力，實在衷心感謝。

僅將此份倫文，獻給所有支持我及我愛的家人、師長、女友與朋友們。

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1 Introduction

As internet usage becomes more and more popular over the world, e-learning system including online learning, employee training courses, and e-book in the past ten years has been accepted globally [6] [13] [20] [52] [53] [61] [69] [86] [88] [99] [130] [131] [138] [143]. In 2000, Urdan et al. [128] considered that e-learning is defined more narrowly than distance learning and defined it as "*the delivery of content via all electronic media, including the Internet, intranets, extranets, satellite broadcast, audio/video tape, interactive TV, and CD-ROM*". E-learning system can make learner conveniently study at any time and any location. However, because the teaching materials in different e-learning systems are usually defined in specific data format, the sharing of the materials among these systems becomes difficult, resulting in increasing the cost of creating teaching materials. In order to solve the issue of the uniform teaching materials format, several standard formats including SCORM (Sharable Content Object Reference Model) of ADL [100], CP (Content Packaging) and QTI (Question & Test Interoperability) of IMS [56], CMI (Computer-Managed Instruction) of AICC [1], LOM (Learning Objects Metadata) of IEEE LTSC [79], etc. have been proposed by international organizations. By these standard formats, the teaching materials in different learning management systems can be shared, reused, and recombined.

SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed. SCORM-compliant courses leverage course development investments by ensuring that compliant courses are **R**eusable**,** **A**ccessible**,** **I**nteroperable**,** and **D**urable **(RAID).** Although SCORM has many advantages of reusing, sharing, and recombining teaching materials among different standards, it is difficult to create, retrieve, and manage the

SCORM compliant course with personalized learning sequences based on the pedagogical theory. For example, the work to create the SCORM compliant teaching materials is still hard, even using the authoring tools. This leads to that teachers or editors may be unwilling to use it.

As we know, if the same teaching materials are provided to all learners based on the predefined strategies or the predefined learning maps, the leaning efficiency will be diminished. Thus, in recent years, adaptive learning environments [22] [43] [98] [111] [113] [122] [123] [140] have been proposed to offer different teaching materials for different students in accordance with their aptitudes and evaluation results. After students learn the teaching materials through the adaptive learning environment, the teachers can further analyze the historical learning records and then refine or reorganize the teaching materials and tests if needed. Therefore, more and more attention has been paid to the research of personalized instruction in computer education environment.

Moreover, because sequencing can help to generate teaching materials which can match the learner's needs, (semi-)automatic sequencing of course materials also becomes an important research issue. However, although the personalized instruction scheme has been emphasized in most of existing e-learning systems, these systems, unfortunately, may not show good personalized and intelligent abilities.

Therefore, to sum up above, for the intelligent e-learning system, the following issues are needed to be solved.

- How to propose a scheme to efficiently create and manage the SCORM compliant learning contents with the desired learning sequencing.
- How to propose a scheme to efficiently create and manage the teaching strategies.
- How to propose an intelligent approach which can automatically generate appropriate learning activity for learners according to the individual learning portfolios, personal aptitudes, and teaching strategies.

- How to propose an efficient approach to evaluate the historical learning portfolio for understanding the mis-concept of learners.

At present, the international organization, IEEE LTSC, analyzed the basic requirements of e-learning system to propose a Learning Technology System Architecture (LTSA) [80] which is as a reference model and identifies the critical interoperability interfaces for the learning technology systems. LTSA, including 4 processes and 2 stores, that is, **Learner Entity**, 2) **Coach**, 3) **Delivery**, 4) **Evaluation**, 5) **Learner Record**, and 6) **Learning Resource**, can provide learners with an adaptive learning environment.

In addition, in order to support the interoperability and scalability of distributed e-learning system, IMS Abstract Framework (AF) [55] proposes a layered model, which defines the interface definition set. Also, E-Learning Framework (ELF) [35] also proposes a layered model, each layer of which defines different functionalities according to the different requirements of an e-learning system. Therefore, based on the layered models of IMS AF and ELF, LTSA reference model can be reorganized into 4 layers: **Resources**, **Common Services**, **Learning Services**, and **Application**, according to the functions of its components.

Besides, because IEEE LTSA is as a reference model of building an e-learning system in support of adaptive learning, it does not clearly specify and define the data format of learning content and activity. Therefore, in order to solve the issue of uniform data format among e-learning systems, how to define the data representation format of learning content and activity is a very important issue.

Furthermore, based on the Knowledge Management concept [39], how to efficiently manage the different resources and information in an adaptive e-learning system is similar to efficiently manage diverse knowledge. Therefore, based on this concept and

IEEE LTSA [80] with layering concept, in this dissertation, an **Intelligent Learning Content Management System (ILCMS)** is proposed to intelligently manage a large number of learning contents and offer learners an adaptive learning strategy which can be refined by means of efficient learning portfolio analysis. The layered architecture of ILCMS consisting of six knowledge modules in corresponding layer respectively, i.e., **1) Knowledge Representation (KR)**, which uses SCORM standard, and new proposed Instructional Activity Model (IAM) [115] and Object Oriented Learning Activity (OOLA) model to represent and manage the learning content and activity, **2) Knowledge Resources (KRes)**, which stores all related learning resources in repositories, **3) Knowledge Manager (KM)**, which efficiently manage a large number of learning resources in repositories, **4) Knowledge Acquirer (KA)**, which provide teachers with useful tools to create the SCORM and OOLA compliant learning content and activity, **5) Knowledge Controller (KC)**, which intelligently deliver the desired learning contents, services, test sheet to learners according to her/his learning results and performance, and **6) Knowledge Miner (KMin)**, which analyzes the learning portfolio to analyze the learning portfolio for constructing the adaptive learning course and the learning concept map automatically.

As mentioned above, the relationship of six knowledge modules in ILCMS are described as follows. First of all, **KRes Module** consists of five types of learning resources, i.e., **Learning Activity**, **Learning Object**, **Test Item**, **Application Program**, and **Learning Portfolio**, which are described by data formats: SCORM and OOLA model defined in **KR Module** and stored in their respective repositories. Then, **KA** module includes a **Learning Content Editor (LCE)** and an **Object Oriented Learning Activity (OOLA) authoring tool** [81]. In LCE, for reusing the existing traditional teaching materials, such as HTML and PPT file format, a **Content Transformation Scheme (CTS)** [114] has also been proposed. CTS approach can divide a traditional teaching material into separate learning objects with SCORM metadata and then package them into one SCORM course. Moreover, in order to edit SCORM 2004 compliant learning contents, an **Object Oriented Course Modeling (OOCM)** [117] approach based upon High Level Petri Nets (HLPN) theory [59] [60] [62] [70] [71][73] [82] [84] has been proposed. OOCM can provide teachers or editors with an authoring tool to efficiently construct the SCORM compliant course with desired sequencing behaviors. Furthermore, OOLA authoring tool can help teachers construct an OOLA learning activity with desired teaching strategy. Moreover, **KM** module includes a **Learning Object Repository (LOR) Manager**, where we apply clustering approach and load balancing strategies to propose a management approach, called **Level-wise Content Management Scheme (LCMS)** [116], to efficiently maintain, search, and retrieve the learning contents in SCORM compliant LOR. When learners initiate a learning activity, the **Learning Activity Controller (LAC)** in **KC** module will retrieve the appropriate learning objects in LOR, testing sheets in Testing Item Bank (TIB), or application program (AP) in AP Repository (APR) according to the personalized learning activity in Learning Activity Repository (LAR) for learners. As mentioned above, the learning contents, test sheet, and AP will be retrieved and triggered according to the specific learning strategy. Those strategies are created by teachers using the authoring tools in KA module. Furthermore, **KMin** module includes a Learning Portfolio Analyzer (LPA), which consists of **Learning Portfolio Mining (LPM)** [118] and **Two-Phase Concept Map Construction (TP-CMC)** [110] algorithms. According to learners' characteristics, the former applies the clustering and decision tree approach to analyze the learning behaviors of learners with high learning performance for constructing the adaptive learning course. The latter applies Fuzzy Set Theory and Data Mining approach to automatically construct the concept map by learners' historical testing records. Therefore, after the learners finished the learning activities, teachers can use LPA module to analyze the learning portfolios of learners for refining their teaching strategies and contents.

The rest of this dissertation is organized as follows. Chapter 2 surveys the background knowledge of this work. Chapter 3 describes the layered architecture of LTSA and introduces the six modules of ILCMS. From Chapter 4 to Chapter 7, the details of Knowledge Representation (KR), Knowledge Acquirer (KA), Knowledge Manager (KM), Knowledge Controller (KC), and Knowledge Miner (KMin), are described. The system implementation and experimental results of ILCMS are shown in Chapter 9, and finally conclusion and future work are given in Chapter 10.

Chapter 2 Related Works

2.1 Intelligent Tutoring System and Adaptive Learning Environment

In 1989, Johnson et al. [61] proposed a software design and development research program called Microcomputer Intelligence for Technical Training (MITT). In order to organize system knowledge and operational information for enhancing the operator performance, Vasandani et al. also developed an intelligent tutoring system [130] [131]. Furthermore, Hwang proposed an intelligent tutoring environment to detect the on-line behaviors of students [52]. Afterward, many related articles had also been proposed to develop the tutoring systems and learning environments [53] [69] [88][138] [143].

In adaptive learning environment, Shang [111] proposed an intelligent environment for active learning to support the student-centered, self-paced, and highly interactive learning approach. The learning environment can use the related learning profile of student, e.g., learning style and background knowledge, to select, organize, and present the customized learning materials for students. Trantafillou [122] also proposed an adaptive learning system, called AHS, in which Learners can be divided into two groups with Field Independence (FI) and Field Dependence (FD) respectively according to their cognitive styles. Then, the AHS system can provide appropriate strategy and learning materials for different groups. Moreover, according to learning styles and learning experience of learners, Gilbert [43] applied the Case Based Reasoning (CBR) technique to assign a new learner to the most similar one of four groups. Based upon the learning experience in group selected by CBR, the proposed system can offer the new learner an adaptive learning material. However, in all systems mentioned above, the information and approaches used to represent and group learners respectively are too easy to provide learners with personalized learning materials.

Carchiolo *et al.* [22] had proposed adaptive formative paths for e-learning environments. They constructed a domain database and student profiles to obtain personalized learning paths. During the learning process, the learning paths can be dynamically modified according to student needs and capabilities. Although this system has some advantages, including consideration of each student's prior knowledge and generation of an adaptive learning path, it does not take pedagogical theory into account, and it is not yet compatible with the SCORM standard. Sheremetov and Arenas [98] also proposed a system, called EVA, for developing a virtual learning space at the National Technical Institute in Mexico. EVA consists of five virtual learning spaces: 1. the Knowledge Space, in which all necessary information exists; 2. the Collaborative Space, in which real or virtual companions get together to learn; 3. the Consulting Space, in which the teachers or tutors (also real or virtual) guide learning and provide consultation; 4. the Experimentation Space, in which the practical work is done by the students in the virtual environment; and 5. the Personal Space, in which records of users are stored. The model of knowledge is represented in the form of graph, where each node, the basic element of the knowledge structure, is a unit of learning material (ULM). ULMs with a related knowledge concept can be grouped into a POLIlibro (or Multi-Book) along the learning trajectory (path), depending on the students. However, the relations between ULMs are not sufficient to express the structure of the knowledge model, and the attributes of a ULM are insufficient for mining the behaviors of students. The authors also proposed some methods for planning trajectories and scheduling learning activities based on the agent technology. However, how to generate a learning path was not discussed.

Therefore, the development of intelligent tutoring system (ITS) or adaptive learning system (ALE) has become an important issue in both computer science and education.

2.2 International Standards in E-Learning System

However, most existing e-learning systems represent student profile, learning management data, test bank and subject contents with different formats, which results in the difficulties of sharing, reusing, and recombining those e-learning resources. Therefore, several international organizations have proposed teaching material standards, such as SCORM proposed by IMS, Simple Sequencing Specification and Content Packaging proposed by IMS, and LOM proposed by IEEE LTSC.

2.2.1 IMS (Instructional Management System)

In 1997, the IMS Project [56], which is part of the nonprofit EDUCAUSE [33], started its work and developed open, market-based standards including specifications of learning resource metadata for online learning. In the same year, the NIST (National Institute for Standards and Technology) and the IEEE P.1484 group, which now is the IEEE Learning Technology Standards Committee (LTSC) [79], also started to do a similar effort. Then, the IMS collaborated with NIST and ARIANDE project [3]. In 1998, IMS and ARIADNE submitted a joint proposal and specification to the IEEE, which is the basis of current IEEE Learning Object Metadata (LOM) base document. Currently, the IMS project have proposed many standard specifications including learning metadata specification, content packaging specification, learner profiles specification, question and test interoperability, and simple sequence specification, etc.

2.2.2 IEEE LTSC

The international organization, IEEE LTSC [79], proposed the Learning Technology System Architecture (LTSA) and Learning Objects Metadata (LOM), described as follows.

LOM (Learning Objects Metadata):

The IEEE's Learning Objects Metadata (LOM) [79] describes the semantics of learning object metadata. Here, a learning object is defined as any entity, including multimedia content, instructional content, and instructional software, which can be used, reused, shared, and recombined. To allow learning objects to be managed, located, and evaluated, the LOM standard makes efforts in the minimal set of properties needed.

The LOM describes learning resources by using the following categories.

- **General:** describe the general information of learning resource.
- **LifeCycle:** describe the history and current state of learning resource and its evolution information.
- **Meta-MetaData:** describe the specific information about the metadata record itself, e.g., who created this metadata record, etc.
- **Technical:** describe the technical requirements and characteristics of learning resource.
- **Educational:** describe the key educational or pedagogic characteristics of learning resource.
- **Rights:** describe the intellectual property rights and conditions of use for learning resource.
- **Relation:** define the relationships among this resource and other targeted resource.
- **Annotation:** provide comments on the educational use of learning resource, e.g., who created this annotation.
- **Classification:** describe classification criteria and hierarchy of learning resource.

Learning Technology System Architecture (LTSA):

IEEE LTSC analyzed the basic requirements of e-learning system to propose a Learning Technology System Architecture (LTSA) [80] which identify the critical interoperability interfaces for learning technology systems. LTSA mainly includes 4 processes and 2 stores, described as follows:

(1) Learner Entity: learners receive the multimedia learning contents delivered by system and the learning progress of learners will be tracked and recorded.

(2) Coach: teachers provide learning system with teaching materials and evaluate the learning performance of learners.

(3) Delivery: it is responsible for delivering the learning contents Coach indicates to learners.

(4) Evaluation: it evaluates the learning performance of learners and diagnoses the mis-concept.

(5) Learner Record: it records the learning behavior of learners, which can be used to analyze and track.

(6) Learning Resource: it stores the learning resources which were created by teachers and can be used to learn for learners.

Figure 2.1 illustrates the components of LTSA system.

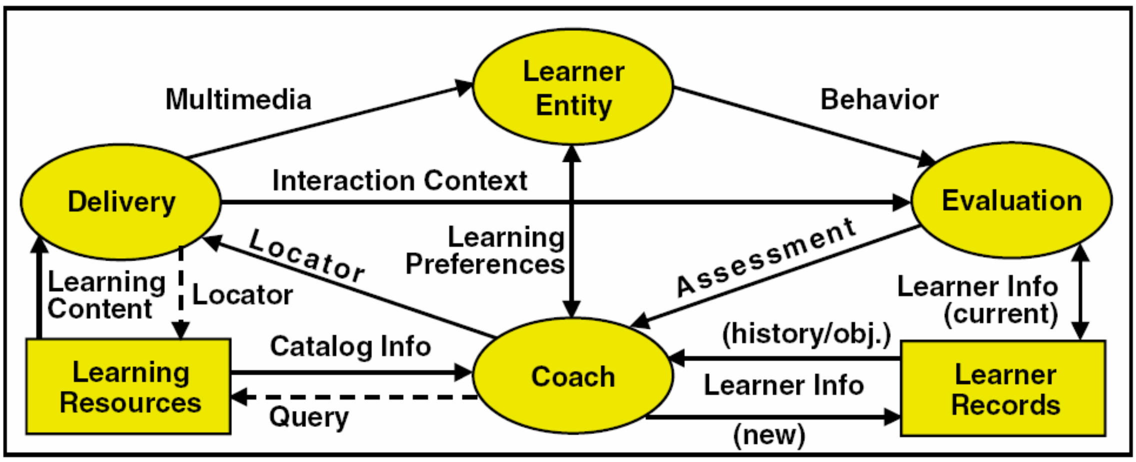


Figure 2.1: The LTSA System Components

2.2.3 SCORM (Sharable Content Object Reference Model)

Among those existing standards for learning contents, SCORM [100], which was proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997, is currently the most popular one. The SCORM specifications are a composite of several specifications developed by international standards organizations, including the IEEE LOM [79], IMS [56], AICC [1] and ARIADNE [3]. 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. SCORM-compliant courses leverage course development investments by ensuring that compliant courses are "RAID:" Reusable: easily modified and used by different development tools, Accessible: can be searched and made available as needed by both learners and content developers, Interoperable: operates across a wide variety of hardware, operating systems and web browsers, and Durable: does not require significant modifications with new versions of system software [58]. The details of SCORM and its Sequencing & Navigation (SN) [109] will be described in Chapter 4.

2.3 The SCORM Compliant Authoring System

Recently, although many SCORM authoring tools have been developed by commercial companies, unfortunately, these tools support SCORM 1.2 only, for example, the Authorware 7 of Macromedia [64], Click2learn Unveils SCORM 1.2 Resource Kit [23], Seminar Author of Seminar Learning System [105], Elicitus Content Publisher [31], and more other SCORM 1.2 compliant authoring tools found in [32].

Because the complicated sequencing rule definitions of SN in SCORM 2004 make the design and creation of course hard, the article in [76] has proposed several document templates to construct SCORM compliant course according to the sequencing definitions of SN. Teachers/authors can design their desired learning activities by modifying the sequencing definitions in document templates. Then, the SCORM course with sequencing definitions can be created by programming. However, for teachers/authors, creating the SCORM course with sequencing behavior rules by document templates is still hard. Moreover, it is time consuming and high cost to create SCORM course by programming.

Moreover, an open source tool, called Reload Editor, developed by [94] can be used to create the SCORM 2004 course. For setting the learning guidance, users have to edit the sequencing rules by clicking in the comboBox of sequencing rules. Although it offers the graphical user interface (GUI) to create SCORM course, the sequence of final course is hard to image and creating course is also time-consuming. Shih et al. [103] also proposed a collaborative courseware authoring tool to edit the SCORM compliant course which can support collaborative authoring and suggest an optimal learning sequence. They analyzed the metadata of SCA in SCORM 1.3 to design the activity rules which can be used to generate lecture sequencing. This tool also offers users the sequencing rules definition page to define the sequencing behavior of courseware. Besides, Yang et al. [144] developed a web-based authoring tool, called Visualized Online Simple Sequencing Authoring Tool (VOSSAT), to provide an easy-to-use interface for editing existing SCORM-compliant content packages with sequencing rules. Nevertheless, the disadvantages in [103][144] are the same as Reload Editor [94].

2.4 Applying Petri Nets in E-Learning System

Lin [70] applied Petri Nets theory to model online instruction knowledge for developing online training systems. Two-level specialized Petri nets including TP-net, which represents goal-oriented training plans, and TS-net, which represents task-oriented training scenarios, are proposed. A Goal-Oriented Training Model Petri net (GOTM-net), which is combined by a TP-net and all TS-nets, is converted as a set of "*if-then*" rules representing the behaviors a learner may perform and the corresponding responses. However, GOTM-net may not be compatible with SCORM standard. Based on SCORM 1.2, Liu et al. [71] discussed meta-data structure which makes a base for reusing and aggregating learning resources in e-learning, and provided an aggregation model, called Teach net, based on High-Level Petri Nets (HLPN). Several routing constructs in workflow are also modeled by HLPN for flexible navigation. However, the Teach net is mainly used to model the content aggregation without considering course sequencing. Besides, the modeled routing constructs may be not sufficient for modeling sequencing definition in SCORM 2004.

2.5 Structured Document Management

For fast retrieving the information from structured documents, Ko et al. [63] proposed a new index structure which integrates the element-based and attribute-based structure information for representing the document. Based upon this index structure, three retrieval methods including 1) top-down, 2) bottom-up, and 3) hybrid are proposed to fast retrieve the information from the structured documents. However, although the index structure takes the element and attribute information into account, it is too complex to be managed for the huge amount of documents.

How to efficiently manage and transfer document over wireless environment has become an important issue in recent years. The articles [75] [142] have addressed that retransmitting the whole document is expensive in faulty transmission. Therefore, for efficiently streaming generalized XML documents over the wireless environment, Wong et al. [133] proposed a fragmenting strategy, called Xstream, for flexibly managing the XML document over the wireless environment. In the Xstream approach, the structural characteristics of XML documents has been taken into account to fragment XML contents into an autonomous units, called Xstream Data Unit (XDU). Therefore, the XML document can be transferred incrementally over a wireless environment based upon the XDU. However, how to create the relationships between different documents and provide the desired content of document have not been discussed. Moreover, the above articles [63] [75] [133] [142] didn't take the SCORM standard into account yet.

2.6 Learning Portfolio Analysis

In addition, for learning portfolio analysis, Chen [15][16] applied decision tree and data cube techniques to analyze the learning behaviors of students and discover the pedagogical rules on students' learning performance from web logs including the amount of reading article, posting article, asking question, login, etc. According to their proposed approach, teachers can easily observe learning processes and analyze the learning behaviors of students for pedagogical needs. However, although their proposed approaches can observe and analyze the learning behavior of students, they don't apply education theory to model the learning characteristics of learners. Therefore, the learning guidance can not be provided automatically for the new learner. For providing the personalized recommendation from historical browser behavior in e-learning system, Wang [140] proposed a personalized recommendation approach which integrates user clustering and association-mining techniques. Based upon a specific time interval, they divided the historical navigation sessions of each user into frames of sessions. Then, a new clustering method, called HBM (Hierarchical Bisecting Medoids Algorithm) was proposed to cluster users according to the time-framed navigation sessions. In the same group, the association-mining technique was used to analyze those navigation sessions for establishing a recommendation model. Thus, this system can offer the similar students personalized recommendations. However, in this approach, the learning characteristics and sequential learning sequence of students were not considered, so that the personalized recommendation may be not appropriate. Of course, it doesn't support SCORM 2004 standard yet.

2.7 Concept Map Construction

In 1984, Novak [85] proposed Concept Map to organize or represent the knowledge as a network consisting of nodes (points/vertices) as concepts and links (arcs/edges) as the relations among concepts. Thus, a wide variety of different forms of concept maps have been proposed and applied in various domains [8][45][46]. In the adaptive learning environment, the Concept Map can be used to demonstrate how the learning status of a concept can possibly be influenced by learning status of other concepts and give learners adaptive learning guidance. Thus, Appleby proposed an approach to create the potential links among skills in math domain [5]. The direction of a link is determined by a combination of educational judgment, the relative difficulty of skills, and the relative values of cross-frequencies. Moreover, a harder skill should not be linked forwards to an easier skill. As shown in Table 2.1, $f_{\bar{A}B}$ represents the amount of learners with wrong answers of skill A and right answers of skill B. If $f_{A\bar{B}} \rangle f_{\bar{A}B}$, a skill ***A*** could be linked to a harder skill ***B***, but backward link is not permitted.

Table 2.1: Relative Skills Frequency

	A is right	***A*** is wrong
B is right	f_{AB}	$f_{\bar{A}B}$
B is wrong	$f_{A\bar{B}}$	$f_{\bar{A}\bar{B}}$

Hsu also proposed a conceptual map-based notation, called Concept Effect Relationships (CER), to model the learning effect relationships among concepts [51]. In brief, for two concepts, C_i and C_j, if C_i is the prerequisite for efficiently learning the more complex and higher level concept C_j, then a CER $C_i \rightarrow C_j$ exists. A single concept may have multiple prerequisite concepts, and can also be a prerequisite concept of multiple concepts. Thus, based upon CER, the learning guidance of necessary concepts to enhance their learning performance can be derived by analyzing the test results of students. Later, based upon statistical prediction and approach of Hsu [51], a CER Builder was proposed by Hwang [49]. Firstly, CER Builder finds the test item that most students failed to answer correctly and then collects the other test items failed to answer by the same students. Thus, CER Builder can use the information to determine the relationships among the test items. Though the CER Builder is easy to understand, only using single rule type is not enough to analyze the prerequisite relationship among concepts of test items, which may decrease the quality of concept map.

Tsai proposed a Two-Phase Fuzzy Mining and Learning Algorithm [126]. In the first phase, **L**ook Ahead **F**uzzy **M**ining Association Rule **Alg**orithm (LFMAlg) was proposed to find the embedded association rules from the historical learning records of students. In the second phase, the AQR algorithm was applied to find the misconcept map indicating the missing concepts during students learning. The obtained misconcept map as recommendation can be fed back to teachers for remedy learning of students. However, because the creating misconcept map, which is not a complete concept map of a course, only represents the missing learning concepts, its usefulness and flexibility are decreased. In addition, their approaches generate many noisy rules and only use single rule type to analyze the prerequisite relationship among learning concepts.

Chapter 3 Intelligent Learning Content Management System (ILCMS)

3.1 The Layered Model of IEEE LTSA

In order to provide learners with an adaptive learning environment, the Learning Technology System Architecture (LTSA) of IEEE LTSC [80] as a reference model identifies the critical interoperability interfaces for learning technology systems. In addition, in order to support the interoperability and scalability of distributed e-learning system, IMS Abstract Framework (AF) [55] proposes a layered model, which defines the interface definition set. Also, E-Learning Framework (ELF) [35] also proposes a layered model, each layer of which defines different functionalities according to the different requirements of an e-learning system. Therefore, based on the layered models of IMS AF and ELF, LTSA reference model can be reorganized into 4 layers: resources, common services, learning services, and application, according to the functions of its components. Figure 3.1 illustrates the layered LTSA model, where the module in higher layer will use the service provided from lower layer to offer more powerful and specific service. For example, the Delivery module in Common layer uses the resources in Resources layer to deliver to the learners.

Furthermore, based on the knowledge management concept [39], how to efficiently manage the different resources and information in an adaptive e-learning system is similar to efficiently manage diverse knowledge. Accordingly, each module in LTSA can be classified into five knowledge types according to its function, i.e., Knowledge Resources including learning resources and records, Knowledge Manager including the

Delivery, Knowledge Controller and Knowledge Acquirer including the Coach, and Knowledge Miner including the Evaluation, as shown in Figure 3.2.

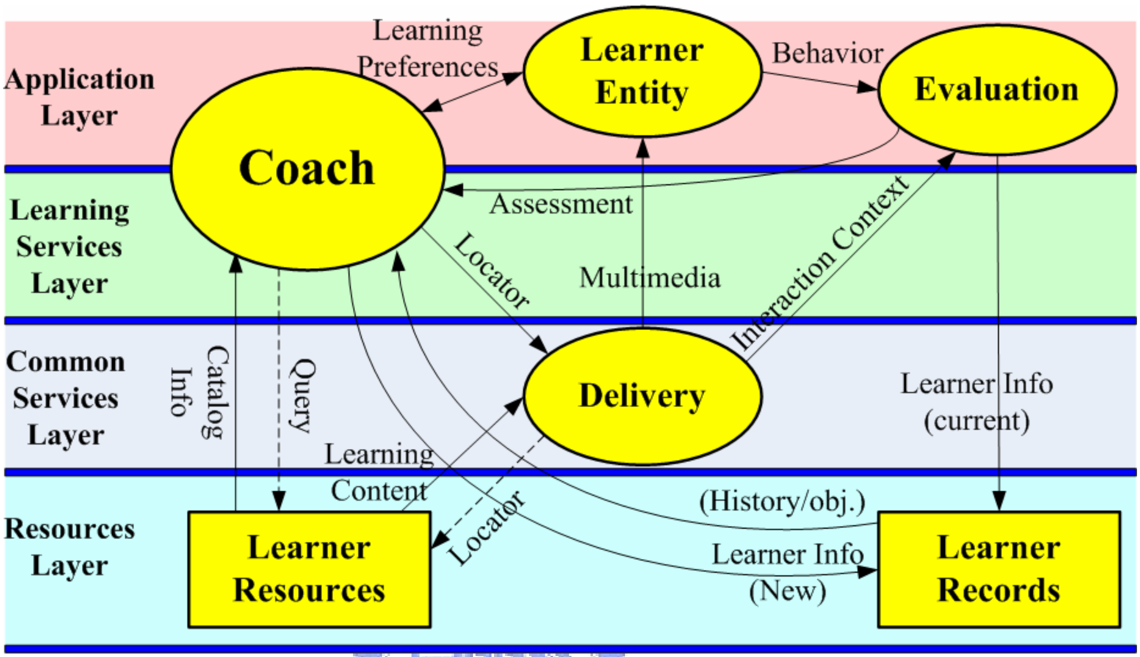


Figure 3.1: The Layered Model of IEEE LTSA

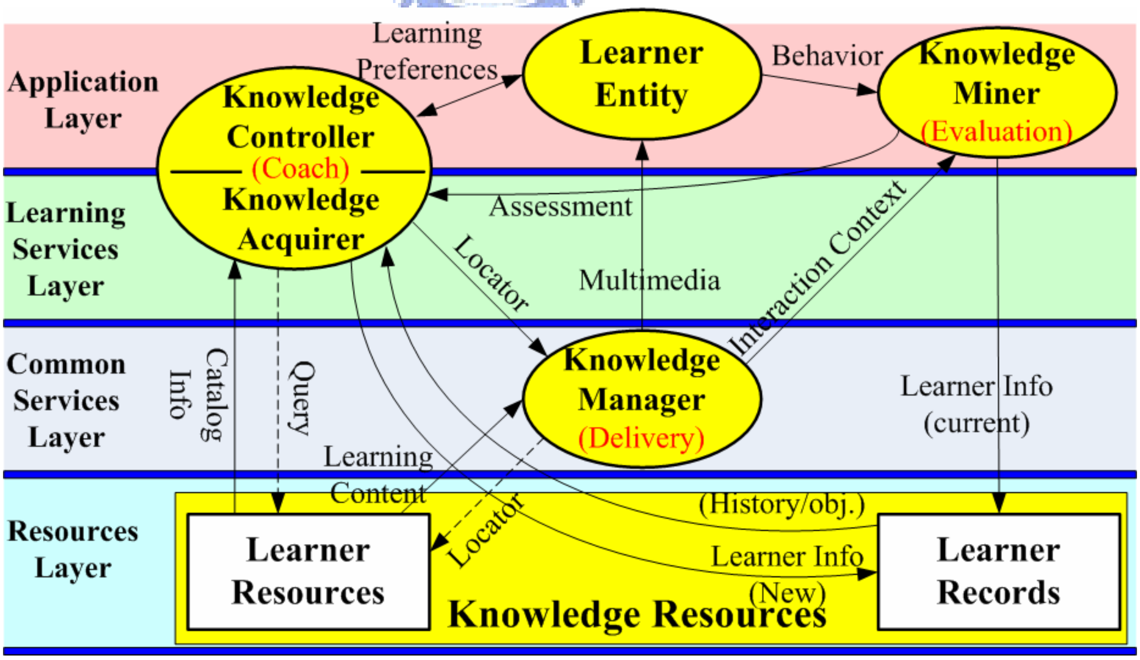


Figure 3.2: Applying Knowledge Management Concept to Layered Model of LTSA

3.2 The Architecture of ILCMS

As mentioned above, LTSA can be layered into 4 layers according to the service function of each layer and classified into 5 knowledge types based on knowledge management concept. However, because IEEE LTSA is as a reference model of building an e-learning system in support of adaptive learning, it does not clearly specify and define how to represent the learning content and activity. Therefore, in order to solve the issue of uniform data format among e-learning systems, how to define the data representation format of learning content and activity is a very important issue.

Therefore, in this dissertation, based on Knowledge Management concept [39] and layered IEEE LTSA [80], an **Intelligent Learning Content Management System (ILCMS)** is proposed to intelligently manage a large number of learning contents and offer learners an adaptive learning strategy which can be refined by means of efficient learning portfolio analysis. Figure 3.3 shows the layered architecture of ILCMS consisting of six knowledge modules in corresponding layer respectively, i.e., **1) Knowledge Representation**, which uses SCORM standard, and new proposed Instructional Activity Model (IAM) and Object Oriented Learning Activity (OOLA) model to represent and manage the learning content and activity, **2) Knowledge Resources**, which stores all related learning resources in repositories, **3) Knowledge Manager**, which efficient manages a large number of learning resources in repositories, **4) Knowledge Acquirer**, which provides teachers with useful tools to create the SCORM and OOLA compliant learning content and activity, **5) Knowledge Controller**, which intelligently delivers the desired learning contents, services, test sheet to learners according to her/his learning results and performance, and **6) Knowledge Miner**, which analyzes the learning portfolio for constructing the adaptive learning course and the learning concept map automatically.

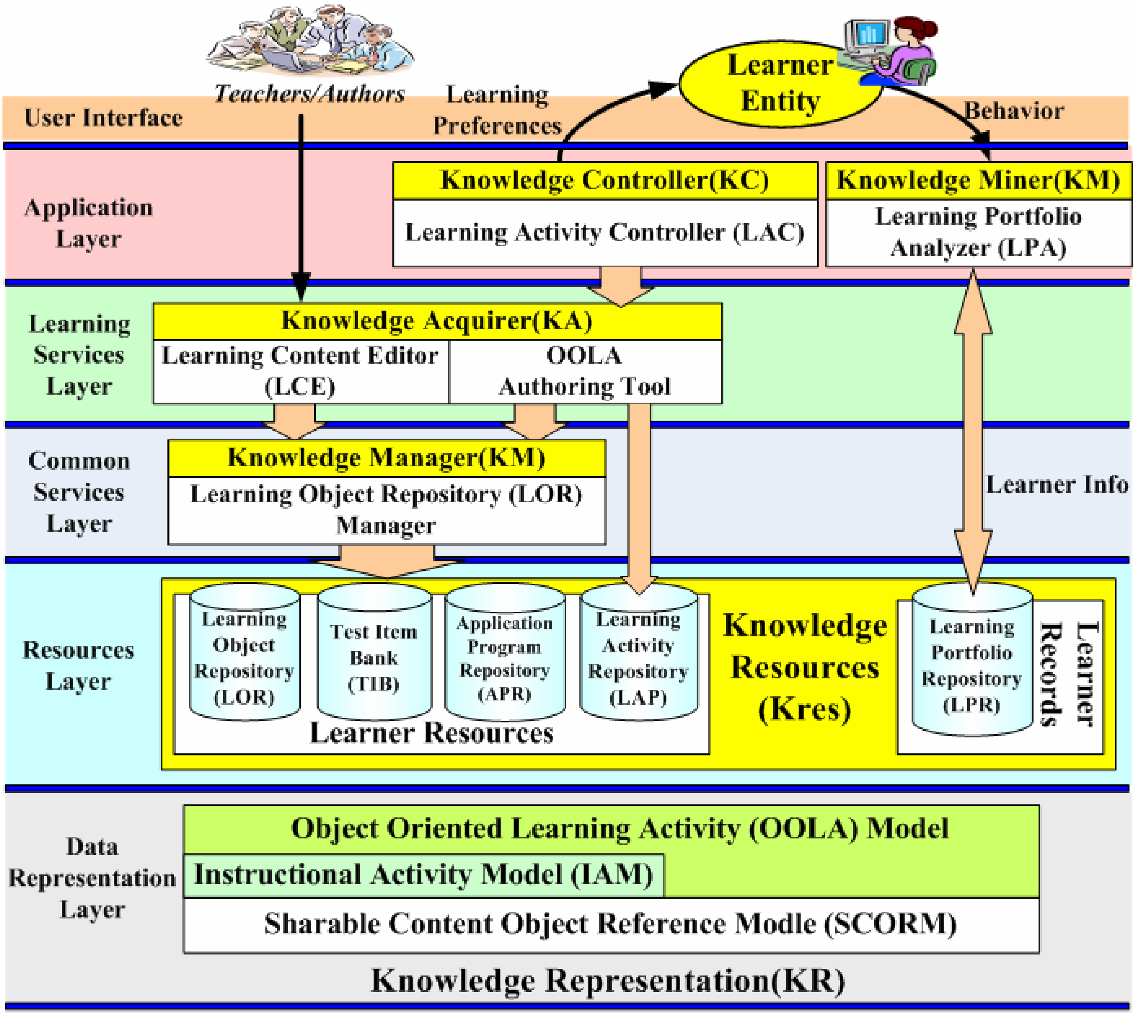


Figure 3.3: The Layered Architecture of Intelligent Learning Content Management System (ILCMS)

Each knowledge module of ILCMS can be described in details as follows:

1. **Knowledge Representation (KR):** it includes 3 data models: **SCORM**, **Instructional Activity Model (IAM)** [115], and **Object Oriented Learning Activity (OOLA)** [81], to represent the learning content and activity, respectively. As state previously, in order to share and reuse the contents among various learning systems, we use the popular SCORM standard to represent the teaching materials so that the issue of uniform content format can be solved. Moreover, in order to

efficient manage and reuse the large-scale Activity Tree (AT) with complex sequencing rules in SCORM. Therefore, we propose an **Instructional Activity Model (IAM)**, which extends and modularizes the structure of AT with inter-relation attributes by means of Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. Furthermore, based on the modularized AT of IAM and object oriented concept, we further propose a model with sequencing rule definition, called **Object Oriented Learning Activity (OOLA)**, to efficiently model a adaptive learning activity by means of three basic elements, that is, **Content**, **Interaction**, and **Assessment**. Thus, an adaptive learning activity can be easily created and offered to learners with a personalized learning contents, services, and assessment.

2. **Knowledge Resources (KRes):** it includes five types of learning resources, i.e., **Learning Activity**, **Learning Object**, **Test Item**, **Application Program**, and **Learning Portfolio**, which are stored in their respective repositories and can be managed, reused, delivered, and analyzed by the sub-module of ILCMS in higher layers.
3. **Knowledge Manager (KM):** it includes a **Learning Object Repository (LOR) Manager**, in which we analyze the content structure of SCORM and then apply clustering technique and load balancing strategies to propose a ***Level-wise Content Management Scheme*** (LCMS)[117]. LCMS can automatically analyze the SCORM compliant contents, group these related objects into a cluster, and then create the relation links among different clusters. Therefore, by means of LCMS, LOR manager can efficiently maintain, search, and retrieve the desired learning objects from the SCORM compliant LOR with a large number of learning objects.
4. **Knowledge Acquirer (KA):** it includes a **Learning Content Editor (LCE)** and an **OOLA authoring tool** [81]. The former proposes a **Content Transformation**

Scheme (CTS)[114], which can efficiently transform the traditional teaching materials, e.g., HTML and PPT file format, into SCORM compliant learning contents, and an SCORM 2004 compliant authoring tool with **Object Oriented Course Modeling (OOCM)** [117] approach based upon High Level Petri Nets (HLPN) theory [59] [60] [62] [70] [71] [73] [82] [84], which can help teachers or editors efficiently create the course with desired learning sequencing guidance of SCORM standard. These created SCORM compliant learning content will be stored in Learning Object Repository (LOR). In addition, in order to construct OOLA compliant learning activity, the latter is a user-friendly GUI authoring tool, by which teachers can efficiently edit desired learning activity with associated SCORM compliant course in LOR, test sheet in TIB, and application program (AP) in APR. AP like an interaction tool, e.g., chat room, browser, messenger, etc., offer learners to interact with other learners and teachers. These edited OOLA learning activities will be transformed into rule format and then stored in Learning Activity Repository (LAR).

5. **Knowledge Controller (KC):** includes a **Learning Activity Controller (LAC)**, which includes a **System Coordinator (SC)** and an **Inference Engine (IE)** to provide learners with personalized learning contents, exercises, and test sheets according to different learner's portfolios and teaching strategies.
6. **Knowledge Miner (KMin):** includes a **Learning Portfolio Analyzer (LPA)**, which consists of **Learning Portfolio Mining (LPM)** [118] and **Two-Phase Concept Map Construction (TP-CMC)** [110] algorithm. According to learners' characteristics, the former applies the clustering and decision tree approach to analyze the learning behavior of learners with high learning performance for constructing the adaptive learning course. The latter applies Fuzzy Set Theory and Data Mining approach to automatically construct the concept map by learners'

historical testing records. Therefore, after the learners finished the learning activities, teachers can use LPA module to analyze the learning portfolios of learners for refining their teaching strategies and contents.

After the explanation above, the relationship of five knowledge module in ILCMS are described as follows. First, LCE and OOLA authoring tool in **KA** module can offer teachers or editors to edit the new SCORM compliant learning contents or transforms existing traditional teaching materials into SCORM compliant ones, and construct an OOLA learning activity, respectively. Then, LOR Manager in **KM** module applies clustering approach and load balancing strategies to efficiently manage a large number of learning objects in LOR. When learners initiate a learning activity, the LAC in **KC** module will retrieve the appropriate learning objects in LOR, testing sheets in Testing Item Bank (TIB), or application program (AP) in APR according to the personalized learning activity in LAR for learners. As mentioned above, the learning contents, test sheet, and AP will be retrieved and triggered according to the specific learning strategy. Those strategies are created by teachers using the authoring tool in KA module. Besides, after the learners finished the learning activities, teachers can use the LPA in **KMin** module to analyze the learning portfolios of learners for refining their teaching strategies and contents.

The topics in this dissertation mentioned above will be detailedly discussed in following Sections.

Chapter 4 Knowledge Representation (KR)

In this chapter, we describe the data format used to represent the learning resources in ILCMS. In order to share and reuse the contents among various learning systems, the popular SCORM standard is used to represent the teaching materials so that the issue of uniform content format can be solved. Moreover, in order to efficient manage and reuse the large-scale Activity Tree (AT) with complex sequencing rules in SCORM. Therefore, we propose an **Instructional Activity Model (IAM)**, which extends and modularizes the structure of AT with inter-relation attributes by means of Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. Furthermore, based on the modularized AT of IAM and object oriented concept, we further propose a learning activity model with sequencing rule definition, called **Object Oriented Learning Activity (OOLA)**, to efficiently model a adaptive learning activity by means of three basic elements, that is, **Content**, **Interaction**, and **Assessment**. Thus, an adaptive learning activity can be easily created and offered to learners with a personalized learning contents, services, and assessment. The details of SCORM, IAM, and OOLA model will be described below.

4.1 Sharable Content Object Reference Model (SCORM)

In SCORM specification, content packaging scheme is proposed to package the learning objects into standard teaching materials, shown in Figure 4.1. The content packaging scheme defines a teaching materials package consisting of 4 parts, that is, **1) Metadata:** describes the characteristic or attribute of this learning content, **2) Organizations:** describe the structure of this teaching material, **3) Resources:** denote the physical file linked by each learning object within the teaching material, and **4) (Sub)**

Manifest: describes this teaching material consists of itself and another teaching material. In Figure 4.1, the organizations define the structure of whole teaching material, which consists of many organizations containing arbitrary number of tags, called *item*, to denote the corresponding chapter, section, or subsection within physical teaching material. Each item as a learning activity can be also tagged with activity metadata which can be used to easily reuse and discover within a content repository or similar system and to provide descriptive information about the activity. Hence, based upon the concept of learning object and SCORM content packaging scheme, the teaching materials can be constructed dynamically by organizing the learning objects according to the learning strategies, students' learning aptitudes, and the evaluation results. Thus, the individualized teaching materials can be offered to each student for learning, and then the teaching material can be reused, shared, recombined.

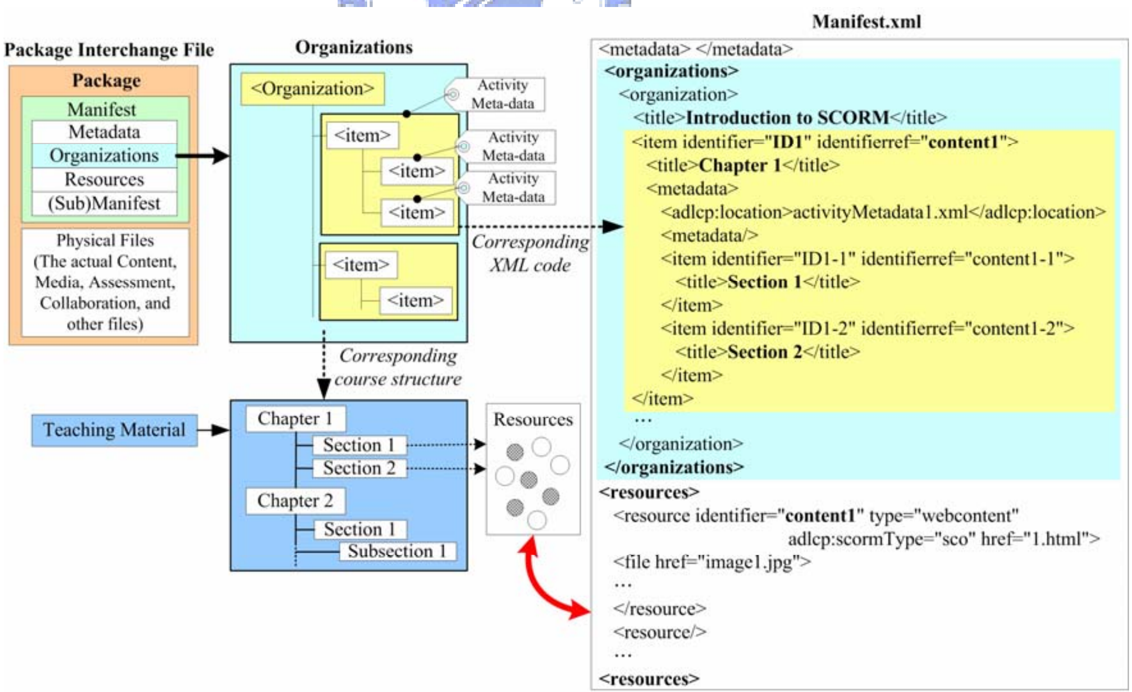


Figure 4.1: SCORM Content Packaging Scope and Corresponding Structure of Teaching Materials

4.1.1 Sequencing and Navigation (SN) of SCORM

At present, Sequencing and Navigation (SN) [109] in SCORM 1.3 (also called SCORM 2004) adopts the Simple Sequencing Specification of IMS [56] based on the concepts of learning activities, each of which may be described as an instructional event, as an event embedded in a content resource. The content in SN is organized into a hierarchical structure, namely, an activity tree (AT) as a learning map. An example of an AT is shown in Figure 4.2. Each learning activity, including one or more child activities, includes two data models: Sequencing Definition Model (SDM) including an associated set of desired sequencing behaviors of content designer and Tracking Status Model (TSM) including the information about a learner's interaction with the learning objects within associated activities. SN uses information in SDM and TSM to control the sequencing, selection, and delivery of activities to the learner.

The sequencing behaviors describe how the activity or how the children of the activity are used to create the desired learning experience. SN places no restrictions on the structure, organization, or instruction of the activity tree. The tree and the associated sequencing definitions may be statically or dynamically created. Therefore, how to create, represent, and maintain the activity tree and associated sequencing definition, which is not specified, is an important issue. SN enables us to share not only learning contents but also intended learning experiences. It also provides a set of widely used sequencing methods so that the teacher could do sequencing efficiently. Accordingly, in this dissertation, SCORM standard is used to represent the learning contents associated with related learning object and sequencing rules.

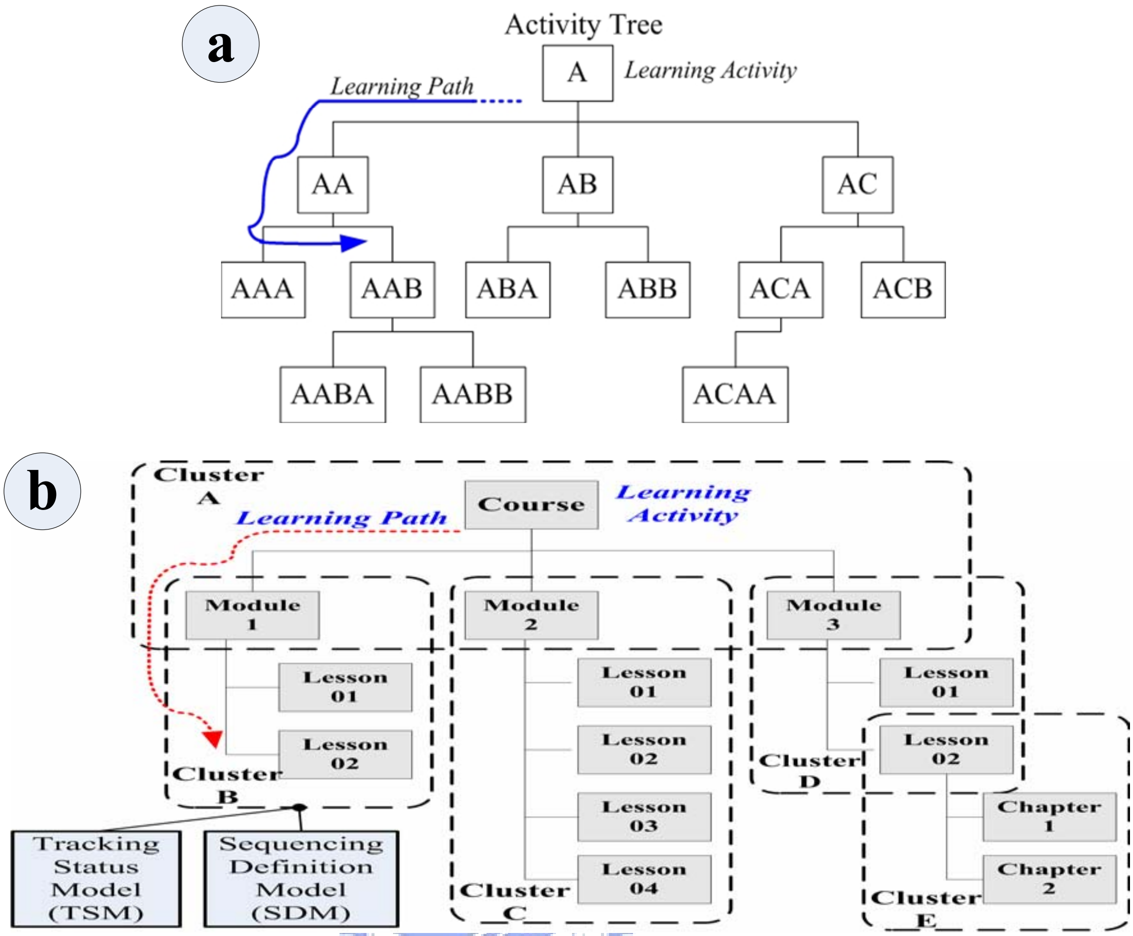


Figure 4.2: An Example of Activity Tree (a) with Clusters (b)

4.2 Instructional Activity Model (IAM)

As mentioned above, in addition to describe the learning contents associated with learning objects, SCORM standard also defines a hierarchical structure, namely, an Activity Tree (AT), used to sequence the delivery of learning content to the learner. By defining the sequencing behavior rules within an AT, we can develop an intelligent approach to (semi-)automatic course and exercise sequencing. Therefore, how to create, represent, and maintain the Activity Tree and associated sequencing definition is our concern. For a large-scale learning activity, the Activity Tree will become too complex to be managed and reused. Besides, it is hard to reuse and integrate ATs without knowing the inter-relations among ATs. This implies that the scalability and flexibility of an adaptive learning system will be limited. Moreover, for modern personalized learning, many researches have used Pedagogical Theory [19] [40] [87] [122] to enhance the evaluation of the personal learning characteristic.

Hence, in this dissertation, we first define the interrelation attributes of an AT, e.g., capability, weight, etc. Then, we extend and modularize the structure of AT by means of Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. As shown on the right side of Figure 4.3, a large AT is divided into three small AT nodes with interrelation attributes. Therefore, by means of the interrelation attributes, the small AT nodes can be integrated and further connected with other AT nodes; e.g., AT_1 connects AT_4 and AT_5. Thus, we propose a novel model, the Instructional Activity Model (IAM) [115], which is composed of related Activity Tree nodes. Based upon Pedagogical Theory, each AT node in IAM is modularized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose an AT Selection algorithm with a pedagogical strategy used to traverse IAM in order to generate dynamic learning content for the learner. In this section, we describe the Instructional Activity Model, including its properties, and the AT Selection algorithm.

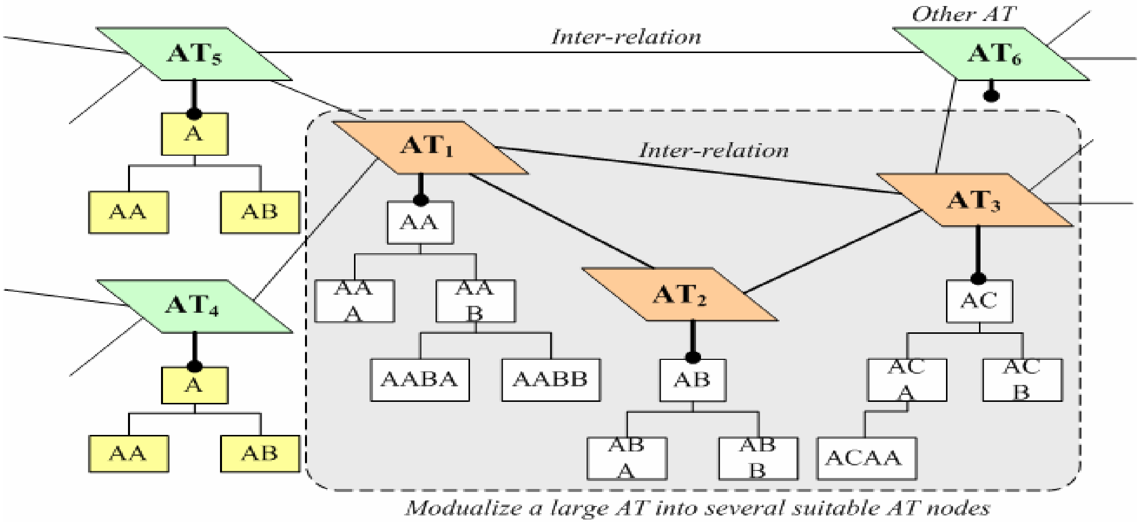


Figure 4.3: The Concept of Modularizing an AT

4.2.1 Instructional Activity Model

In Sequencing and Navigation (SN), we can create an AT on the fly. As mentioned above, in a large AT, its organization and sequencing rules are hard to manage and reuse. However, a large number of ATs will also make the management of AT nodes and rules complicated. Therefore, to strengthen the scalability and flexibility of AT, we must define a suitable unit of an AT. According to Bloom's Mastery Theory [10], a suitable unit of learning content is a chapter or a section for learning. Thus, in IAM, we define the unit of an AT as a chapter or a section.

Assume there are *n* ATs. We define an AT set as $\mathbf{AT_{set}} = \{AT_1, AT_2, \ldots, AT_n\}$. According to the formulation of Gagne [42], "*A capability is a knowledge unit stored in a person's long term memory that allows him/her to succeed in the realization of physical, intellectual or professional activity*." Suppose there are *m* capabilities; we can obtain $\mathbf{C_{set}} = \{c_1, c_2, \ldots, c_m\}$. Before learning an AT, students are supposed to possess some capabilities, called ***Prerequisites***. Similarly, after learning an activity tree, students can acquire further capabilities, called ***Contributions***. Every ***prerequisite*** or ***contribution*** has its own weight representing the significance of learning capabilities before and after learning. Therefore, in IAM, the $\mathbf{C_{set}}$ can be regarded as the union of all ***prerequisites*** and ***contributions***, and an AT, thus, has several capabilities.

A learning activity or a course is composed of several ATs with input/output capabilities. The student learns a suitable AT and gains further capabilities, which enable the student to learn another advanced AT. This learning process is repeated until the student has finished all the learning objectives predefined by teachers. Then, every student will have an individual value of C_{set}. Figure 4.4 shows a diagram of IAM.

In Table 4.1, we define the related attributes of interrelations, measure functions, AT selecting criteria, etc. in IAM as shown in Figure 4.4

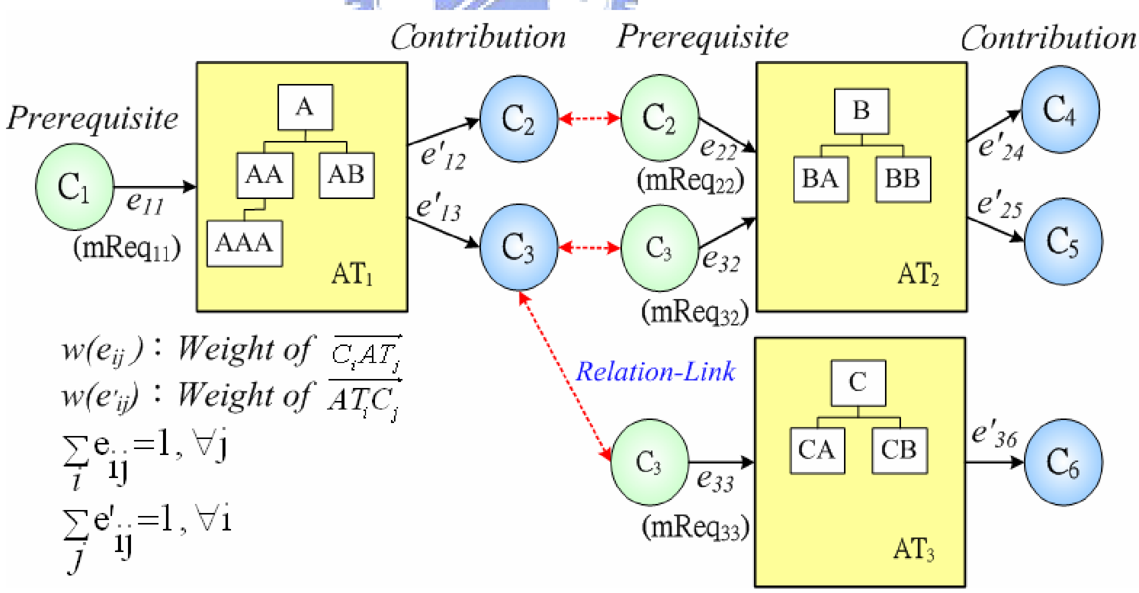


Figure 4.4: The Diagram of IAM

Table 4.1: The Definitions of Related Symbols in IAM

Symbols	Description
e_{ij}	The edge from c_i to AT_j, called "*prerequisite edge*", means that before learning AT_j, the student is supposed to possess this ability c_i.
e'_{ij}	The edge from AT_i to c_j, called "*contribution edge*", means that after learning AT_i, the student will gain the ability c_j.
$w(e_{ij})$	The weight of e_{ij} denotes the significance of c_i before learning AT_j where the sum of $w(e_{ij})$ of an AT is 1, i.e., $\sum_i w(e_{ij}) = 1, \forall j$.
$w(e'_{ij})$	The weight of e'_{ij} denotes the significance of c_j after learning AT_i where the sum of $w(e'_{ij})$ of an AT is 1, i.e., $\sum_j w(e'_{ij}) = 1, \forall i$.
$mReq_{ij}$	The minimum requirement of c_i for learning AT_j is used to determine whether the student is qualified to learn AT_j or not.
$grade(e'_{ij})$	The learning grade after learning AT_i.
$val(c_m)$	The evaluation function of a capability, i.e., $val(c_m) = \dfrac{\sum_j w(e'_{jm}) \times grade(e'_{jm})}{\sum_j w(e'_{jm})}$

Table 4.1: (Cont'd) The Definitions of Related Symbols in IAM

The Related Measure Functions of AT	
Acquired Capability (AC)	It records student's learning results. $AC=\cup(c_i, val(c_i))$,
Course Objectives (CO)	It records student's learning objectives. $CO=\cup(c_i)$.
Potential Capability List (PCL)	Each AT has a PCL recording all the contribution capabilities which can be reached from this AT via edges in IAM. It can be formulated as $PCL_{ATk}=\cup(c_i)$, where c_i can be reached from AT_k by connecting edges, e.g., in Figure 7.2 the PCL_{AT1} equals $\{c_2, c_3, c_4, c_5, c_6\}$.
Student's Grade Prediction (SGP)	SGP denotes the performance prediction of the specific student related to the AT, i.e., $SGP_k=\boldsymbol{\Sigma}_i (val(c_i)\times w(e_{ik}))$.
Normalized Objective Weight (NOW)	NOW denotes the relativity between an AT and the student's CO. Higher objective weight implies better learning efficiency. Empirically, selecting function tends to select the AT with higher SGP and higher NOW for students. $NOW = \dfrac{\text{the number of } c_i (c_i \in PCL_{ATj} \,\&\, c_i \in CO)}{\text{the number of } c_i (c_i \in PCL_{ATj})}$
Chosen Factor (CF)	CF, a linear combination of selecting criteria, is used to select a suitable AT for learner. For example, for AT_i, $CF_i=\alpha NOW_i + \beta SGP_i$, where $\alpha+\beta=1$, $0\le \alpha, \beta \le 1$.

In brief, the Instructional activity model (IAM), a graphical representation of a learning activity or course, contains a set of **AT**s; **Capabilities**, including *prerequisites* and *contributions*; a set of **Relations Edges**, including eij with mReqij and e'_{ij}; and a set of **Measure Functions**. Assume IAM has *n* ATs and *m* capabilities. Then, it can be formulated as a quadruple, **IAM** = ($\mathbf{AT_{set}}$, $\mathbf{C_{set}}$, $\mathbf{E_{set}}$, $\mathbf{E'_{set}}$),

where

$\mathbf{AT_{set}} = \{AT_1, AT_2, \ldots, AT_n\}$.

●$\mathbf{C_{set}} = \{c_1, c_2, \ldots, c_m\}$.

●$\mathbf{E_{set}} = \cup(E_j)$, where $\mathbf{E_j} = \cup_i (e_{ij}, mReq_{ij})$, $e_{ij} \in AT_j$.

●$\mathbf{E_{set}}$ is the set of all prerequisite edges with minimum requirement value in an IAM.

●$\mathbf{E'_{set}} = \cup(E'_j)$, where $\mathbf{E'_j} = \cup_j (e'_{jk})$, $e'_{jk} \in AT_j$.

●$\mathbf{E'_{set}}$ is the set of all contribution edges in an IAM.

4.2.2 Basic Functionalities

Based upon the structure of IAM described above, we can develop several approaches to provide students with a learning environment for a dynamic and adaptive course. The learning process can be simply considered as the sequencing of activity trees in IAM in order to enable students to satisfy the learning objectives. The flowchart and algorithm of AT Selection is shown in Figure 4.5 and Algorithm 4.1, respectively.

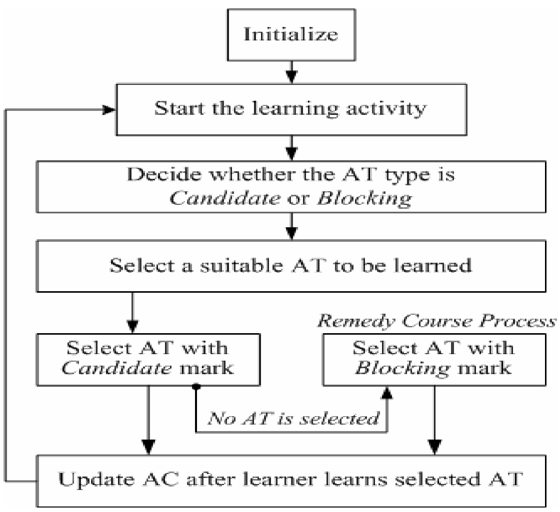


Figure 4.5: The Flowchart of AT Selecting Process

Here, we will explain the AT Selection algorithm of IAM. First, we initialize the learning status by loading AC and CO, evaluate the PCL_{AT} of every AT **(Setp1)**, and then enter the loop of the learning activity **(Step2)**. During the AT selection process, we mark each AT with *Candidate* or *Blocking* after comparing the $mReq(e_{ij})$ with $val(c_i)$ **(Step2.1)**. *Candidate* indicates that this AT will be selected later, and *Blocking* indicates the opposite. Before delivering AT to the learner, we have to execute the selection process to choose a suitable AT. In general, we only use the **CF** value to choose one suitable AT **(Step 2.2.2)**. However, to meet specific needs, e.g., to apply Pedagogical Theory, we can define other selection criteria and a strategy in the *extended selection scheme*, which will be described later in Section 4.2.3 **(Step 2.2.1)**. After completing the AT selection process, we choose a suitable AT marked *candidate* and deliver it to the learner **(Step 2.2)**. However, if no AT marked *candidate* exists, the AT selection process proceeds to the **Remedy Course Process (Step 2.2.3-Step 2.2.7)**. In the Remedy Course Process, we select an AT with the largest value of $c_m \in CO$ **(Step 2.2.3-Step 2.2.4)** and then find a c_i with the *smallest*, *largest*, or *medium* value of $(mReq(e_{ij}) - val(c_i))$, according to the type of ***SelectingPolicy*** **(Step 2.2.5)**. In this algorithm, we use three policies to select different capabilities for adaptive learning. The policy "*Easiest First*" tends to select a c_i in which the learner has earned a high grade, but the policy "*Hardest First*" does the opposite. After selecting a c_i, we can decide which AT connected with c_i to deliver to the learner by computing **MAX**$((mReq(e_{ij}) - grade(e'k_i)) \times w(e'k_i))$, which implies that the progress of the learner is the largest **(Step 2.2.6)**. Figure 4.6 shows in detail the Remedy Course Process. Finally, when the learner has finished and satisfied all the course objectives (CO), the AT selection process stops.

Algorithm 4.1: AT Selection Algorithm

Input: IAM, AC and **CO** of learner, and ***SelectingPolicy*** = {*Easiest First*, *Medium First*, *Hardest First*}.

Output: the new **AC** after learner has finished learning activity.

Step1: Evaluate the PCL_{AT} of every AT in IAM.

Step2: while($CO \not\subset AC$) *//start the learning activity*

// decide whether the type of AT is Candidate or Blocking state

2.1: for each c_i with e_{ij} in AC

{ if ($mReq(e_{ij}) > val(c_i)$)

then mark the AT_j with *Blocking*

else if (AT_j has not been learned yet)

then (compute CF_j)and (mark the AT_j with *Candidate*) **}**

//select a suitable AT to be learned

2.2: if (∃AT with *Candidate* mark) *// select the AT with Candidate mark*

then

2.2.1: if ∃ *extended selecting scheme* of AT **then** do it. *// for specific needs*

2.2.2: Select an AT with the highest CF and deliver it to the learner.

else if (∃AT with *Blocking* mark)

then *//go to Remedy Course Process & select a suitable AT*

2.2.3: for each AT_j with *Blocking* mark

{Count the amount of $c_m \in CO$ which is connected by e'_{jm}.}

2.2.4: Select the AT_j with the largest amount of $c_m \in CO$.

2.2.5: for all c_i with e_{ij}

{ if *SelectingPolicy* = "*Easiest First*", "*Medium First*" or "*Hardest First*"

then Find the c_i with the *smallest*, *medium*, *largest* value of ($mReq(e_{ij})-val(c_i)$), respectively.**}**

2.2.6: for all $e'_{ki} \in E_i$ in c_i,

Select the AT_k with **MAX**(($mReq(e_{ij})$–$grade(e'_{ki})$)$\times w(e'_{ki})$).

2.2.7: Clear the mark of AT_j and deliver the AT_k to the learner.

2.3: if learner passes the selected AT

then mark this AT with *Learned*.

2.4: update AC after the learner learns selected AT.

Step3: return new AC.

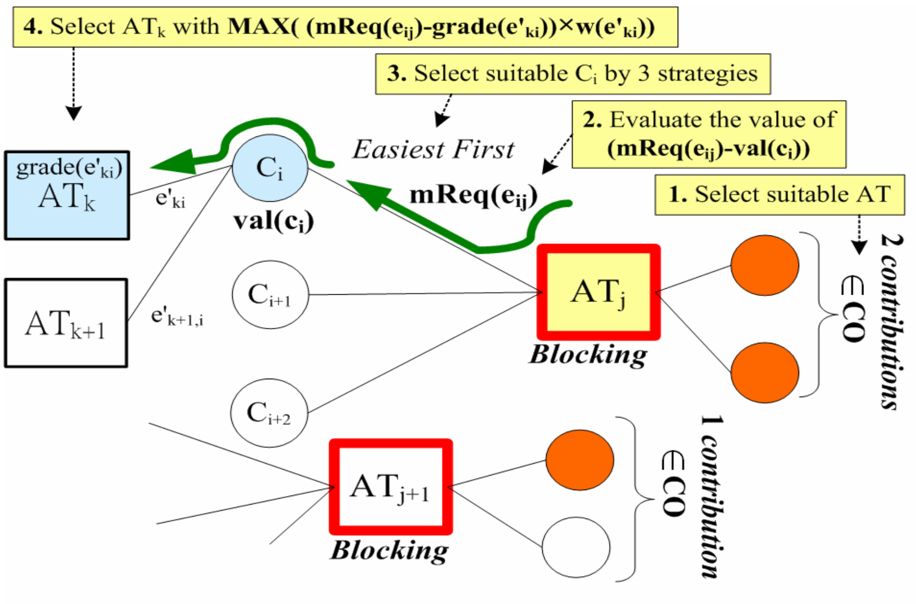


Figure 4.6: The Diagram of Remedy Course Process

Example 4.1:

This IAM in Figure 4.7 can be represented as follows:

IAM = ({AT_1, AT_2 AT_3, AT_4, AT_5,}, {c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9}, {(e_{11},0.8), (e_{22},0.7) , (e_{23},0.8), (e_{33},0.8) , (e_{44},0.8) , (e_{55},0.8) , (e_{65},0.6)}, {e'_{14}, e'_{15}, e'_{25}, e'_{36}, e'_{47}, e'_{48}, e'_{58}, e'_{59}}).

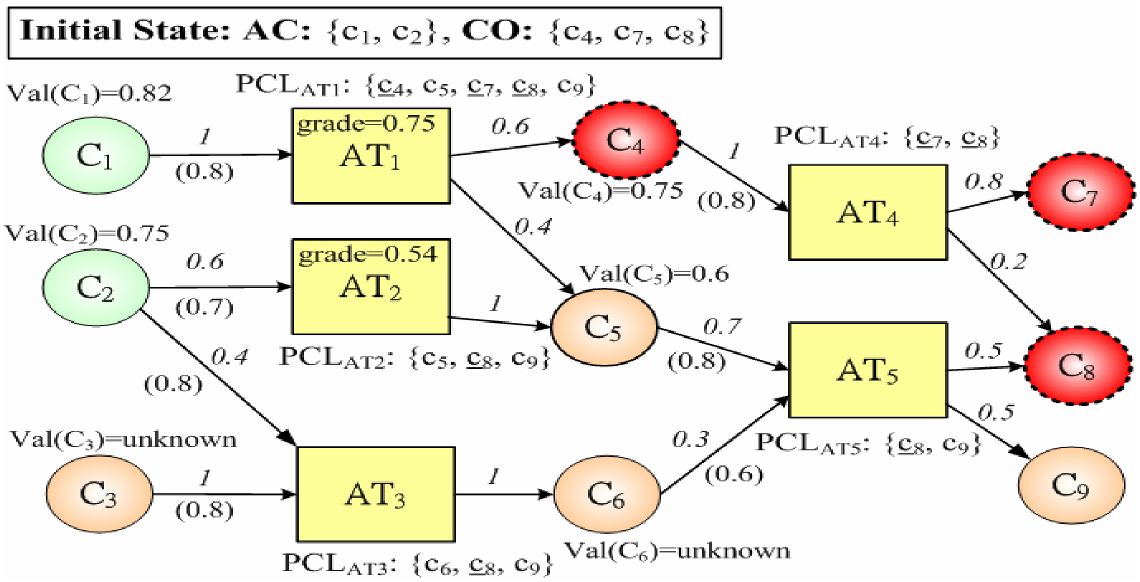


Figure 4.7: The Example of IAM

Case 1: We assume that **AC**={(c_1, 0.82), (c_2, 0.75) } and **CO**={c_4, c_7, c_8}. *Note that the value in parenthesis is the val(c_i).*

The PCL_{AT} has been evaluated as shown in Figure 4.7. After the first iteration of the **While** loop of Algorithm 4.1, we can get results as shown in Table 4.2. Thus, AT_1 will be delivered to the learner because it has the highest CF value.

Case 2: we assume that AC={ (c_1, 0.82), (c_2, 0.75), (c_4, 0.75), (c_5, 0.6), (c_6, unknown) }, CO={c_4, c_7, c_8}, and Blocking AT={AT_3, AT_4, AT_5}. The AT selecting process has moved into **Remedy Course Process**.

Before **Step 2.2.5**, because AT_5 has one $c_m \in CO$, AT_5 is selected. If the **Selection-Policy** is "*Easiest First*," the c_5 with the smallest value, 0.2, of (mReq(e_{55}) − val(c_5)) is selected. Then, by computing (mReq(e_{55}) − grade(e'_{15}))×w(e'_{15})) and (mReq(e_{25}) − grade(e'_{25}))×w(e'_{25})), we can decide to deliver the AT_2 with a value of 0.26 to the learner.

Table 4.2: The Related Values of AT_1 and AT_2

	SGP	NOW	CF
AT_1	SGP_1= val(c_1) ×w(e_{11}) =0.82×1 =0.82	$NOW_1 = \frac{\text{the number of } \{c_4, c_7, c_8\}}{\text{the number of } \{c_4, c_5, c_7, c_8, c_9\}} = \frac{3}{5} = 0.6$	$CF_1 = \alpha \times SGP_1 + \beta \times NOW_1 = 0.5 \times 0.82 + 0.5 \times 0.6 = 0.71$
AT_2	SGP_2=0.45	NOW_2=0.33	CF_2=0.39

4.2.3. Applying Pedagogical Theories in IAM

As mentioned above, the Instructional Activity Model (IAM), which is composed of related AT nodes with inter-relations and specific attributes, can be easily managed, reused, and integrated. Our proposed AT Selection Algorithm can then generate the dynamic learning content for the learner by traversing the IAM. In addition, due to strengthened the scalability and flexibility of IAM, appropriate pedagogical theories can be selected and applied to provide personalized learning guidance according to extension schemes for specific needs. Therefore, in this section, we will show how well-known pedagogical theories can be applied in IAM by means of extension schemes.

Extension Scheme of IAM:

We can consider three aspects of pedagogical theories: **1.** the **Capability Taxonomy**, **2.** the **Learning Style**, and **3.** the **Organization of Teaching Material**. We can describe these three aspects as follows.

- **Capability Taxonomy:** By learning different Learning content, the learner will acquire different knowledge or capabilities. Thus, Gagne [40] considered that the learning outcomes of learners can be classified into five types: **Verbal Information**, **Intellectual Skills**, **Cognitive Strategies**, **Motor Skills**, and **Attitude**. Accordingly, we can categorize the learning capabilities in IAM into five types and define each c_i in $C_{set} = \{c_1, c_2, \ldots, c_m\}$ as having five dimensions: $\langle vc_i, ic_i, cc_i, mc_i, ac_i \rangle$, where vc_i denotes **v**erbal **c**apability, ic_i denotes **i**ntellectual **c**apability, cc_i denotes **c**ognitive **c**apability, mc_i denotes **m**otor **c**apability, and ac_i denotes **a**ttitude **c**apability.
- **Learning Style:** The learner's learning style is the way s/he prefers to learn. Therefore, learners have individual learning preferences during learning activities designed for specific instructional approaches or teaching materials. Many articles [26] [72] [107] [111] [120] [122] have proved that learners can achieve excellent learning performance if we can give them instruction and teaching materials according to their individual learning styles. Sternberg [102] also collected many taxonomies of learning style based upon different criteria. Thus, we apply three features of learning styles, **V**isual, **A**uditory, and **K**inesthetic, in IAM to generate

adaptive learning guidance. To provide a learner with suitable learning contents, we have to define not only the learning style of the learner, but also the learning content of AT. Therefore, we need to select a suitable AT whose learning style is similar to that of the learner. Moreover, we can use existing questionnaires [50][102] to extract the values of individual learning styles of learners.

- **Organization of Teaching Material:** It is essential to organize suitable teaching materials for students. According to Bassing [7], we can categorize the organization of teaching materials into three types: **(1) Logical Organization**, where the teaching materials are ordered in a systematical fashion as traditional teaching strategies, e.g., teaching the mathematics from basic to advanced concept in a fixed order; **(2) Psychological Organization**, where emphasis is placed on the student's own interest, ability, and needs; and **(3) Eclectic Organization**, which takes both Logical Organization and Psychological Organization into consideration. Therefore, in IAM, the learning guidance and selected AT have to be based on the concepts of Logical Organization and Psychological Organization, respectively. Table 4.3 shows the related symbol definitions used when applying Pedagogical Theory in IAM.

Table 4.3: The Symbol Definitions of Pedagogical Theory in IAM

Symbols	Description
LgOrg$_i$	This denotes the **Logical Organization** of AT_i. The value of $LgOrg_i$ is mapped to the difficulty of AT_i.
LnSty$_i$	This denotes the value of **Learning Style**, including **V**isual, **A**uditory, and **K**inesthetic in AT_i. The LnSty is represented as a vector, i.e., <$\mathbf{V}_{ATi}$, $\mathbf{A}_{ATi}$, $\mathbf{K}_{ATi}$>, where the value is between 0 and 1.
SLS	This denotes the **Student Learning Style** (SLS) for representing the learning style of the student. **SLS** is represented as a vector like **LnSty$_i$**, i.e., <$\mathbf{V}_s$, $\mathbf{A}_s$, $\mathbf{K}_s$>, where the value is between 0 and 1.

Based upon the symbols shown in Table 4.3, we can define the **S**imilarity **F**actor, **SF**, and redefine the **C**hosen **F**actor, **CF**, for AT_i as follows:

- SF_i = SLS・$LnSty_i$, where the symbol "•" represents the dot product.
- $CF_i = \alpha NOW_i + \beta SGP_i + \gamma LgOrg_i$, where $\alpha+\beta+\gamma=1$.

The **SF** is used to compute the similarity of the learning style between the learner and ATs. Thus, we can filter out ATs with low SF values and then select the AT with the highest CF value. Although we have defined the selection formula and strategy according to Pedagogical Theory, teachers also can redefine them by themselves.

AT Selection Process Using Pedagogical Theories:

Therefore, in the **AT Selection Algorithm**, we can compute CF and SF to acquire the psychological organization and logical organization characteristics of every AT **(Step 2.1)**. The SF, which is computed as the dot product of the student's learning style vector (SLS) and the AT's learning style vector (LnSty), can denote the similarity of the learning style between the AT and Learner. Thus, using the value of SF, we can get a suitable AT form IAM **(Step 2.2.1)**. Finally, the CF can be used to determine the most suitable AT for the learner **(Step 2.2.2)**.

Example 4.2: Learning in IAM using pedagogical theories

We present a simple example of learning in IAM using pedagogical theories. First, we define IAM and the related attributes of each AT, and then we demonstrate the process of the AT Selection Algorithm for a specific student. An example of IAM is shown in Figure 4.8.

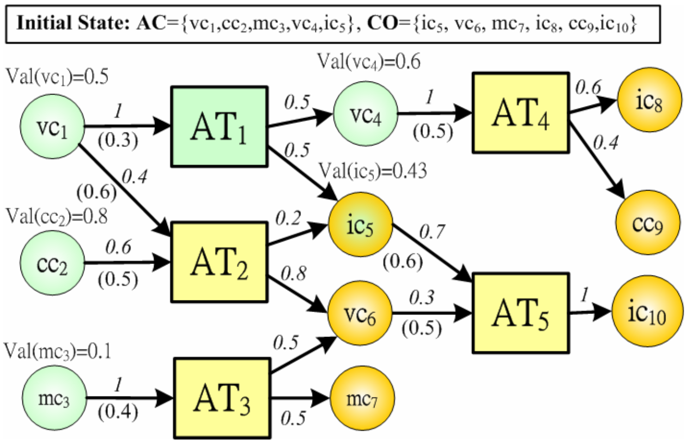


Figure 4.8: An Example of IAM with Pedagogical Theories.

IAM in Figure 4.8 is represented as follows:

IAM=({AT_1, AT_2, AT_3, AT_4, AT_5}, {vc_1, cc_2, mc_3, vc_4, ic_5, vc_6, mc_7, ic_8, cc_9, ic_{10}}, {(e_{11},0.3), (e_{12},0.6), (e_{22},0.5), (e_{33},0.4), (e_{44},0.5), (e_{55},0.6), (e_{65},0.5)}, {e'_{14}, e'_{15}, e'_{25}, e'_{26}, e'_{36}, e'_{37}, e'_{48}, e'_{49}, $e'_{5,10}$})

Table 4.4: Learning style and logical organization of each AT.

	AT_1	AT_2	AT_3	AT_4	AT_5
LnSty	<0.8, 0.1, 0.1>	<0.1, 0.8, 0.1>	<0.6, 0.1, 0.3>	<0.2, 0.1, 0.7>	<0.1, 0.2, 0.7>
LgOrg	0.3	0.3	0.5	0.3	0.7

The Learning Style and Logical Organization used in the AT Selection Algorithm are shown in Table 4.4. Because the value of LgOrg is mapped to the difficulty of AT, the difficulty of the metadata in SCORM can be used to define the value range, e.g., {*Very Easy*, *Easy*, *Medium*, *Difficult*, *Very Difficult*} corresponding to {0.1, 0.3, 0.5, 0.7, 0.9}. Suppose there is a learner who is learning in this IAM; her/his personal information is as follows:

●**AC** = {(vc_1, 0.5), (cc_2, 0.8), (mc_3, 0.1), (vc_4, 0.6), (ic_5, 0.43)},

●**SLS**= <0.1, 0.2, 0.7>,

●**CO** = {ic_5, vc_6, mc_7, ic_8, cc_9, ic_{10}}.

Since s/he has learned AT_1, the AT Selection Algorithm will choose the next AT for her/his learning. CF_i and SF_i are defined as follows:

●$CF_i=0.25\times NOW_i+0.25\times SGP_i+0.5\times LgOrg_i$,

●$SF_i= SLS \cdot LnSty_i$.

The related results obtained by the AT Selection algorithm are shown in Table 4.5.

Table 4.5: Selecting Criteria for Each Activity Tree.

	AT_2	**AT_3**	**AT_4**
PCL	{ic_5,vc_6,ic_{10}}	{ vc_6,mc_7,ic_{10}}	{ic_8, cc_9}
NOW	1	1	1
SGP	0.5×0.4+0.8×0.6=0.68	0.1×1=0.1	0.6×1=0.6
LgOrg	0.3	0.5	0.3
SF_i	0.24	0.29	0.53
CF_i	0.57	0.525	0.55

Then, we can use the following selection strategy: for smart students, select the AT with the highest CF_i value; for other students, select the AT with the highest SF_i value. With this strategy, we select AT_2 for smart students, and AT_4 for other students. In addition, we can revise CF_i and SF_i for specific purposes. For example, some teachers believe that learning style of a student is related to student's grade, and they can modify CF_i and SF_i as $CF_i = 0.5 \times NOW_i + 0.5 \times LgOrg_i$, $SF_i = 0.5 \times SGP_i + 0.5 \times (SLS \cdot LnSty_i)$. If the selection strategy remains the same, we will provide AT_3 for smart students and AT_4 for other students.

Evaluating of the Expressive Power of IAM:

We have shown that it is possible to apply pedagogical theories in IAM for specific need. How many pedagogical theories can be applied in IAM? In this section, we will evaluate that how many different structures IAM can support to meet pedagogical needs. Educational researchers have proposed various types of course structures to facilitate learning. Posner [90] proposed three types of structures including discrete structure, linear structure, and hierarchical structure. Bruner [12] proposed the concept of a spiral curriculum. Efland [34] also proposed the lattice curriculum. Each structure satisfies certain kinds of pedagogical needs. IAM can be applied to these course structures, as shown in Figures 4.9 and 4.10.

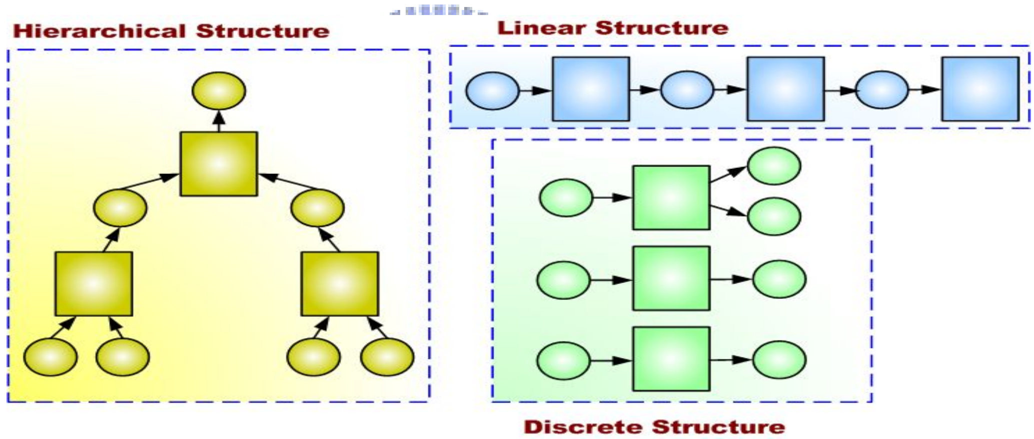


Figure 4.9: IAM Mapping to Discrete Structure, Linear Structure, and Hierarchical structure

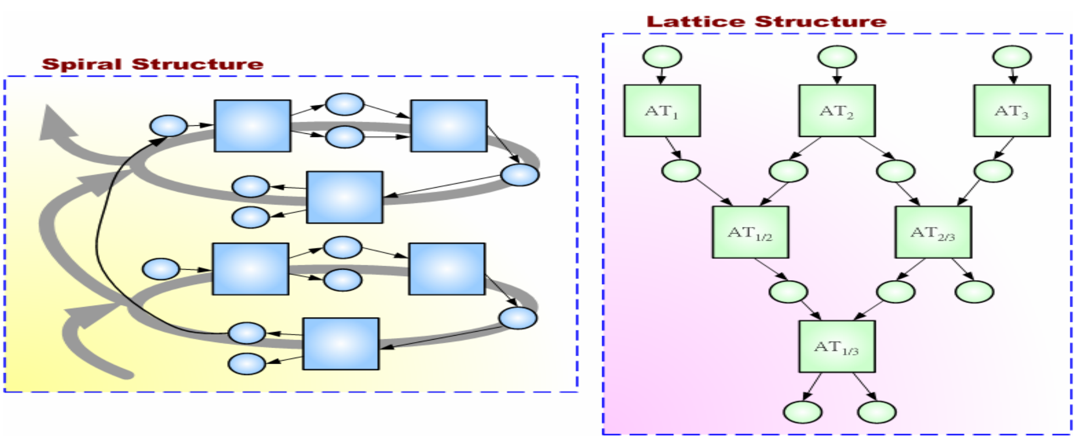


Figure 4.10: IAM Mapping to Spiral Curriculum and Lattice Curriculum.

4.2.4 The Construction of IAM

As mentioned in previous sections, based upon the OO Methodology and SCORM standard, we have proposed an Instruction Activity Model (IAM) which is composed of related AT components with inter-relations and specific attributes designed to meet pedagogical needs. However, for teachers and authors, how to apply IAM in real learning environments is also an important issue. Therefore, in this section, we propose a systematic approach to fast and easily construct IAM using traditional course resources. First, the teacher has to create the *Content-Contribution Relationship Table* denoting the potential concept which will be acquired by learning the learning content. For example, assume that a course, *Introduction to Computers*, includes three chapters as shown in Table 4.6. According to the content of Chapter A, the teacher can write down its possible contributions, including the related $w(e'_{ij})$ and difficulty level; e.g., $A_1(0.5, 1)$ indicates that the contribution, called *Hardware*, has $w(e'_{ij}) = 0.5$ and *difficulty level* = 1. Then, we use the concept of the Adjacency Matrix to create the Weight Matrix of *Contribution* as shown in Tables 4.7 and 4.8. Thus, assuming that there is an m . n Weight Matrix (M), the weight of m_{ij} in M denotes the significance of c_i before learning c_j. Hence, the teacher can write down the value of m_{ij} to define the related weight between *contribution* c_i and c_j using the following formula:

$$m_{ij} = \begin{cases} x, & \text{if Difficulty } (c_i) \leqq \text{ Difficulty } (c_j) \text{, where } 0 \leqq x \leqq 1 \\ 0, & \textit{otherwise} \end{cases}$$

For example, in Table 4.7, $A_1(1)$ indicates that the *Contribution* A_1 has difficulty level = 1. The m_{11} between A_1 and B_1 can be written as 0.3 by teachers because the Difficulty(A_1) $\leqq$ Difficulty(B_1). After finishing the Weight Matrix, the teacher can compute the value of $w(e_{ij})$ of every *contribution* using the following equation (*the* *equation will normalize* $w(e_{ij})$):

$$w(e_{ij}) = \frac{\sum_{1 \le j \le n} m_{ij}}{\sum_{1 \le i \le m} \sum_{1 \le j \le n} m_{ij}}$$

Table 4.6: The Content-Contribution Relationship Table of Course

	Contributions (*w*(*e'*$_{ij}$), *difficult level*)		
	1	2	3
Chapter A: ***Introduction***	A_1 (0.5, 1) *(Hardware)*	A_2 (0.3, 1) *(Software)*	A_3 (0.2, 1) *(Application)*
Chapter B: ***Hardware***	B_1 (0.2, 3) *(CPU)*	B_2 (0.4, 3) *(Main Memory)*	B_3 (0.4, 3) *(Auxiliary Memory)*
Chapter C: ***Software System***	C_1 (0.5, 4) *(System Software)*	C_2 (0.5, 2) *(Application Software)*	

Table 4.7: The weight matrix of contribution B

	$B_1(3)$	$B_2(3)$	$B_3(3)$	$w(e_{ij})$ of B
$A_1(1)$	0.3	0.3	0	0.67
$A_2(1)$	0	0	0.3	0.33
$A_3(1)$	0	0	0	0
$C_1(4)$	0	0	0	0
$C_2(2)$	0	0	0	0

Table 4.8: The weight matrix of contribution C

	$C_1(4)$	$C_2(2)$	$w(e_{ij})$ of C
$A_1(1)$	0	0	0.0
$A_2(1)$	0.8	0.3	0.5
$A_3(1)$	0.2	0.5	0.32
$B_1(4)$	0.2	0	0.09
$B_2(2)$	0.2	0	0.09
$B_3(2)$	0	0	0

Finally, based upon the Weight Matrix, $w(e'_{ij})$, and $w(e_{ij})$, the teacher can construct IAM as shown in Figure 4.11.

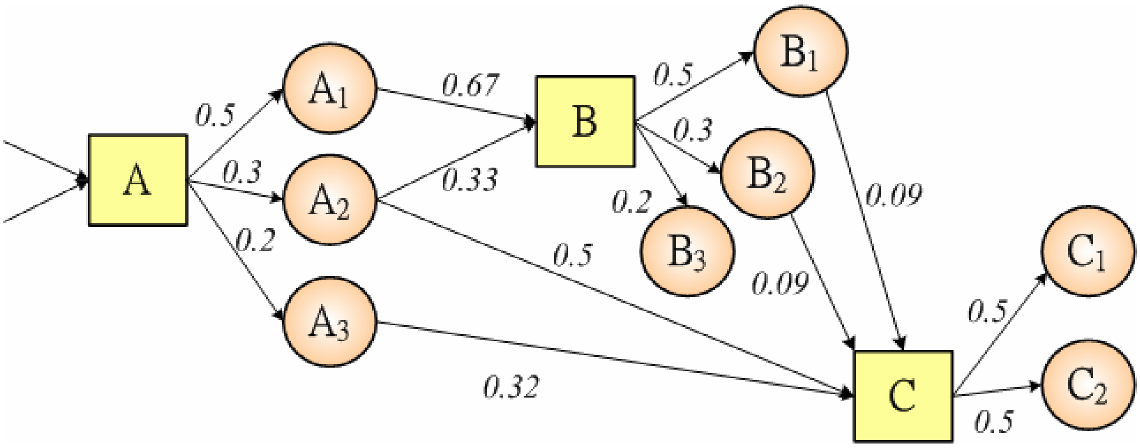


Figure 4.11: The Design of IAM in Part of "*Introduction to Computer*"

4.3 Object Oriented Learning Activity (OOLA) Model

As stated previously, the **Instructional Activity Model (IAM)** is composed of related AT nodes. Each AT node in IAM is modularized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. Accordingly, based on the IAM concept, an **Object Oriented Learning Activity (OOLA)** [81] model is proposed to efficiently represent an adaptive learning activity, which can provide learners with **Content**, **Interaction**, and **Assessment**.

6.2.1 The Definition of Object Oriented Learning Activity (OOLA)

As stated previously, in order to provide teachers with an efficient adaptive learning activity model which can be used to design desired learning activity based on pedagogical theory, reuse the existing learning resources, and share the instructional experiences. Therefore, based on the modularized AT of IAM and object oriented concept, we propose a model, called **Object Oriented Learning Activity (OOLA)**, according to the three basic elements in a learning activity, that is, **Content**, **Interaction**, and **Assessment**. The OOLA Model represents the learning activity with learning content, interaction activity and assessment activity. The directed graph representation and object oriented property can improve the flexibility of constructing an adaptive learning activity. The definition of OOLA is as follows:

Definition 4.1: OOLA is a directed graph, **OOLA** = (**V**, **E**), where

●**V**=$\{N_1, N_2, \ldots, N_n\}$. It denotes a Learning Unit (LU) in a Learning Activity (LA).

The node of OOLA can be divided into the following three types:

(1) N_{LA}**:** denotes a SCORM or IAM compliant learning activity or a single course.

(2) N_{AP}**:** denotes an Application Program (AP), such as Chat Room, Searching Engine

(SE), etc.

(3) N_{EA}: denotes an Exam Activity (EA).

In addition, every node has an attribute, *Learning Duration*, which can be used to control the learning progress by teachers.

●**E={e_1, e_2,.., e_n}**. It is a finite set of directed edge.

In **E** set, some edges$\in$E have Condition Attribute (α) which can be used to set the learning rule for controlling the learning sequencing. In the definition of OOLA, the edges from N_{EA} to other three nodes, N_{LA}、N_{AP}、N_{EA}, have the Condition Attribute, i.e., $\overrightarrow{N_{EA}N_{LA}}$, $\overrightarrow{N_{EA}N_{AP}}$, and $\overrightarrow{N_{EA}N_{EA}}$. The Rule Conditions is represented as "if *condition* then *action*" format. Therefore, if the condition is satisfied, the specified action will be performed and next activity will be triggered for the learner.

Figure 4.12 shows the OOLA model. N_{LA} will be associated with a SCORM or IAM compliant LA or single course to provide learners with a learning content. N_{AP} will be linked to a specific AP. Thus, the AP will be executed by system to offer learner to use while learner is studying the N_{AP} node. While the N_{EA} node in a LA is triggered, the system will display a test sheet for learner. The testing results will be evaluated by assessment scheme to decide whether the learner will go to next advance course or the remedial course. The directed edges denote the learning flow in a LA. These edges from node N_{EA} have *Condition Attribute,* α. After the examination, the concept achievement variables representing the assessment result can be referred by the OOLA to decide the next activity for the learner based on the satisfaction of the rule conditions. If the concept achievement value is lower than a predefined threshold, the remedial course will be provided to the learner. Otherwise, if the achievement value is satisfied, the activity of next step will be provided. In OOLA model, these three nodes can be combined arbitrarily. Therefore, teachers or instructional designers can design their desired learning activity with applying pedagogical theory for providing an adaptive learning environment.

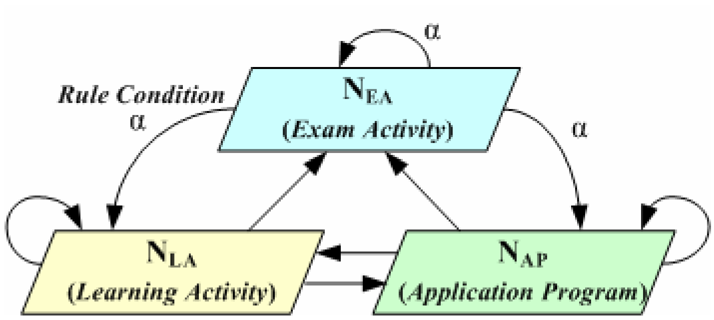


Figure 4.12: The Diagram of OOLA Model

Figure 4.13 shows an example of using OOLA model to represent an adaptive learning activity, which can provide learners with SCORM compliant courses with sequencing rules, learning services, e.g., Chat Room and Searching Engine, and Examines. Also, the remediation will be given according to learners' learning results. Therefore, the learning path will be intelligently guided according to the rule definitions of OOLA model and learners' capabilities.

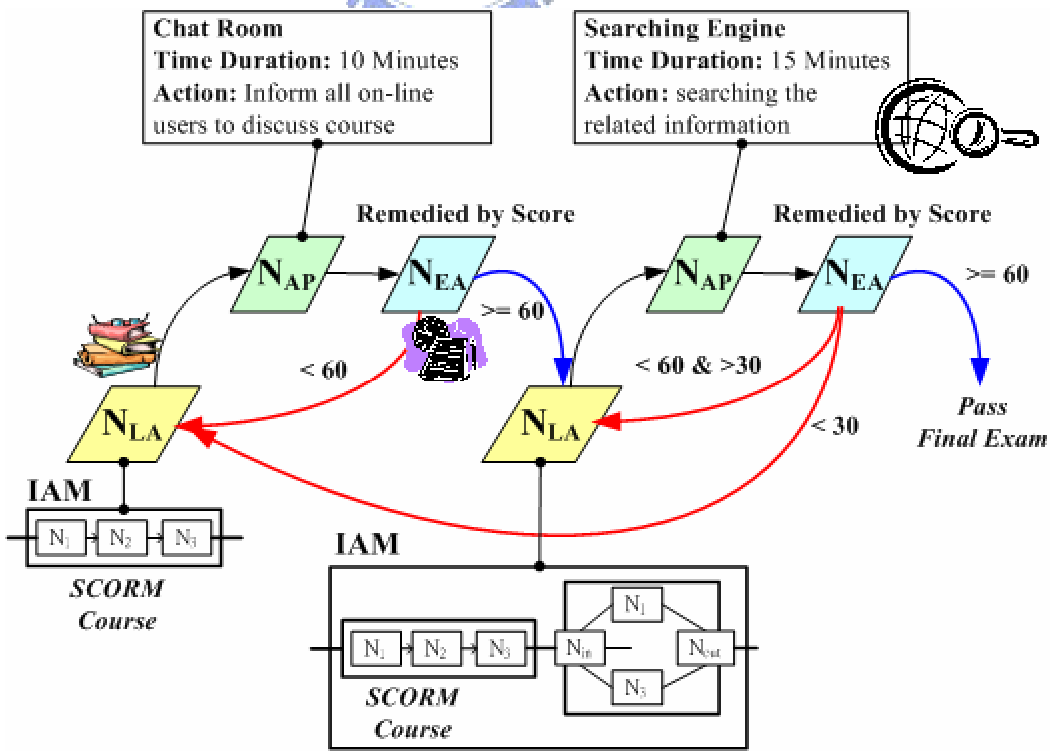


Figure 4.13: An Example of Representing an Adaptive Learning Activity by OOLA

Chapter 5 Knowledge Acquirer (KA)

How to create the standard teaching materials is an important issue. Although most of approaches usually offer an authoring tool to help users, authoring standard teaching materials is still time-consuming, even though to often practice it. In addition, the traditional teaching material without concept of learning object is difficult to offer appropriate teaching materials for students in accordance with their aptitudes. Therefore, in this dissertation, in Knowledge Acquirer (KA) module of ILCMS, a **Learning Content Editor (LCE)** and an **OOLA authoring tool** [81] are developed. The former proposes a **Content Transformation Scheme (CTS)** [114], which can efficiently transform the traditional teaching materials, e.g., HTML and PPT file format, into SCORM compliant learning contents, and a SCORM 2004 compliant authoring tool with **Object Oriented Course Modeling (OOCM)** [117] approach based upon High Level Petri Nets (HLPN) theory, which can help teachers or editors efficiently create the course with desired learning sequencing guidance of SCORM standard. In addition, in order to construct OOLA compliant learning activity, the latter is a user-friendly GUI authoring tool, by which teachers can efficiently edit desired learning activity with associated SCORM compliant course in LOR, test sheet in TIB, and application program (AP) in APR. The details of KA module are described below.

5.1 Transformation of Traditional Teaching Material

Based upon the concept of learning object, the Content Transformation Engine segments the traditional teaching materials into several learning objects. The original teaching materials are divided into several objects according to the instructional objectives defined by teachers or educational experts. Moreover, we adopt the SCORM standard to present and package the learning object with Extensible Markup Language

(XML) format in each teaching material for achieving the reusing, sharing, and interoperability of these learning objects. In this section, we will present whole transformation process of traditional teaching material.

5.1.1 Concept of Learning Object

The concept of learning object is to define a meaningful learning content, including multimedia content or instructional content, which can be used, reused, shared, and recombined. The learning object model can be presented as independent chunks of educational content which can be created to provide an educational experience or teaching strategy. Like the concept of object-oriented programming (OOP), the learning objects are self-contained, and they can contain references to other learning objects and may be combined or sequenced to form longer educational units. By the concept learning object, we can develop an individualized tutoring system to offer appropriate teaching materials for students in accordance with their aptitudes.

Unfortunately, at present, the most popular teaching materials, e.g., either the PowerPoint or HTML, are the traditional teaching materials. How to distinguish whether the teaching material is traditional or not? In accordance with our definition, if a teaching material without concept of learning object, we categorize it into traditional teaching material. As shown in Figure 5.1(a), the traditional teaching material usually arranges the learning content and quizzes in sequence monotonously. It means that all the students learn the same teaching materials sequentially without allowing skipping the subsections they have learned. In this way, without appropriate segmenting and labeling the teaching materials, it is difficult for an individualized tutoring system to offer learner an appropriate teaching material.

Therefore, how to create a teaching material with concept of learning object has become an important issue. Therefore, in this dissertation, we propose an approach based upon the idea of segmenting the original teaching materials into several objects, called learning objects as mentioned above. Figure 5.1(b) shows the object-oriented teaching material based upon this idea, and the original teaching materials are divided into several learning objects according to the instructional objectives defined by teachers or educational experts.

In addition, to be able to use learning objects in an intelligent content management system, we have to tag or label learning object with metadata to describe that what they contain, communicate, and require. Thus, how to reliably tag or label these learning objects is necessary. For the reason of reusing, sharing, and interoperability, we adopt the SCORM 2004 standard to present and package the learning object with Extensible Markup Language (XML) [132] [141] format in each teaching material.

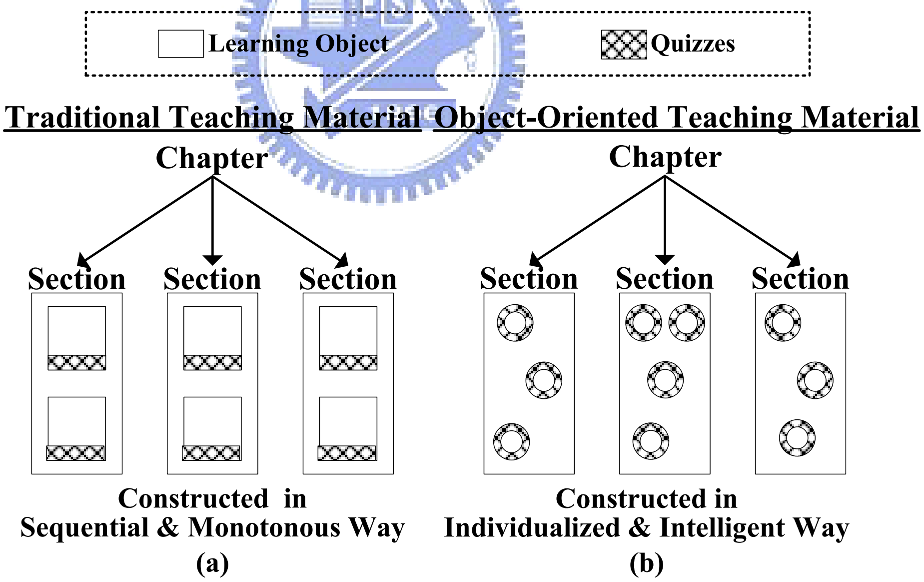


Figure 5.1: Traditional Teaching Material and Concept of Learning Object.

5.1.2 Content Transformation Scheme (CTS)

Currently, most of existing teaching materials without characteristics of learning object are sequential format and presented by the PowerPoint or HTML file format. For this reason, if we can transform the traditional teaching materials into a type of learning object with SCORM standard, the creating time will be considerably decreased. Therefore, how to fast and easily create SCORM complaint teaching materials from traditional teaching materials becomes an interesting and important issue. Thus, in this dissertation, we develop a transformation scheme to divide the sequential teaching material into separate learning objects with SCORM metadata tags.

For content transformation scheme, Figure 5.2 offers the diagram of three approaches which help teachers or editors create the SCORM complaint teaching materials and their explanations are described as follows:

- **User Definition Mode:** As shown in Figure 5.2(a), this mode provides teachers or editors to define the learning content of each chapter or section, which consists of some related file, by themselves. Then, they can upload these files onto server through the content transformation engine. The content transformation engine will tag every learning-content, i.e. chapter or section, and then package the each learning object including related resources into SCORM compliant teaching material by the content packaging scheme.
- **PPT Transformation Mode:** We also propose an (semi-)automatically extracting scheme to transform a single PowerPoint file into several learning objects. As shown in Figure 5.2(b), our proposed scheme can automatically extract each chapter or section from uploaded PowerPoint files and package learning objects into SCORM compliant teaching materials.
- **Authoring Mode:** Not only transforming traditional teaching materials into SCORM standard is important, but also offering the authoring tool to edit SCORM

compliant teaching material is necessary. Accordingly, we also develop a browser based authoring tool to help teachers or editors edit standard teaching materials, as shown in Figure 5.2 (c).

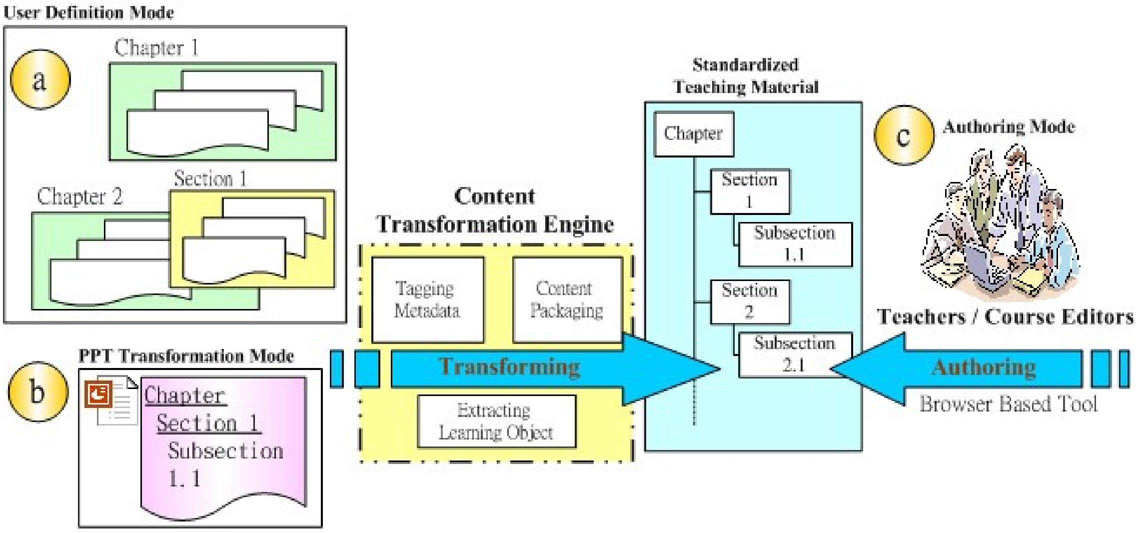


Figure 5.2: Diagram of Three Modes for Standardized Transformation Scheme of Traditional Teaching Material.

Figure 5.2 presents the whole concept of transformation scheme. Now, we will precisely explain the process of content transformation scheme in more detail. As shown in Figure 5.3, the content transformation scheme consists of following five steps:

Step 1: ***Tagging the SCORM Metadata:*** in this step, teachers or editors have to fill the related metadata information of learning content, which can easily and fast use, search, and manage the learning content.

Step 2: ***Defining the Section Unit:*** how to define a meaningful learning object is a difficult issue, so we consider that the teachers or editors themselves are the most appropriate person to determine the coverage of each learning object, i.e., section or chapter. For this reason, in this step, the teachers or editors have to define the coverage of learning object and then the teaching material will be segmented accordingly.

Step 3: ***Extracting Section Unit:*** extract each learning object into independent form original teaching material and then each separate meaningful leaning object can be placed at appropriate location and tagged with SCORM metadata. However, in the user definition mode, this step will be passed since the independent files of learning object have been offered by authors.

Step 4: ***Generating Organization of Learning Content*****:** In accordance with the defined coverage of learning object, automatically generate the organization of whole teaching material with XML language.

Step 5: ***Packaging the Tailored Courseware*****:** After the former four steps, we can use the SCORM content packaging scheme to integrate the metadata, organization, and learning resource into SCORM compliant courseware package.

By these methods mentioned above, we believe that teachers or editors can easily and fast transform the existing traditional teaching materials into SCORM compliant.

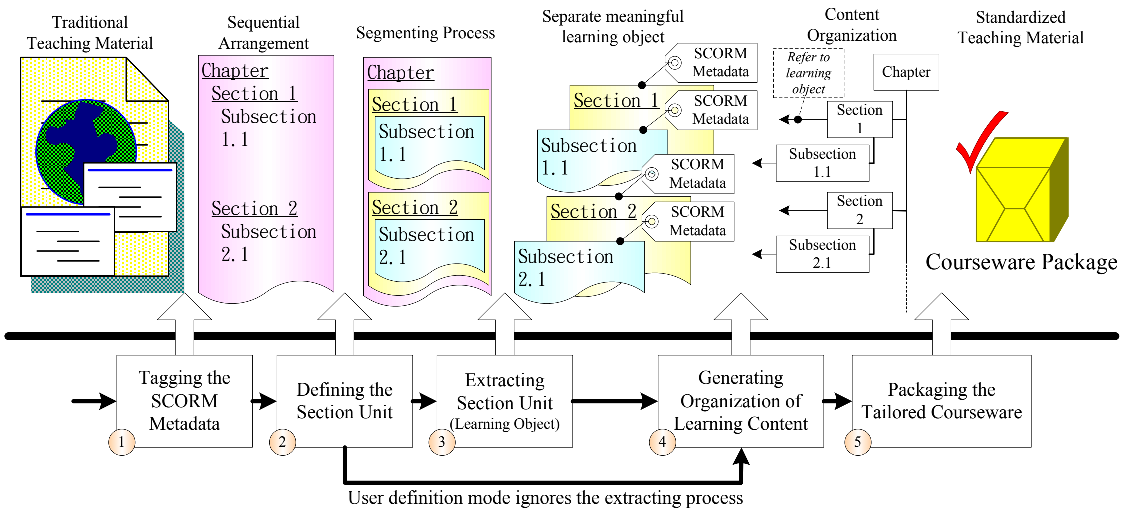


Figure 5.3: Flowchart of Content Transformation Scheme (CTS)

5.2 Object Oriented Course Modeling (OOCM)

The structures with complicated sequencing rules of activity tree in SCORM make the design and creation of course sequences hard. Therefore, how to provide a user-friendly authoring tool, which can represent the course as a graph and transform it into SCORM compliant course file automatically, to efficiently construct SCROM compliant course becomes an important issue. However, before developing this kind of authoring tools, how to provide a systematic approach to analyze the sequencing rules and transform the created course into SCORM compliant are our concerns. Therefore, in this dissertation, we apply the High-Level Petri Nets (HLPN) [59] [60] [62] [70] [71] [73] [82] [84], which is a powerful language for system modeling and validation, to model the basic sequencing components as the middleware, called *Object-Oriented Activity Tree* (OOAT), for constructing the SCORM course with complex sequencing behaviors. Thus, according to these OOATs, we can model a complex structure of course with different learning guidance. Then, two transformation algorithms are also proposed to transform the created course into SCORM compliant one described by XML language. The Figure 5.4 shows the idea of Object Oriented Course Modeling (OOCM) approach.

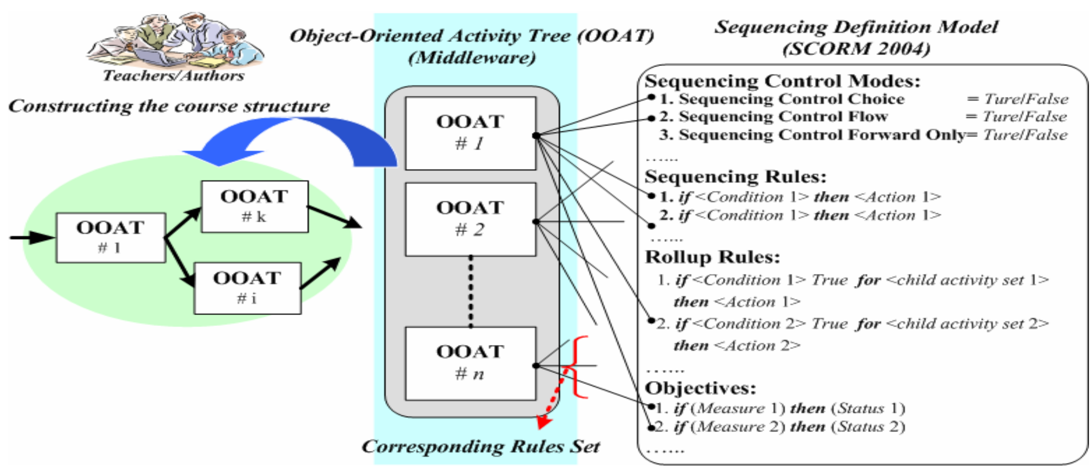


Figure 5.4: The idea of Object Oriented Course Modeling (OOCM)

5.2.1 The Scheme of OOCM

Based upon the concept of object-oriented methodology (OOM) and High-Level Petri Nets (HLPN) theory, we can model several basic sequencing components with specific sequencing behaviors in SN, which can be easily used to model complex structure of course. Therefore, in Figure 5.5, the OOCM process includes four processes as follows:

(1) **OOAT Modeling with HLPN**: apply HLPN to model five basic sequencing components as the middleware with corresponding structure of AT and specific basic sequencing behaviors, called *Object-Oriented Activity Tree* (OOAT).

(2) **Course Construction with OOAT:** use these basic sequencing components (OOAT) to model complex structure of course with different learning guidance based upon the HLPN theory.

(3) **PN2AT Process**: transform the modeled course structure into tree-like SCORM-compliant AT with sequencing definition of SN.

(4) **AT2CP Process:** package the transformed AT structure with corresponding physical learning resources and then generate the content packaging course of SCORM.

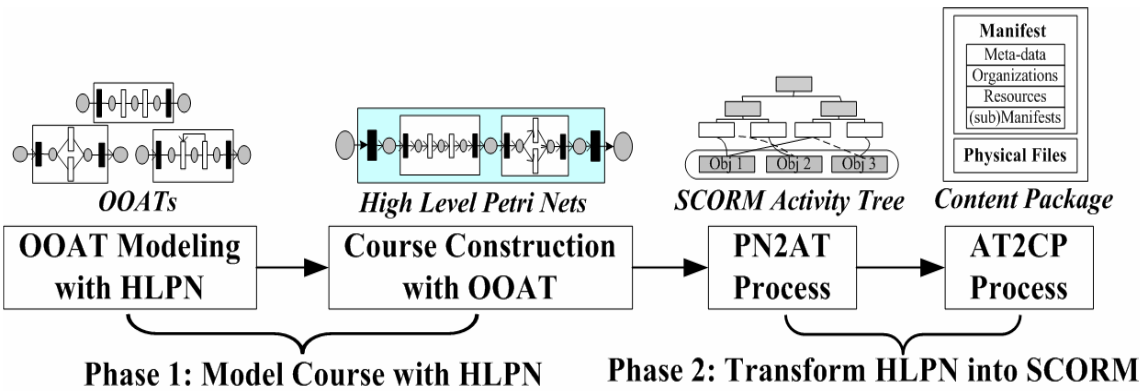


Figure 5.5: The Flowchart of Object Oriented Course Modeling (OOCM)

5.2.2 The OOAT Modeling with High Level Petri Nets (HLPN)

As shown in Figure 4.2, an AT in SCORM 2004 is structured by a set of clusters. A cluster, the basic sequencing building block, is an organized aggregation of activities consisting of a single parent activity and its first level children, but not the descendants of its children. The parent activity of a cluster will contain the information about the sequencing strategy for the cluster. The status information of all child activities will be collected and used to sequence these activities in the structure. Each cluster has a *Sequencing Definition Model* (*SDM*) to define a set of elements that can be used to describe and affect various sequencing behaviors. In this dissertation, we only take six out of ten rule definitions in SDM into account, that is, **1) Sequencing Control Modes**, **2) Sequencing Rules**, **3) Rollup Rules**, **4) Objectives**, **5) Objective Map**, and **6) Delivery Controls**, because these six rule definitions can perform the most of sequencing behaviors in SN. Therefore, we apply HLPN to model several basic sequencing components as a cluster with corresponding structure of AT and specific basic sequencing behaviors, called OOAT, which can be used to model a complex structure of a course. Thus, based upon these OOATs and OOCM approach, the remaining rule types in SDM could be analyzed and modeled in a similar way. Here, an OOAT can be represented as a Chapter or Section. For modeling the sequencing behaviors in SN, firstly, the OOAT in HLPN is defined as follows:

Definition 5.1: The HLPN of Object-Oriented Activity Tree (OOAT) is a 6-tuple,

OOAT = (P, T, Σ, A, G, E), where

1. **P** = $\{p_1, p_2, \ldots, p_m\}$ is a finite set of places. **P** includes five types of places: $\mathbf{P_G}$ denotes the global objectives, $\mathbf{P_L}$ denotes the local objectives, $\mathbf{P_M}$ denotes the connector between transitions, $\mathbf{P_R}$ checks whether the transition executes the Rollup process or not, and $\mathbf{P_W}$ checks whether the transition defines the global objective

($\mathbf{P_G}$) or not. Besides, in connective places ($\mathbf{P_M}$), we use $\mathbf{P_{in}}$ and $\mathbf{P_{out}}$ to represent the starting place and ending place of an OOAT component. $\mathbf{P_G}$ and $\mathbf{P_L}$ contain tokens recording the information in Tracking Status Model (TSM).

2. $\mathbf{T}$ = $\{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions ($P \cap T = 0$). **T** includes four types of transitions: $\mathbf{T_A}$ denotes a learning activity or a sub-OOAT component, $\mathbf{T_M}$ denotes the connector between OOAT components, $\mathbf{T_R}$ rolls up all learning status of its children, and $\mathbf{T_O}$ will set the global objective ($\mathbf{P_G}$) of an activity according to its local objective ($\mathbf{P_L}$).

3. $\mathbf{\Sigma} = \langle \mathbf{C_{TSM}}, \mathbf{C_O} \rangle$ is the non-empty finite color sets of tokens. $\mathbf{C_{TSM}}$ represents the *Tracking Status Model* (TSM) in SN, which records the learning information of *Activity Progress Information*, *Attempt Progress Information* and *Objective Progress Information* of learners. $\mathbf{C_O}$ denotes the *ordinary color*, corresponding tokens without information, which is applied to initialize or trigger a learning process.

4. $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed arcs. $\overrightarrow{PT}$ is the arc from a place to a transition; $\overrightarrow{TP}$ is the arc from a transition to a place.

5. **G:** is a *guard function*. The firing rule $G(t)$ of a transition ($t \in T$) is defined as "*if-else*" form in SDM. The guard function can generate specific sequencing behaviors. In OOAT, we define the following *guard functions*:

   - **G($\mathbf{T_A}$):** define the sequencing rules of SDM and specify whether a learner is ready or not to learn the activity according to her/his learning results in previous activity.
   - **G($\mathbf{T_R}$):** control the rollup process of an activity based upon the Rollup rules definition of SDM.
   - **G($\mathbf{T_O}$):** set the learning status of the global objective according to local objective of activity (**$\mathbf{T_A}$**). In SDM, teachers can define how to read/write a global objective for different course sequencing.

6. **E:** is an *arc expression function*. **E(*a*)**, $\forall a \in A$, denotes the information that how many and which kinds of token colors should be removed from the input places and added to the output places. In OOAT, we define the *expression functions* as shown in Table 5.1.

In addition, Figure 5.6 shows the basic diagram of HLPN of OOAT. As mentioned in Definition 5.1, the connectors, P_M and T_M, pass the token only, T_R enabled by P_R with ordinary token $<C_O>$ executes the rollup process according to the token $<C_{TSM}>$ carrying the learning information. Besides, in the right part of Figure 5.6, T_O will change the type of a place, e.g., P_{out}, into P_G if P_W has ordinary token $<C_O>$.

Table 5.1: The Arc Expression Function **E(*a*)** and its Related Token Color.

Arc Expression Function	Token
E($\overrightarrow{P_G T_A}$), E($\overrightarrow{P_G T_M}$)	$<C_O+C_{TSM}>$
E($\overrightarrow{T_A P_L}$), E($\overrightarrow{T_R P_L}$), ($\overrightarrow{P_L T_R}$), E($\overrightarrow{P_M T_R}$), E($\overrightarrow{T_R P_M}$), E($\overrightarrow{T_O P_G}$), E($\overrightarrow{P_L T_O}$), E($\overrightarrow{P_L T_A}$)	$<C_{TSM}>$
E($\overrightarrow{T_A P_M}$), E($\overrightarrow{P_M T_A}$), E($\overrightarrow{T_A P_G}$), E($\overrightarrow{T_M P_G}$), E($\overrightarrow{P_M T_M}$), E($\overrightarrow{T_M P_M}$), E($\overrightarrow{P_W T_O}$), E($\overrightarrow{P_R T_R}$)	$<C_O>$

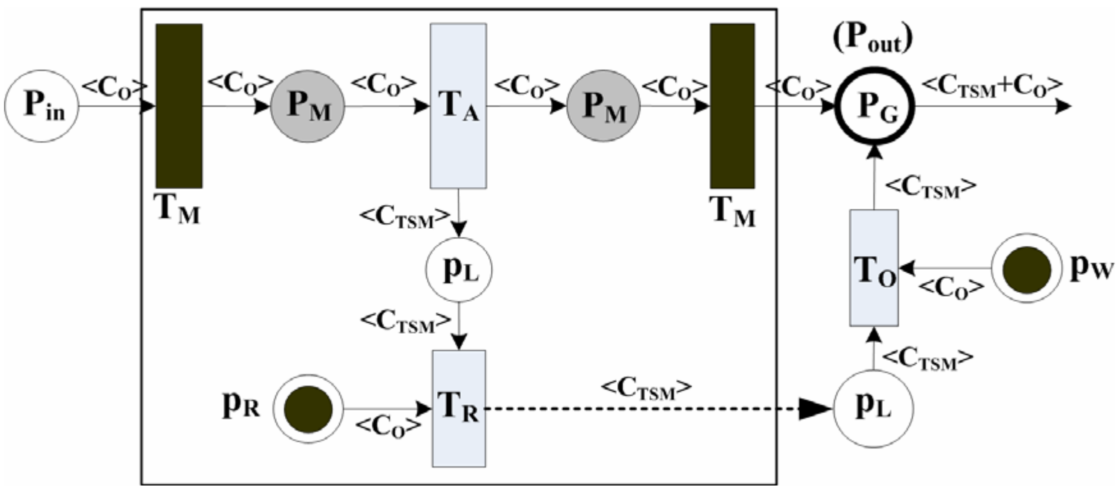


Figure 5.6: The Diagram of HLPN of OOAT

According to the sequencing behaviors in SN specification, we propose five OOAT components, 1.***Linear***, 2.***Choice***, 3.***Condition***, 4.***Loop***, and 5.***Exit***, to model different learning strategies. Figure 5.7 shows these five basic sequencing components of OOATs with its corresponding structures of courses and related definitions of Guard functions, and Table 5.2 shows their related **Sequencing Definition Model (SDM)** including *Sequencing Control Mode (SCM)* which controls the navigation behaviors, *Objective* which defines the requirements of evaluated conditions, and *Sequencing Rules* which define the evaluated conditions of course sequencing during learning activity. Here, every *guard function* of OOAT can be mapped to corresponding *sequencing rules* in SDM, which record the sequencing behaviors of learning activity in SCORM AT. In Figure 5.7, the ***Linear*** OOAT (5.7.a) denotes that the learners can learn the activity (transition) straightforward. Therefore, "*Sequencing Control Flow*" in SCM is set as true. The Rollup transition (T_{Rollup}) will collect the status information of related local objective places (P_L) in included child transitions (activities) to evaluate the value of P_L in parent transition. The ***Condition*** OOAT includes ***Conditional Linear*** (5.7.c) and ***Conditional Choice*** (5.7.d). The former is a ***Linear*** OOAT with conditional criteria (α) that checks whether an activity will be assigned to a learner or not according to his/her learning result in previous activity. For example, in Figure 5.7.c, the token, <$\mathbf{C_{TSM}}$>, will be delivered to the local objective (P_{L1}) after learning the activity (T_{A1}). Then, according to the activity's tracking information (TSM) and related guard function in T_{A2}, the next transition (T_{A2}) may be accessible (fired) if the condition α_1 is true. The latter is similar to the ***Choice*** component. According to the previous learning status stored in global objective P_G, an activity (T_{Ai}) can be selected by learners if its conditional criterion (α_i) is true. Figure 5.7.e shows the ***Loop*** OOAT which can control the learners to study continuously the same activity or previous one according to the conditional criteria (α_1 and α_2). In addition, in Figure 5.7.f, the **Exit** OOAT controls the termination of learning process. For example, after learning the T_{A1}, the token, <C_{TSM}>, will be delivered to P_{L1}. Then, according to the tracking information of T_{A1}, learners will finish the component if the condition α is true.

Table 5.2: The Related SDM definition of OOAT.

OOAT Types	Sequencing Control Mode	*Objective*	Sequencing Rules
Linear	Flow = *true* Forward Only = *true* Choice Exit=*true*		
Choice	Choice = *true* Choice Exit = *true*		
Conditional Linear	Flow = *true* Forward Only = *true* Choice Exit=*true*	***Objective:*** ● Satisfied by Measure = *true* ● Minimum Satisfied Normalized Measure = α_i	***Postcondition Rule:*** ● **if** α_i =*true* **then** *continue* **else** *retry, 1≦i≦n-1.*
Conditional Choice	Flow = *true* Choice = *true* Choice Exit = *true*	***Objective:*** ● Satisfied by Measure = *true* ● Target Objective ID = OBJ P_G ● Read Satisfied Status = *true* ● Read Normalized Measure	***Precondition Rule:*** ● T_{Ai}: **Read** OBJ P_G (Global Objective) ● **if** α_i ≠*true* **then** *hiddenFromChoice, 1≦i≦n.*
Loop	Flow = *true* Choice Exit=*true*	***Objective:*** ● Satisfied by Measure = *true* ● Minimum Satisfied Normalized Measure = α_1 / α_2	***Postcondition Rule:*** ● T_{A2}: **if** α_1 / α_2 =*true* **then** *previous / retry* **else** *continue*
Exit	Flow= *true* Forward Only = *true* Choice Exit= *true*		***Postcondition Rule:*** ● T_{A1}: **if** α =*true* **then** *Exit Parent / exitAll*

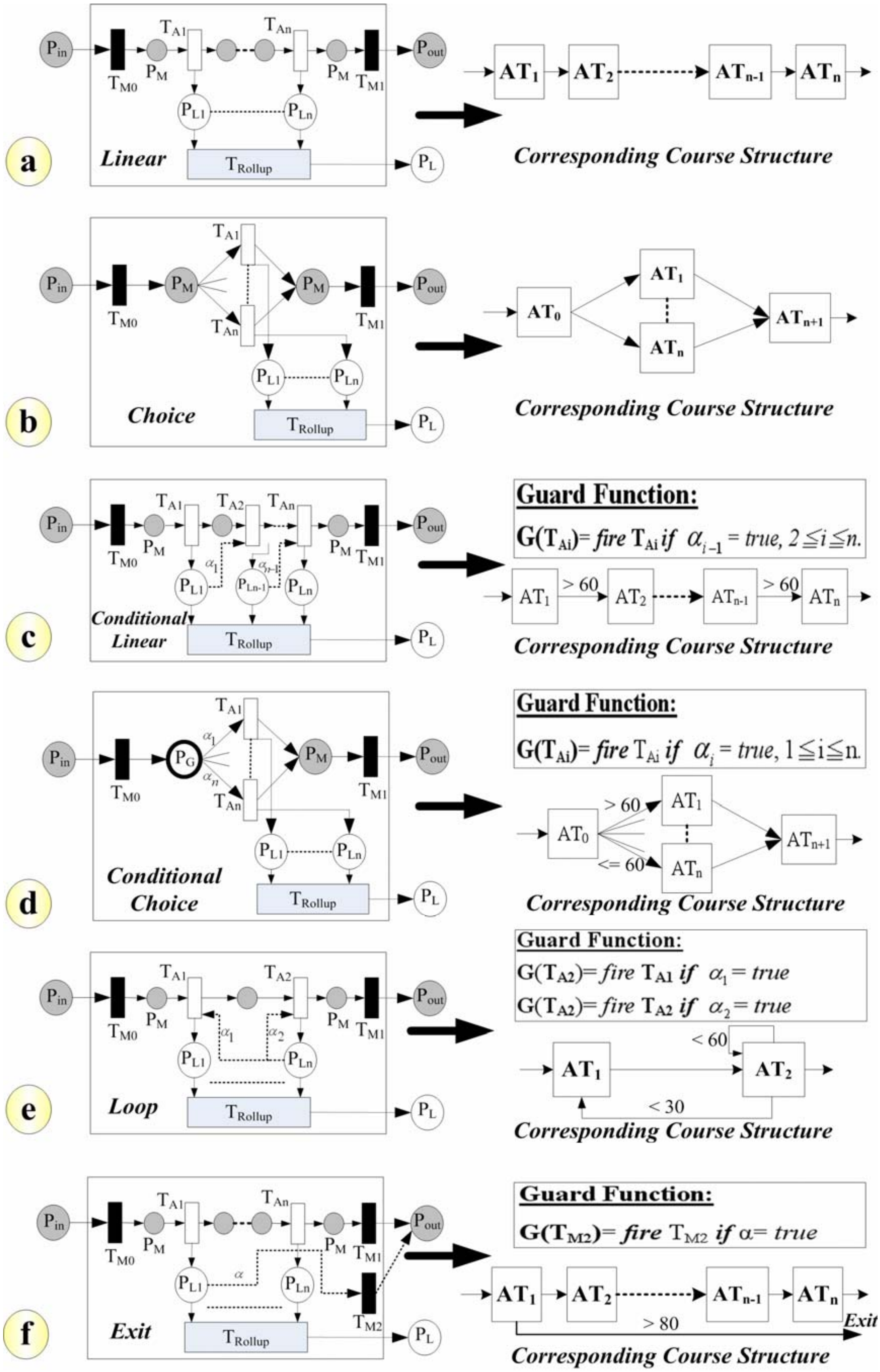


Figure 5.7: The Five Sequencing Components of OOATs

5.2.3 Sequencing Rules Modeling of SDM

In SDM, each Sequencing Rule consists of a set of conditions and a corresponding action in ***if*** *[condition_set]* ***then*** *[action]* format. A sequencing behavior of activity associated with the rule's action will be executed if the rule's condition-set evaluates to True. Thus, different definition of sequencing rules will result in different learning guidance. However, how to define the appropriate sequencing rules within course is an important issue. Therefore, in this section, we define these sequencing conditions as tokens used to determine whether an activity is accessible or not, e.g., symbol ”α_i” in Figure 5.7. Besides, the OOATs are used to model the rule's actions for modeling the sequencing behaviors of SCORM course. The structure of a sequencing rule is shown in Figure 5.8.

In SN specification, the sequencing rules of SDM include the following rule's actions:

(1) Precondition Actions: decide whether an activity will be selected or not for learning. These actions will be executed while an activity will be selected. Its action elements and corresponding OOATs are shown in Table 5.3.

(2) Postcondition Actions: control the sequencing flow according to learning result of learners after learning an activity. These actions will be executed while an activity has been finished. Its action elements and corresponding OOATs are shown in Table 5.4.

(3) Exit Actions: This action will be executed after a descendant activity has been finished or some condition is satisfied. It is controlled by a SCORM complaint learning management system (LMS). Thus, we can set the system commend, ***Exit***, to inform LMS for finishing the whole course.

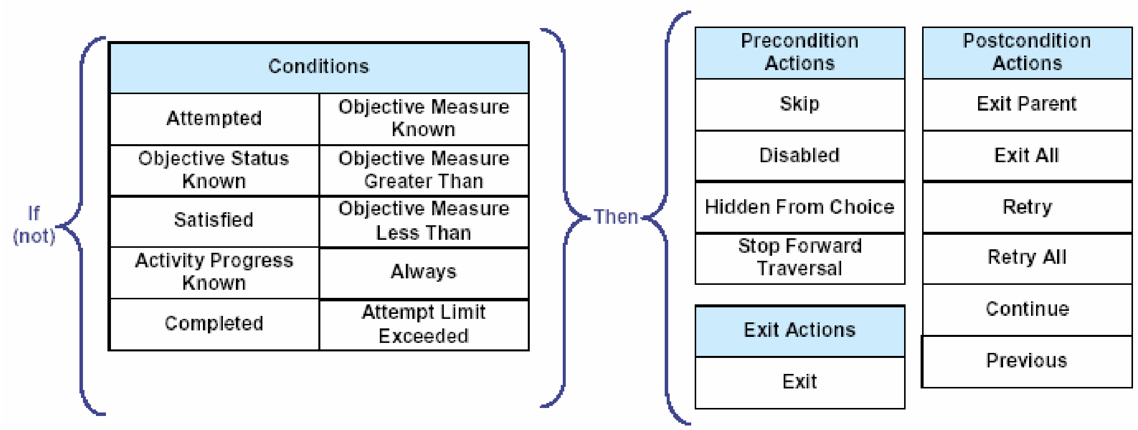


Figure 5.8: The Structure of Sequencing Rules

Figure 5.9 shows the example of ***Skip*** action modeled by *Conditional Choice* OOAT, which represents that if the rule condition α is false, the activity T_{A1} will be skipped and then the T_{A2}, which doesn't execute any learning activity, will be triggered according to the definition of guard function. The ***Disabled*** Action can also be modeled by *Conditional Linear* OOAT as shown in Figure 5.7.c. In postcondition actions, the **Exit Parent** action can be modeled by **Exit** component shown in Figure 5.7.f. For **Retry** action in Figure 5.7.e, the token $<C_{TSM}>$ of T_{An} is delivered to P_{Ln}. Then, T_{An} will be relearned if condition α_2 is true according to its learning status of local objective (P_{Ln}).

Table 5.3: The Action Types and Corresponding OOATs of **Precondition** in Sequencing Rules.

Action Element	Description	OOATs
Skip	this action will omit an activity to be learned.	*Conditional Choice*
Disabled	this action will block an activity to be learned.	*Conditional Linear*
Stop Forward Traversal	this action will terminate learners to continuously navigate learning activity forward.	*Conditional Linear*
Hidden From Choice	this action will stop the choice of activity	*Conditional Choice* with "*Sequencing Control Choice*" is false.

Table 5.4: The Action Types and Corresponding OOATs of **Postcondition** in

Sequencing Rules.

Action Element	Description	OOAT
Exit Parent	this action terminates a activity	*Exit*
Exit All	this action terminates whole activity tree (course)	*Exit*
Retry	this action makes learner to relearn some previous activities if its condition is evaluated as true.	*Loop*
Retry All	this action makes learners to relearn all previous activities if its condition is evaluated as true.	*Loop*
Continue & Previous	this action makes learners to learn next or previous activity respectively.	*Conditional Linear & Loop*

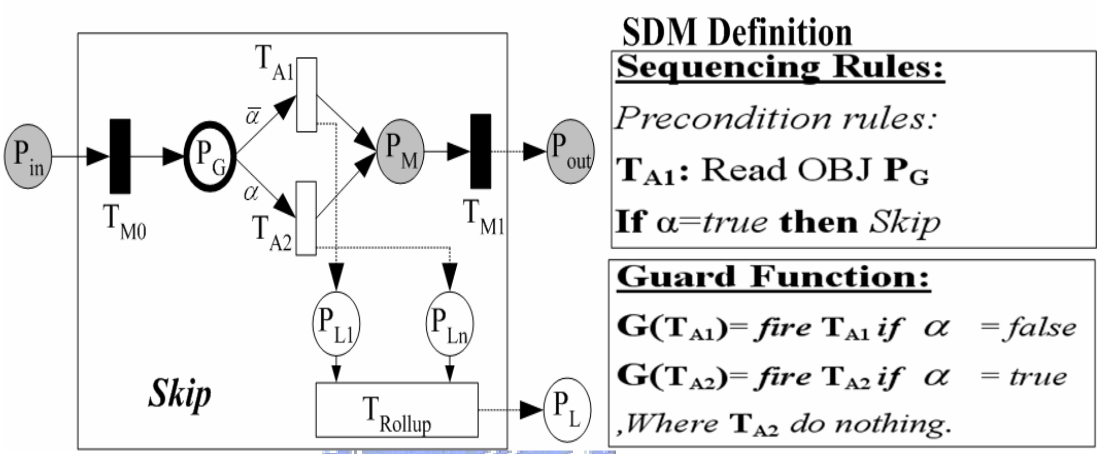


Figure 5.9: An Example of modeling *Skip Action* in Sequencing Rules by *Conditional Choice* OOAT

5.2.4 Objective Modeling

In SN, each activity has many associated learning objectives which include two types: *local objectives* and *global objectives*. The local objective which can only be referred by its associated activity and the global objective which can be shared between activities for the more complex instructional designs define how to evaluate an activity's *objective progress information*. Therefore, in OOATs, each transition (activity) has one local objective (P_L) and global objective (P_G) which will be defined if necessary. As shown in Figure 5.10, in general, the transition (T_{A1}) only has one local objective (P_L) and no global objective. Here, the "Minimum Satisfied Normalized Measure = 0.6" means that the score of learner must exceed 0.6. After learning T_{A1}, a Token <C_{TSM}> with *Objective Progress Information* of T_{A1} is delivered to P_L for recording the related learning information. Then, if P_w is assigned an ordinary Token <C_O>, the T_O will set the connector transition (P_M) as a global transition (P_G) for sharing the learning results with another transition (T_{A2}). Then, according to guard function $G(T_O)$, the P_G will be set as *satisfy* because the score (0.7) is greater than 0.6.

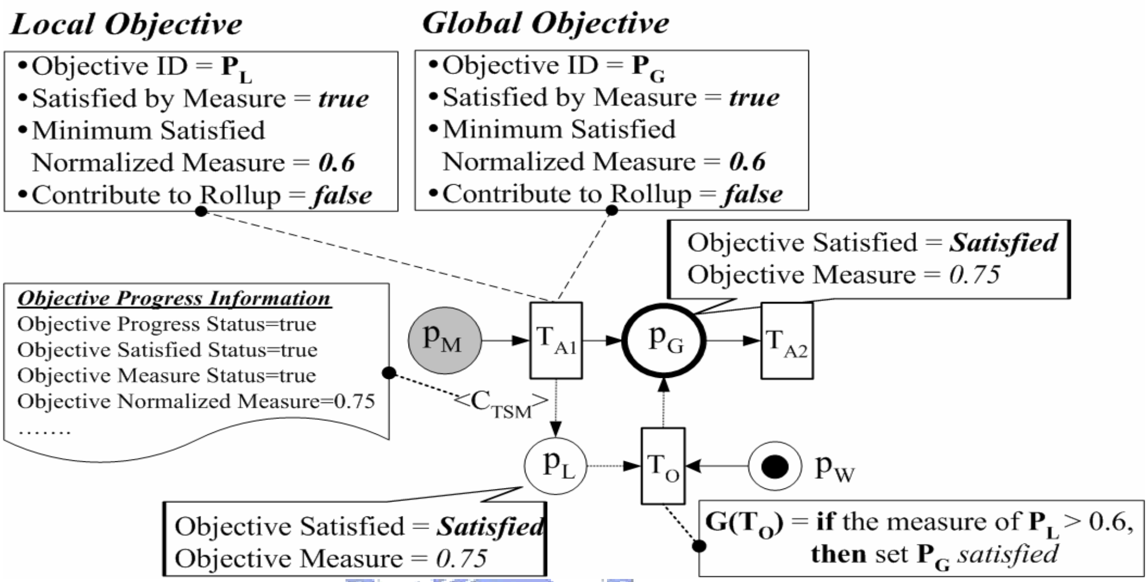


Figure 5.10: The Process of Objective Reference

5.2.5 Rollup Rule and Delivery Control Modeling

In SN, cluster activity, which is the basic sequencing building block, can be applied with a set of zero or more Rollup Rule*s* which are evaluated during the overall Rollup Process. Each Rollup Rule is defined as "***if** [condition_set] True **for** [child activity set] **then** [action]"* format, which denotes that if the set of conditions (*condition_set*) evaluates to True from the tracking information of included child activities (*child activity set*), corresponding action (*action*) will set the cluster's tracking status information. Figure 5.11 shows the structure of a Rollup Rule.

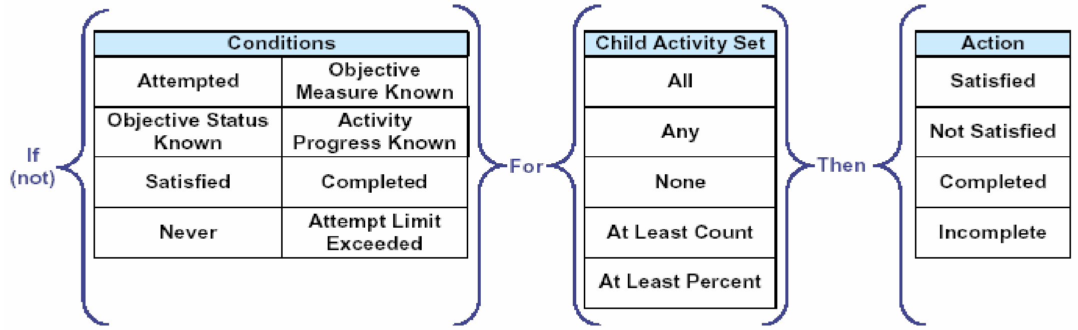


Figure 5.11: The Structure of Rollup Rules

As mentioned above, in OOATs, we use the T_{Rollup} transition to process the Rollup rules for evaluating the learning results of learners in a cluster. The T_{Rollup} transition can be modeled by HLPN as shown in Figure 5.12. Here, in T_{Rollup}, each T_R transition will evaluate the learning status recorded in associated local objective (P_L) if its P_R transition is marked by an ordinary token $<C_O>$, where P_R transition enables or disables the Delivery Controls, which is used to manage the activity's tracking status information, in SDM. For example, in Figure 5.12, because the P_{R1} of T_{A1} isn't marked by a token $<C_O>$, T_{A1} won't be triggered but others with Tokens will be triggered to execute the rollup process. Moreover, according to the definition of Rollup Rules, the learning status of OOAT will be set as *satisfied* in P_L if at least two activities (transitions) within it are *satisfied*.

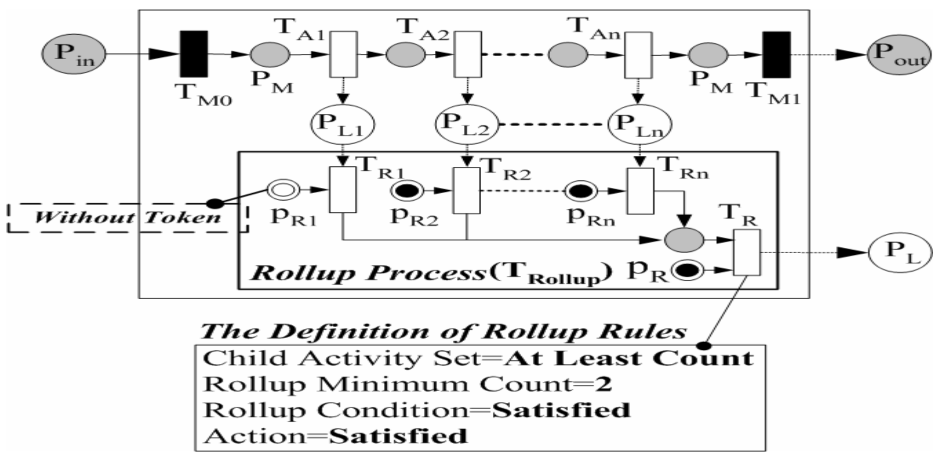


Figure 5.12: The Rollup Model in OOATs

5.2.6. Activity Tree Transformation Process

As mentioned above, we have described how to model the HLPN model of course sequences in SCORM by our proposed OOATs. Therefore, in this section, how to transform the HLPN model into SCORM compliant course will be described. In this dissertation, we propose two algorithms, called **PN2AT** (Petri Nets to Activity Tree) and **AT2CP** (Activity Tree to Content Package), to do the activity tree transformation process.

PN2AT Process:

In OOAT, each transition with included child transitions can be represented as a cluster of AT in SCORM. Thus, an algorithm, called **PN2AT**, transforms each non-terminated transition into a cluster with associated sequencing definitions in SDM and integrates them to construct the structure of AT. For example, in Figure 5.13, an HLPN model of course can be decomposed as a hierarchical structure. In every level, a non-terminated transition, e.g., TA_1, will be represented as a root-node (AA) and included sub-transitions (TA'_1 and TA'_2) will be represented as the child nodes (AA and AB), which form a tree-like structure as a cluster with associated sequencing definition of SDM in AT. Then, we can recursively transform all non-terminated transitions by the same process.

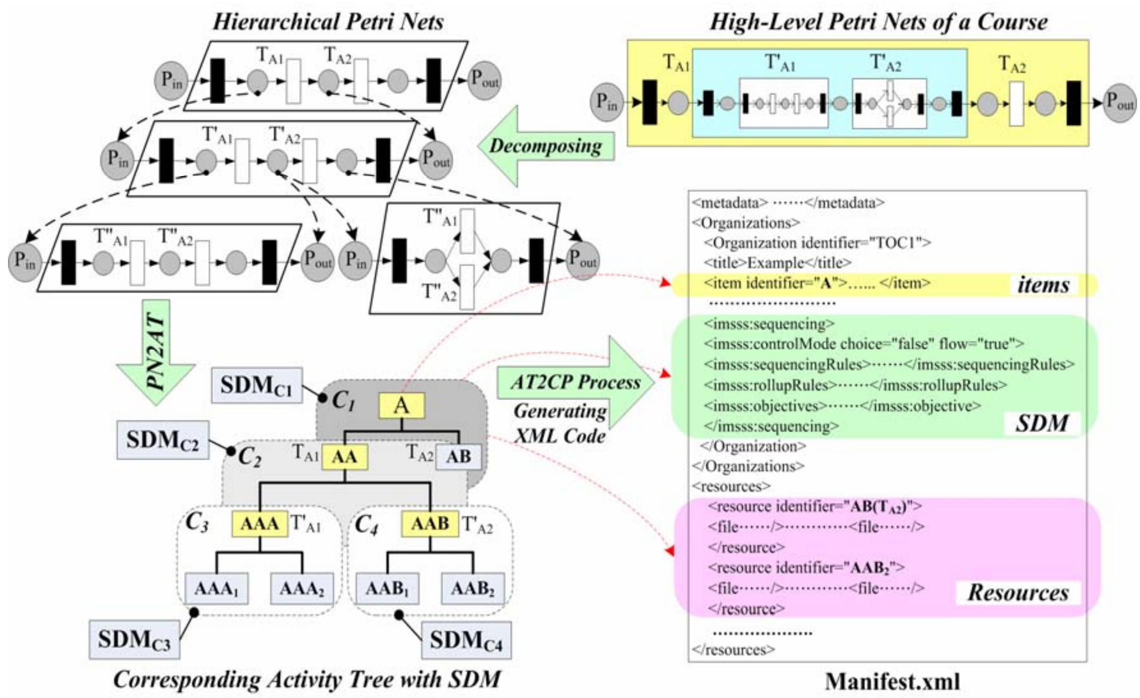


Figure 5.13: An Example of PN2AT and AT2CP Process

Algorithm 5.1: PN2AT Algorithm

Definition of Symbols:

AT_F: denote the final AT with XML code.

C_i : denote a tree-like cluster.

Input: The HLPN model of a course

Output: **AT_F**

Step 1: for each $T_i \in$ HLPN model

1.1: if T_i is a non-terminated transition

then create a tree-like cluster $\boldsymbol{C_i}$

1.2: insert T_i as root node and its included sub-transitions T_k as child nodes into $\boldsymbol{C_i}$

1.3: generate the corresponding XML codes according to its structure type of OOAT and sequencing definitions including *Sequencing Control Mode, Sequencing Rules, Rollup Rules,* and *Objective definitions* in SDM for $\boldsymbol{C_i}$ into appropriate position of AT_F.

1.4: if $\exists T_k \in \boldsymbol{C_i}$ is a non-terminated transition

then execute recursively the same processes as **Step 1.1**.

Step 2: Output the **AT_F**

AT2CP Process:

After transforming the HLPN model of a course by PN2AT process, the structure and sequencing definitions of SCORM course without physical learning resources can be generated. Therefore, according to content packaging scheme of SCORM, an algorithm, called **AT2CP**, will be used to package the structure of AT and its related physical learning resources into a SCORM compliant course file described by XML language. The AT2CP process is also shown in right side of Figure 5.13.

Algorithm 5.2: AT2CP Algorithm

Definition of Symbols:

PF: denote a temporary place which collects related physical learning resources of AT.

CP: denote the content package file of SCORM.

Input: Activity Tree (**AT**) generated by PN2AT algorithm.

Output: Content Package (**CP**).

Step 1: For each leaf node in AT

- **1.1:** retrieve the related physical learning content to store in PF according to its information of learning resource.
- **1.2:** generate the corresponding XML code including <resource>, <file>, etc. to integrate the leaf node and its learning resources.

Step 2: Generate the *manifest* file which describes the structure of course and related learning resources.

Step 3: Package the *manifest* file and **PF** into the CP;

Step 4: Output the **CP**

5.2.7 Example of Object Oriented Course Modeling (OOCM)

In this dissertation, we use the course “Photoshop” as experimental example, which is released by ADL SCORM organization, to show the process of Object Oriented Course Modeling (OOCM).

Figure 5.14.a is the HLPN Model of Photoshop course created by 6 OOATs and Figure 5.14.b shows its corresponding AT structure transformed by PN2AT and AT2CP processes. Its creating steps are described as follows:

Step 1: Select a ***Linear*** $OOAT_1$ for creating a course structure with 4 learning activities (node).

Step 2: Insert a ***Linear*** $OOAT_2$ into *Node* 1 and ***Linear*** $OOAT_3$ into *Node* 2 in $OOAT_1$ for creating the **Course A** and **B**, respectively.

Step 3: Insert a ***Choice*** $OOAT_4$ into *Node* 3 in $OOAT_1$ for creating the **Course C** and then set that the $Test_2$ node will write its testing result into *Global Objective* P_G so the P_W with token C_O enables the T_O to set P_G according to the learning result in P_{LC}..

Step 4: Insert a ***Linear*** $OOAT_5$ into *Node* 4 in $OOAT_1$ and then insert a ***Conditional Linear*** $OOAT_6$ into $OOAT_5$ for creating the **Course D-1**. The $OOAT_6$ will read *Global Objective* P_G and then select different learning activities (T_{A8} or T_{A9}) for learners according to the testing result of **Course C**.

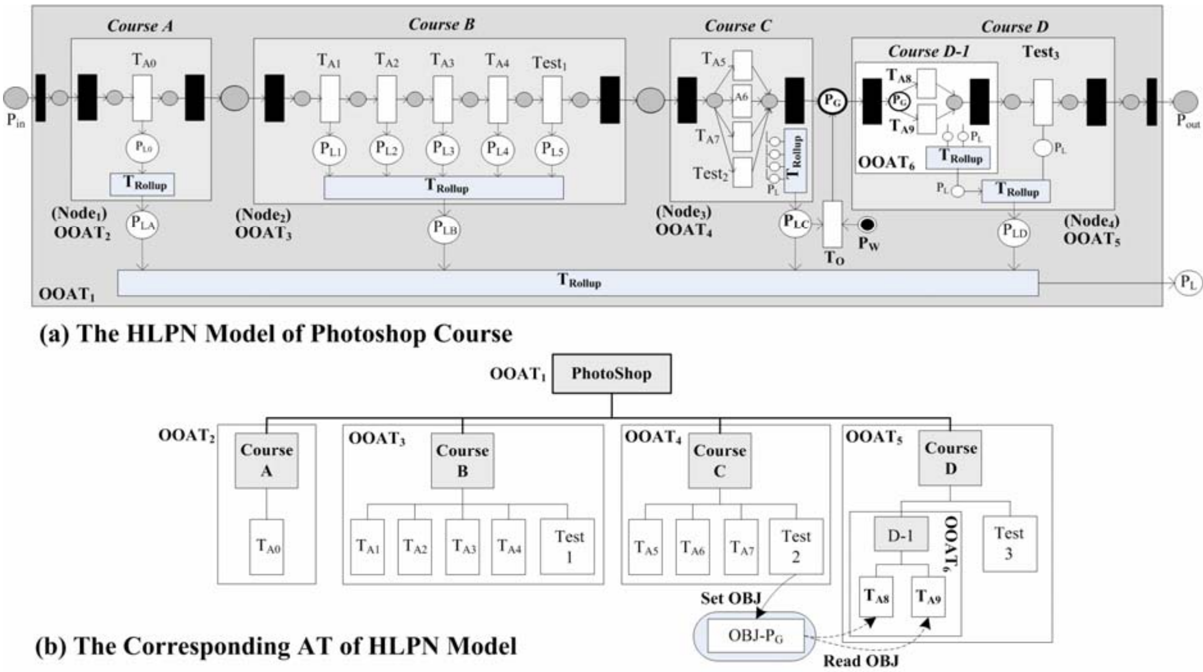


Figure 5.14: The HLPNs Model and AT Structure of Course "PhotoShop"

5.3 Object Oriented Learning Activity Authoring Tool

Based on OOLA model, in this dissertation, we develop an OOLA authoring tool with user-friendly GUI interface. As shown in Figure 5.15, OOLA authoring tool can provide teachers with an efficient learning design environment to create, retrieve, and edit the learning activity, learning objects, application programs, and test sheet. Teachers can use the GUI interface to design the desired learning activity and then the Rule Transformer (RT) will transform the created LA into rule format stored in Learning Activity Repository (LAR). The rules in LAR will be used by Inference Engine (IE), called DRAMA [78], in Knowledge Controller (KC) module to control the learning guidance during the period of learning activity.

Figure 5.15 illustrates the details of OOLA authoring tool which consists of four types of resources in ILCMS, that is, Learning Object Repository (LOR), Testing Item Bank (TIB), Application Program Repository (APR), and Learning Activity Repository (LAR). Therefore, User can upload or edit a SCORM compliant content through LOR. The SCORM standard transformation module can transform the ordinary PowerPoint or HTML files to SCORM compliant content packages [114], the SCORM content uploading module can reuse and share the SCORM compliant contents in the repository, and the learning object manager can provide efficient content searching and browsing services [116]. Moreover, a variety of application programs of ILCMS such as chat room or the URL of existing search engines stored in APR can be provided for learners to enhance the interactive learning and discussions. Regarding the TIB, each testing item stored in TIB is associated to several related learning concepts. Therefore, the learner's learning achievement can be detected. Accordingly, the appropriate learning activity or remedial activity can be provided according learners' learning results. These edited learning activities will be transformed into rule formats by means of the Rule

Transformer (RT) and then stored in LAR.

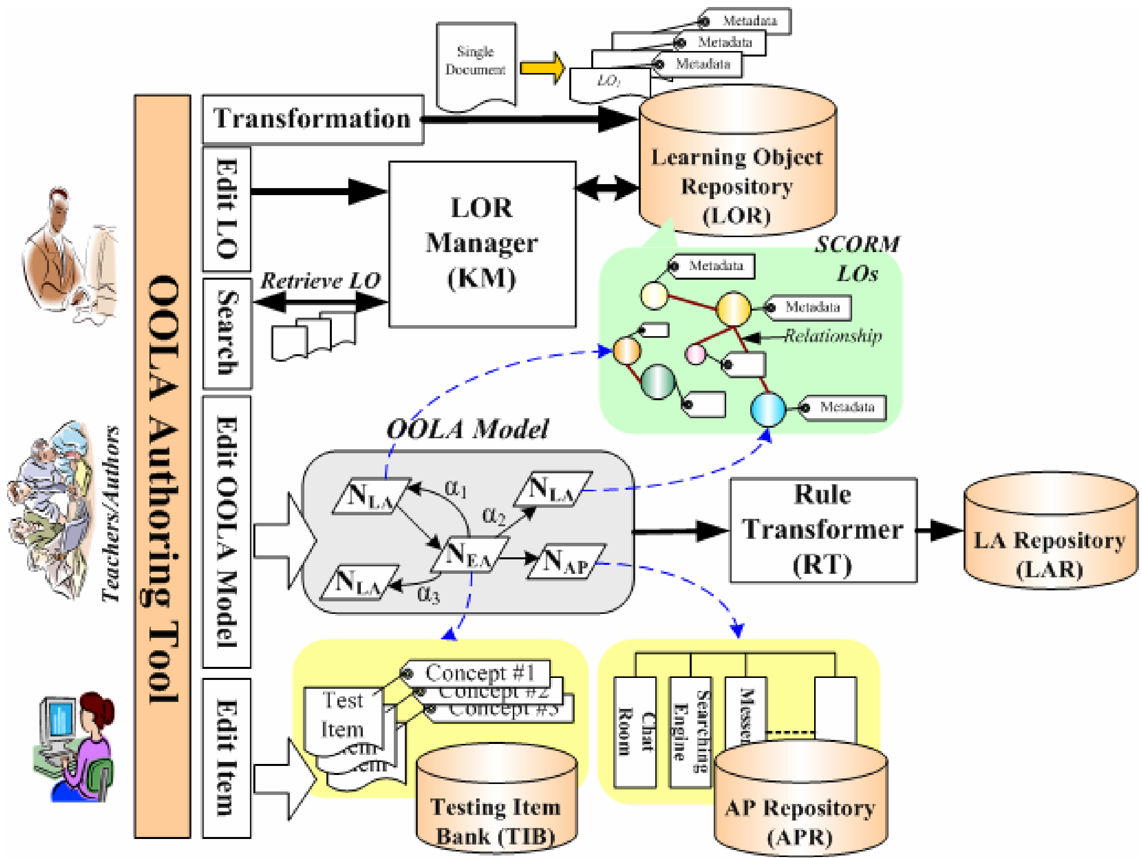


Figure 5.15: The Diagram of OOLA Authoring Tool

5.3.1 Rule Transformation of OOLA Model

Due to the pedagogical needs, the teachers can use OOLA model to edit the learning activity as directed graph that includes learning object, application and assessment quiz. The OOLA is a rule-based model and the rule representation of OOLA model is based on the New Object oriented Rule Model (NORM) architecture [78] [123] [124] [125] [139] which modularizes the rules as rule classes [11] [92]. While the rule set of specific domain is acquired, the rule classes are instantiated as rule objects and can be executed on the NORM inference engine, DRAMA [78]. The "if *condition* then *action*" rule format is compliant with rule format of SCORM Sequencing & Navigation. Therefore, we proposed an OOLA2NORM Algorithm (OOLA Model to

NORM Rule) implemented in Rule Transformer (RT) to transform OOLA model to NORM architecture. The learning achievements are represented as facts and rule conditions are inferred to choose an appropriate activity for student. The transformation algorithm is shown in Algorithm 5.3.

Algorithm 5.3: OOLA Model to NORM Rule (OOLA2NORM)

Definition of Symbols:
C_{ik}: The k-th associated concept of the i-th N_{EA} node
Factnow: The name of current node being studied.
E_{ij}: The edge from N_i to N_j
$Fact_{Nj}$: The next node to be studied, if N_j is satisfied.
$Fact_{Cik}$: The score of C_{ik}
<OP>: The relational operator, i.e., $=, >, <, \leqq, \geqq$ and $\neq$

Input: The XML file of OOLA model
Output: The XML file with a Rule Class of DRAMA Inference Engine

Step 1: Create a **$Fact_{now}$**

Step 2: For each **E_{ij}**$\in$OOLA model

2.1: Create a Fact for **N_j**, called **$Fact_{Nj}$**, which denotes that if **$Fact_{Nj}$** = *true*, then **N_j** is the next node.

2.2: If $N_i \in N_{LA}$ or **N_{AP}**

Then create a rule type: *“if ($Fact_{now}$ = ’N_i’) then $Fact_{Nj}$ = true”*

else if $N_i \in N_{EA}$

then (1) Acquire all concepts **C_{ik}** of **N_i** and conditional threshold (α) which are included in **E_{ij}**

(2) Create the $Fact_{Cik}$ for C_{ik}.

(3) Create a rule type: *“if ($Fact_{now}$ = ’N_i’) and ($\sum Fact_{Cik} <OP> \alpha$) then $Fact_{Nj}$ = true”*

Step 3: Output the XML file of DRAMA

Chapter 6 Knowledge Manager (KM)

In e-learning system, teaching materials are usually stored in database, called Learning Object Repository (LOR). Because the SCORM standard has been accepted and applied popularly, its compliant teaching materials are also created and developed. Therefore, in LOR, huge amount of SCORM teaching materials including associated learning objects (LO) will result in the issues of management. Recently, SCORM international organization has focused on how to efficiently maintain, search, and retrieve desired learning objects in LOR for users. Therefore, in this dissertation, , we propose a new approach, called ***Level-wise Content Management Scheme*** (LCMS) [116], to efficiently maintain, search, and retrieve the learning contents in SCORM compliant LOR and it is implemented within a **Learning Object Repository (LOR) Manager** in **Knowledge Manager (KM)** of ILCMS.

6.1 Level-wise Content Management Scheme (LCMS)

6.1.1 The Processes of LCMS

As shown in Figure 6.1, the scheme of LCMS is divided into C*onstructing Phase* and S*earching Phase*. The former creates the content tree form SCORM content package by CP2CT process, and then creates and maintains a multistage graph as Directed Acyclic Graph (DAG) with relationships among LOs, called Level-wise Content Clustering Graph (LCCG), by applying clustering techniques. The latter traverses the LCCG by LCCG Content Searching Algorithm (LCCG-CSAlg) to retrieve desired learning content with general and specific LOs according to the query of users over wire/wireless environment.

Constructing Phase includes the following three processes:

- **Content Package to Content Tree (CP2CT) Process:** it transforms the content structure of SCORM teaching materials (Content Package) into tree-like structure with the representative feature vector and the same depth, called Content Tree (CT), for representing each teaching materials.
- **Level-wise Content Clustering Process:** it clusters LOs according to content trees (CTs) to establish the *level-wise content clustering graph* (LCCG) for creating the relationships among LOs .
- **LCCG Maintaining Process:** it monitors the condition of each node within LCCG and to rebuild the LCCG if necessary.

Searching Phase includes the following two processes:

- **SCORM Metadata Searching:** it first searches the desired whole teaching materials by the associated SCORM metadata for addressing the related nodes as entries of LCCG.
- **Level-wise Content Searching:** it then traverses the LCCG from these entry nodes to retrieve the more precise learning objects in LOR and to deliver these for learners.

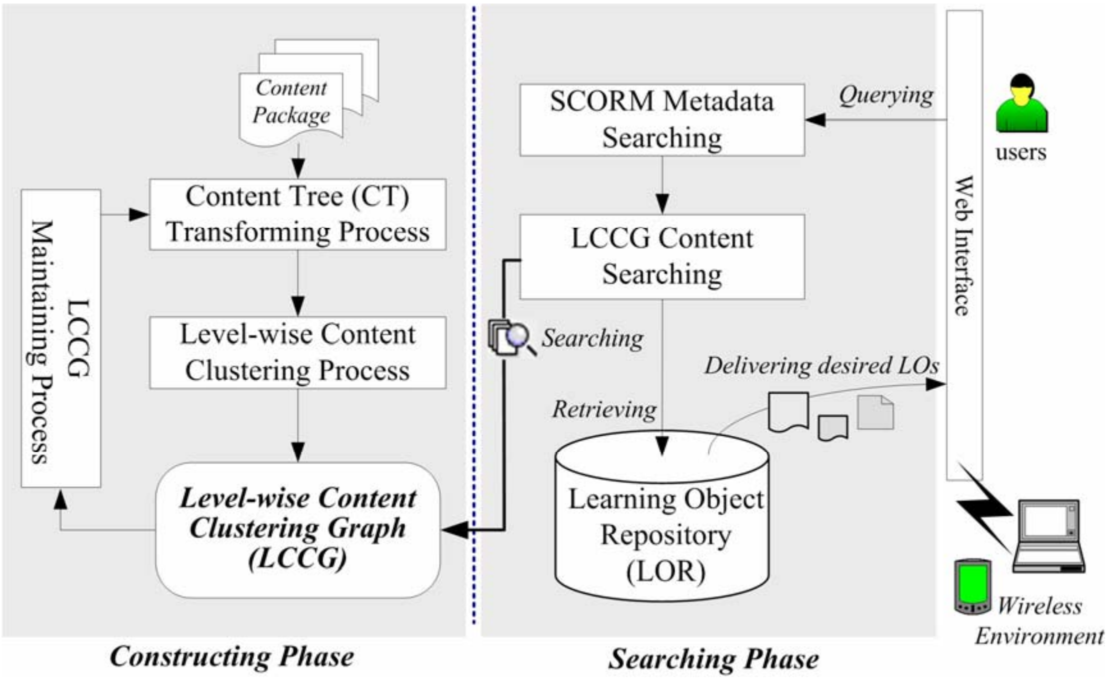


Figure 6.1: The Flowchart of Level-wise Content Management Scheme (LCMS)

6.1.2 Content Package to Content Tree (CP2CT) Process

Because we want to create the relationships among LOs according to the content structure of teaching materials, the organization information in SCORM content package will be transformed into a tree-like representation with representative feature vector, called ***Content Tree*** (CT). For clustering process conveniently, the depth of every CT is the same. Its definition is described as follows.

Definition 6.1: Content Tree (CT) = (*N*, *E*), where

- $\boldsymbol{N} = \{ n_0, n_1, \ldots, n_m \}$.
- $\boldsymbol{E} = \{ \overrightarrow{n_i n_{i+1}} \mid 0 \leqq i < \text{the depth of } \boldsymbol{CT} \}$.

In CT, each node is called "*Content Node* (**CN**)" containing a *feature vector* $\vec{V}$ which denotes the representative feature of learning contents within this node. **E** denotes the link edges from node n_i in upper level to n_{i+1} in next lower level.

In this dissertation, we apply the **Vector Space Model** (**VSM**) approach [24] [96] to represent the learning contents in CN. Thus, based upon the *Term Frequency* - *Inverse Document Frequency* (**TF-IDF**) weighting scheme [2] [30] [68] [106] [136], each CN can be represented by an N dimensions vector as $<tf_1 \times idf_1, tf_2 \times idf_2, \ldots, tf_n \times idf_n>$, where tf_i is the frequency of the *i-th* term (keyword) and $idf_i = \log(m/df(t))$ is the *Inverse Document Frequency* (*IDF*) of the *i-th* term in the document (where m is total number of documents and $df(t)$ is the number of documents that contains the term).

For conveniently creating the relationships among learning objects according to the content structure, we assume that every content tree (CT) transformed from content package will have the same depth of tree. However, in many teaching materials, the depths of content structures are different. Therefore, in CT, if the depth of a leaf CN is too short, the ***Virtual Node*** (**VN**) will be repeatedly inserted as its child node until the difference of the desired depth has been filled. The feature vector of every VN is the same as its parent CN or VN. Besides, if the depth of a leaf CN is too long, its parent CN in the desired depth will merge the information of all included child nodes into one new CN whose feature vector is generated by averaging these included child nodes.

The Example 6.1 shows the process of transforming the organization information of SCORM content package into Content Tree (CT) with the feature vector $\vec{V}$ and the same depth.

Example 6.1:

Given a SCORM content package shown in the left side of Figure 6.2, we take *TF-IDF* as weighting scheme to create the *feature vector* $\vec{V}$ in each CN node. Because the depth of CN, "**Chapter 1**", is too short, the VN named "**1.1**" is inserted and its feature vector $\vec{V}_{21}$=<3, 2, 2> is the same as $\vec{V}_{11}$. Moreover, the CN, "3.1", is too long, so that its included child nodes, i.e., "3.1.1" and "3.1.2", are merged into one CN, "3.1", and their feature vector $\vec{V}_{24}$ is the average of <1,0,1> and (<2,1,0>+<0,1,2>)/2 after the rolling up process. Then, The CT after CP2CT Process is shown in the right part of Figure 6.2.

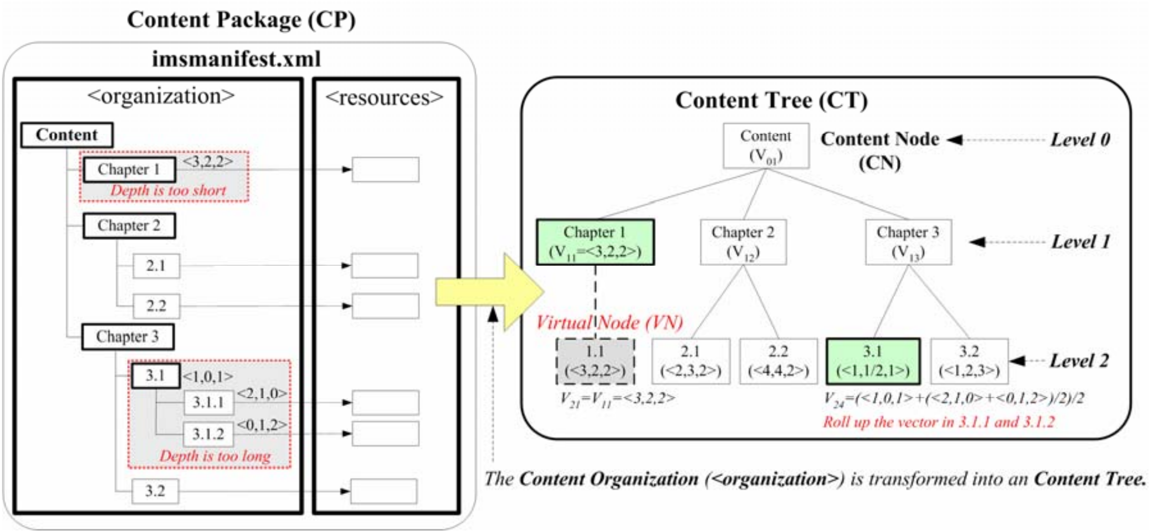


Figure 6.2: The Corresponding Content Tree (CT) of the Content Package (CP) by CP2CT process

Algorithm 6.1: Content Package to Content Tree Algorithm (CP2CTAlgo)

Symbols Definition:

CP: denote the SCORM content package.

CT: denote the Content Tree transformed the CP.

CN: denote the Content Node in CT.

CN_{leaf}: denote the leaf node CN in CT.

D_{CT}: denote the desired depth of CT.

D_{CN}: denote the depth of a CN

Input: SCORM content package (CP)

Output: Content Tree (CT) with feature vector

Step 1: For each element **<item>** in CP

  1.1: Create a CN with feature vector based upon TF-IDF weighting scheme.

  1.2: Insert it into the corresponding level in **CT**.

Step 2: For each CN_{leaf} in CT

  If the depth of CN_{leaf} **<** $\mathbf{D_{CT}}$

   Then a **VN** will be repeatedly inserted as its child node until the depth of CN_{leaf} **=** $\mathbf{D_{CT}}$.

   Else If the depth of CN_{leaf} **>** $\mathbf{D_{CT}}$

    Then its parent CN in depth **=** $\mathbf{D_{CT}}$ will merge the information of all included child nodes and run the rolling up process to average their feature vectors.

Step 3: Content Tree (CT) with feature vector

6.1.3 Level-wise Content Clustering Process

After transforming the organization information of content package into content tree (CT), the clustering technique can be applied to create the relationships among content nodes (CNs) in CT. Thus, in this dissertation, we propose a **Level-wise Content Clustering Graph**, called **LCCG**, to store the related information of each cluster. Based upon the LCCG, the desired learning content including general and specific LOs can be retrieved for users.

Level-wise Content Clustering Graph (LCCG):

LCCG is a multistage graph with relationships information among LOs, e.g., Directed Acyclic Graph (DAG). Its definition is described as follows:

Definition 6.2: Level-wise Content Clustering Graph (LCCG) = (N, E), where

- **N**={(CF_0,CL_0), (CF_1,CL_1),…, (CF_m,CL_m)}. It stores the related information, ***Cluster Feature*** (**CF**) and ***Child List*** (**CL**), in a cluster, called ***LCC-Node***. The **CL** stores the CF value of included child LCC-Nodes in next stage.
- **E**={$\overrightarrow{n_i n_{i+1}}$ | $0 \leqq i <$ *the depth of LCCG*}. It denotes the link edge from node n_i in upper stage to n_{i+1} in next lower stage.

For the purpose of content clustering, the number of the stages of LCCG is equal to the depth of CT, and each stage handles the clustering result of these CNs in the corresponding level of different CTs. That is, the top/ lowest stage of the LCCG stores the clustering results of the root/leaf nodes in the CTs, respectively. In addition, in LCCG, the ***Cluster Feature (CF)*** stores the related information of a cluster. It is similar with the *Cluster Feature* proposed in the *Balance Iterative Reducing and Clustering* *using Hierarchies* (**BIRCH**) [145] clustering algorithm and defined as follows.

Definition 6.3: The **Cluster Feature (CF)** of a cluster is defined as a triple: **CF=(N, $\overrightarrow{VS}$, CS)**, where

- **N:** it denotes the number of the content nodes (CNs) in a cluster.
- $\overrightarrow{VS} = \sum_{i=1}^{N} \vec{V}_i$. It denotes the sum of feature vectors ($\vec{V}$) of CNs.
- **CS**=$|\sum_{i=1}^{N} \vec{V}_i / N| = |\overrightarrow{VS} / N|$. It denotes the average value of the feature vector sum in a cluster. The **| |** denotes the *Euclidean distance* of the feature vector. The ($\overrightarrow{VS}$ /**N**) can be seen as the **Cluster Center (CC)** of a cluster.

Moreover, during content clustering process, if a content node (CN) in content tree (CT) with feature vector ($\vec{V}$) is inserted to the cluster **CF$_A$=(N$_A$, $\overrightarrow{VS}_A$, CS$_A$)**, the new CF$_A$**=(N$_A$+1, $\overrightarrow{VS}_A + \vec{V}$, |($\overrightarrow{VS}_A + \vec{V}$)/(N$_A$+1)|).** An example of Cluster Feature (CF) and Child List (**CL**) is shown in Example 6.2.

Example 6.2:

Assume a cluster **C$_0$** stores in the LCC-Node **N$_A$** with (CF$_A$, CL$_A$) and contains four CNs, which include four feature vectors, <3,3,2>, <3,2,2>, <2,3,2> and <4,4,2>, respectively. Then, the $\overrightarrow{VS}$ =<12,12,8>, the **CC**= $\overrightarrow{VS}$ **/4=**<3,3,2>, and the **CS**=$\sqrt{9+9+4} = 4.69$. Thus, the **CF$_A$** = (4, <12,12,8>, 4.69). Moreover, assume the **CL$_A$** = <CF$_1$, CF$_2$>. A new Content Node **CN B** with feature vector $\vec{V}_B$ = <8,3,2> is inserted to the cluster **C$_0$** in **N$_A$**. The child nodes of **CN B** belong to the clusters **C$_3$** and **C$_4$** respectively. Then, the new **CF$_A$** = (5, <20,15,10>, 5.385) and **CL$_A$** = <CF$_1$, CF$_2$, CF$_3$, CF$_4$>.

Level-wise Content Clustering Algorithm (LCCAlg):

Based upon the definition of LCCG, we propose a **Level-wise Content Clustering Algorithm,** called **LCCAlg**, to create the LCCG according to the CTs transformed from CPs. The **LCCAlg** includes three phases: 1) *Single Level Clustering Phase*, 2) *Content Cluster Refining Phase*, and 3) *Concept Generalizing Phase*. Figure 6.3 illustrates the flowchart of **LCCAlg.**

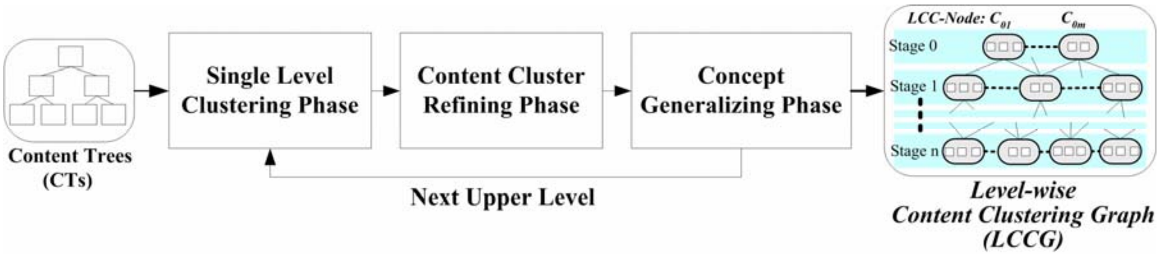


Figure 6.3: The Flowchart of Level-wise Content Clustering Algorithm (LCCAlg)

(1) ***Single Level Clustering Phase:***

In this phase, the content nodes (CNs) of CT in each tree level can be clustered by different similarity threshold. The content clustering process is started from the lowest level to the top level in CT. All clustering results are stored in the LCCG. In addition, during content clustering process, the similarity measure between two *CNs* is defined by the *cosine* function which is the most common for the document clustering [101] [134]. It means that, given two CN N_A and N_B, the similarity measure is calculated by

$$Similarity = \text{cosine}(V_A, V_B) = \frac{V_A \bullet V_B}{|V_A||V_B|}$$

, where V_A and V_B are the *feature vectors* of N_A and N_B respectively. The larger the value is, the more similar two vectors are. For example, two CNs are most similar, the cosine value of their feature vectors is equal to 1. The *Single Level Clustering Algorithm* (**SLCAlg**) is shown in Algorithm 6.2.

Algorithm 6.2: Single Level Clustering Algorithm (SLCAlg)

Symbols Definition:

CN_{set}: the content nodes (CNs) in the same level (**L**) of content trees (CTs).

T : the similarity threshold for clustering process.

Input: CN_{set} and **T.**

Output: The set of LCC-Nodes storing the clustering results of CTs.

Step 1: insert a CN node $n_0 \in$ **CN_{set}** into a cluster in the LCC-Node.

Step 2: $\forall$ $n_i \in$ **CN_{set}**.

  2.1: If $\exists$ a cluster with similarity value > T

    Then insert the n_i into this cluster and update the related **CF** and **CL** in LCC-Node.

   Else insert the n_i into a new cluster stored in a new LCC-Node.

Step 3: Return the set of the LCC-Nodes.

(2) Content Cluster Refining Phase:

Due to the SLCAlg algorithm which runs the clustering process by inserting the content trees (CTs) incrementally, the content clustering results are influenced by the inputs order of CNs. In order to reduce the effect of input order, the **Content Cluster Refining Phase** is necessary. Given the content clustering results of SLCAlg, Content Cluster Refining Phase utilizes the *cluster centers* of original clusters as the inputs and runs the single level clustering process again for modifying the accuracy of original clusters. Moreover, the similarity of two clusters can be computed by the Similarity Measure as follows:

$$Similarity = Cos(CC_A, CC_B) = \frac{CC_A \bullet CC_B}{|CC_A||CC_B|} = \frac{(\overrightarrow{VS_A}/N_A) \bullet (\overrightarrow{VS_A}/N_A)}{CS_A * CS_B}$$

After computing the similarity, if the two clusters have to be merged into a new cluster, the new **CF** of this new cluster is: **CF_{new}= (N_A+N_B,** $\overrightarrow{VS}_A + \overrightarrow{VS}_B$ **,** **$|(\overrightarrow{VS}_A+\overrightarrow{VS}_B)/(N_A+N_B)|$).**

(3) Concept Generalizing Phase:

The concept generalization phase is used to make the feature vectors of CNs of internal LCC-Nodes in LCCG more objective and representative. Thus, we propose a *roll-up operation* to compute the feature vectors of CNs by averaging the cluster centers of the content clusters which their included child CNs belong to.

The **Level-wise Content Clustering Algorithm (LCCAlg)** is shown in Algorithm 6.3. Moreover, an example of creating Level-wise Content Clustering Graph (LCCG) is also described in Example 6.3.

Example 6.3:

As shown in the left part of Figure 6.4, assume that there are two content trees CT_A and CT_B. The content nodes CN A and CN B belong to the cluster C_{01}, and their included child CNs belong to the C_{11}, C_{12}, and C_{13}. After Level-wise Content Clustering Process, the LCCG is showed in the right part of Figure 6.4. The LCCG-Node C_{01} in stage S_0 contains two CNs (CN A and CN B) and three included child LCCG-Nodes, C_{11}, C_{12}, and C_{13}, in stage S_1. Moreover, in Figure 6.4, the CN A with feature vector <1, 1, 2> contains 2 child CNs where A_0 in cluster C_{11} and A_1 in cluster C_{12}. The cluster centers (CC) of C_{11} and C_{12} are <3, 3, 2> and <3, 2, 4>, respectively. Then, after running roll-up operation, the new feature vector of the CN A is: *Average*((<3,3,2>+<3,2,4>)/2+<1,1,2>) = <2, 7/4, 5/2>.

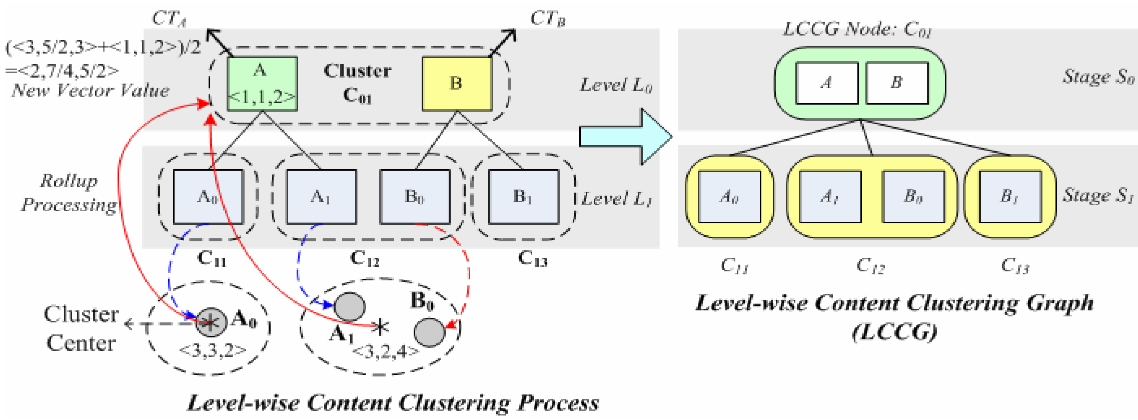


Figure 6.4: An Example of Creating Level-wise Content Clustering Graph (LCCG)

Algorithm 6.3: Level-wise Content Clustering Algorithm (LCCAlg)

Symbols Definition:

D: is the depth of the content tree (CT).

L_0~L_{D-1}: denote the levels of CT descending from the top level to the lowest level.

S_0~S_{D-1} : denote the stages of LCCG.

T_0~T_{D-1}: denote the similarity thresholds for clustering the CNs in the level L_0~L_{D-1} respectively.

CT_{set}: the set of content trees (CTs) with the same depth (***D***).

CN_{set}: the content nodes (CNs) in the same tree level (***L***).

Input: CT_{set}

Output: LCCG which holds the clustering results in every content tree level.

Step 1: For *i* = L_{D-1} to L_0, do the following Step 2 to Step 4.

Step 2: Single Level Clustering:

Step 2.1: CN_{set} = the CNs ∈ **CT_{set}** in **L_i**.

Step 2.2: Run ***Single Level Clustering Algorithm*** (**SLCAlg**) for **CN_{set}** with threshold **T_i**.

Step 3: Content Cluster Refining:

Step 3.1: Execute the following sub-Steps (3.2-3.4) repeatedly until there is no difference between two iterations.

Step 3.2: CN_{set}= the nodes with cluster center (CC) ∈ the set of LCC-Nodes in ***S_i***.

Step 3.3: Run the **SLCAlg** for **CN_{set}** with threshold **T_i**.

Step 3.4: Store the resulted clusters in LCC-Nodes of LCCG in stage S_i.

Step 4: Concept Generalizing:

Step 4.1: If $i \neq L_0$

Then Run *roll-up operation* to compute the *feature vectors* of CNs from the level L_{i-1}

Step 5: Output the **LCCG**

6.1.4 LCCG Maintaining Process

As mentioned above, every SCORM Content Package (CP) will be transformed into Content Tree (CT) with the representative feature vector for representing each teaching materials. Because the feature vector is computed based upon the *Term Frequency - Inverse Document Frequency* (**TF-IDF**) weighting scheme [2] [30] [106], a set of all keywords, called **KeywordSet**, has to be integrated from the *activity metadata* of *item* in content package. However, for incrementally updating the learning content in LOR, the keywords within the new SCORM content package may be ***Partially*** or ***Not*** included in the **KeywordSet**, which results in their feature vectors are not accurate.

Therefore, in this dissertation, we propose *LCCG Maintaining Algorithm* (**LCCG-MAlg**), which rebuilds the LCCG if necessary by monitoring the condition of each node within LCCG, to solve above issue. In LCCG-MAlg, a content node (CN) is defined into three types: 1) **CN:** denotes its keywords which are all in **KeywordSet**, and 2) **Partial CN (PCN):** denotes its keywords which are partial in **KeywordSet**, and 3) **New CN (NCN):** denotes its keywords which are not in **KeywordSet**. During **Level-wise Content Clustering Process,** the CN and PCN can be inserted into a suitable cluster stored in LCC-Node but the NCN will be inserted in a new cluster stored in **LCC-Node$_{\mathbf{new}}$**. Moreover, we also define a cluster type call "*Saturation*" to denote that the amount of PCNs is larger than that of CNs in the same cluster.

For checking when to recreate the **KeywordSet** and LCCG, we define the following two *rebuilding conditions*:

(1) The amount of clusters with "*Saturation Tag*" is larger than that of clusters.

(2) The amount of new clusters stored in **LCC-Node$_{\mathbf{new}}$** is larger than that of clusters.

Therefore, for every stage in LCCG, if any of two rebuilding conditions is satisfied, the **KeywordSet** and LCCG will be recreating. An example is given to illustrate the

LCCG Maintaining Process in Figure 6.5. The CN A_1 and CN A_2 in new CT_A are inserted into the C_1 in LCC-Node and C_2 in LCC-$Node_{new}$, respectively. Thus, the inserted CN A_1 results in that the C_1 is marked with the Saturation Tag and the *num*(*LCC-Node* with *Saturation Tag*) is larger than *num*(*LCC-Node*) in stage S_1.

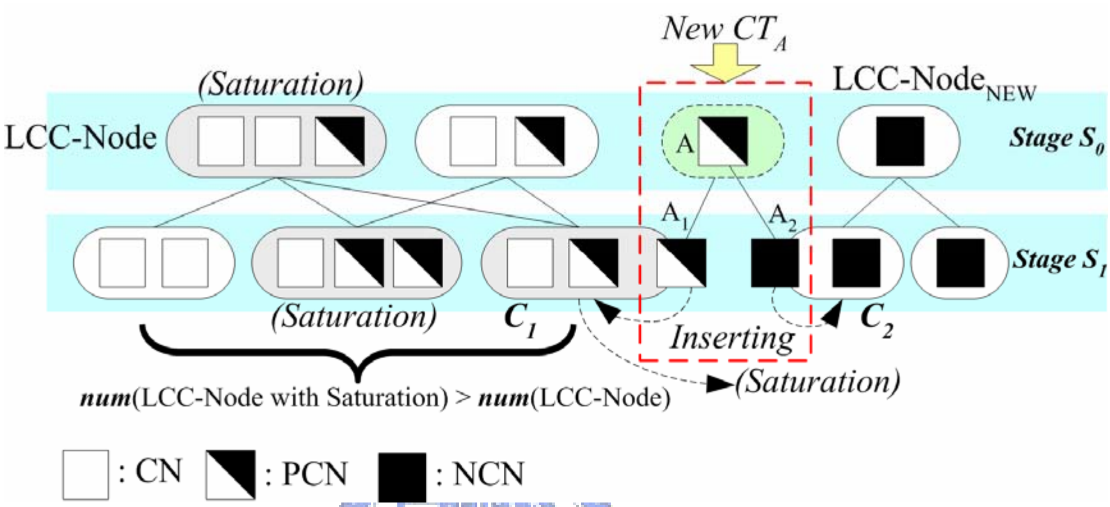


Figure 6.5: An Example of LCCG Maintaining Process

Algorithm 6.4: LCCG Maintaining Algorithm

Symbols Definition:

KeywordSet: the set of keywords in original LCCG used to create *feature vector*.

CT_{set}: the set of content trees (CTs) with the same depth ***D***.

CN: the content node in CT whose keywords are all in **KeywordSet.**

PCN: the *partial* content node in CT whose keywords are partial in **KeywordSet.**

NCN: the *new* content node in CT whose keywords are not in **KeywordSet.**

***$LCC\text{-}Node_{new}$*:** it stores the NCN.

Input: CT_{set}

Output: A new LCCG.

Step 1: during **Content Tree Transforming Process**, mark the content nodes in CT_{set} as ***CN***, ***PCN***, or ***NCN***.

Step 2: during **Level-wise Content Clustering Process,**

  2.1: For each ***node*** in **CT**

    If ***node***= ***CN or PCN*** **Then** insert it into a suitable cluster stored in *LCC-Node*.

    If ***node***= ***NCN*** **Then** insert it into a cluster stored in $LCC\text{-}Node_{new}$.

  2.2: If *num*(*NPN*) > *num*(*CN*) in a *LCC-Node*

    Then mark the *LCC-Node* with *Saturation Tag*.

Step 3: for every stage in LCCG,

  3.1: If (*num*(*LCC-Node* with *Saturation Tag*) >*num*(*LCC-Node*)) or

    (*num*($LCC\text{-}Node_{new}$) > *num*(LCC-Node))

Then re-execute the ***Constructing Phase*** in LCMS for creating new **KeywordSet** and new LCCG.

6.2 Searching Process of LCMS

In this section, we describe the searching process of LCMS, which includes *SCORM Metadata Searching* and *LCCG Content Searching*, shown in the right part of Figure 6.1.

6.2.1 SCORM metadata Searching

As mentioned in Chapter 4, the SCORM compliant teaching materials include 4 parts: 1) **Metadata**, 2) **Organizations**, 3) **Resources**, and 4) **(Sub) Manifest**. Here, Metadata, which is referred from the IEEE's Learning Objects Metadata (LOM), describes the characteristic or attribute of the teaching materials. The LOM describes learning resources including nine categories: 1) **General:** describes the general information of learning resource, 2) **LifeCycle:** describes the history and current state of learning resource and its evolution information, 3) **Meta-MetaData:** describes the specific information about the metadata record itself, 4) **Technical:** describes the technical requirements and characteristics of learning resource, 5) **Educational:** describes the key educational or pedagogic characteristics of learning resource, 6) **Rights:** describes the intellectual property rights and conditions of use for learning resource, 7) **Relation:** defines the relationships among this resource and other targeted resource, 8) **Annotation:** provides comments on the educational use of learning resource, and 9) **Classification:** describes classification criteria and hierarchy of learning resource.

Therefore, as shown in Figure 6.6, the desired whole teaching materials in learning object repository (LOR) can be retrieved by the associated SCORM metadata first for addressing the related LCC-Nodes as entries of LCCG and then according to the entry LCC-Nodes, e.g., C_{0m}, the more precise learning objects (LOs) of retrieved teaching materials will be further searched by *LCCG Content Searching* (described later) based upon LCCG.

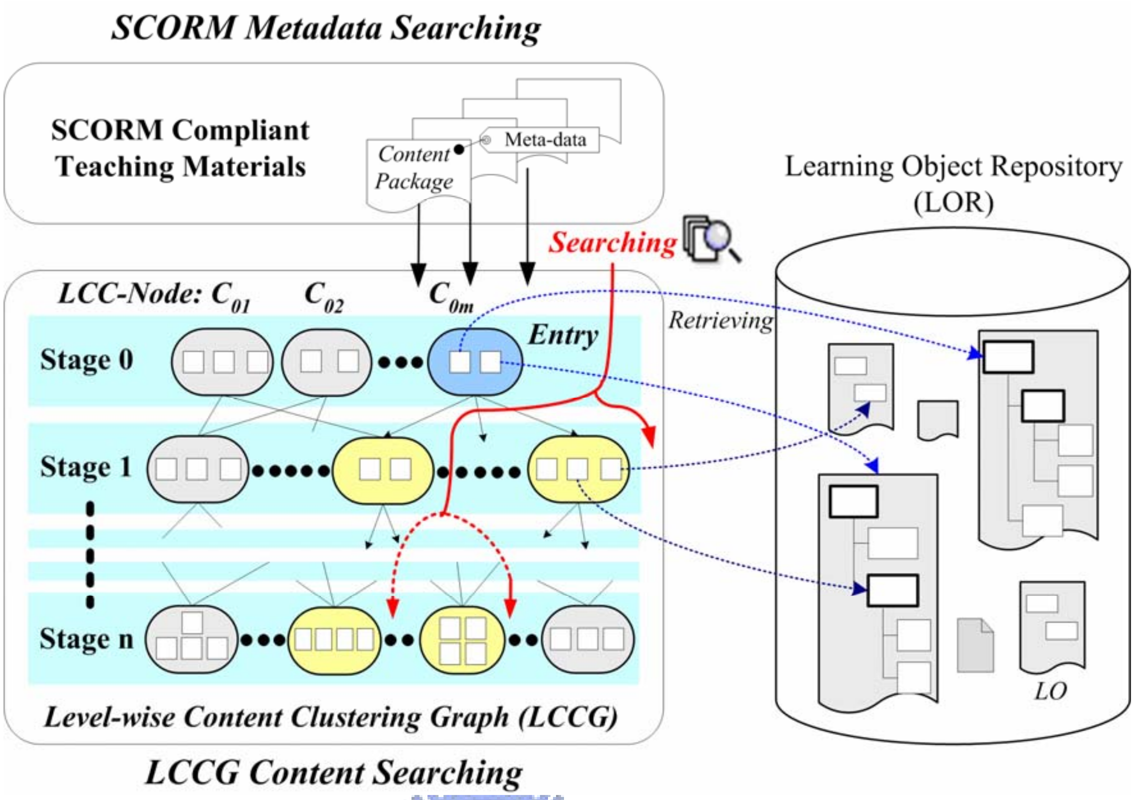


Figure 6.6: The Searching Process in LCMS

6.2.2 LCCG Content Searching

In LCCG, every LCC-Node contains several similar content nodes (CNs) in different content trees (CTs) transformed from content package of SCORM compliant teaching materials. The content within LCC-Nodes in upper stage is more general than the content in lower stage. Therefore, based upon the LCCG, users can get their interesting learning contents which contain not only general concepts but also specific concepts. The interesting learning content can be retrieved by computing the similarity of cluster center (CC) stored in LCC-Nodes and the query vector. If the similarity of LCC-Node satisfies the query threshold users defined, the information of learning contents recorded in this LCC-Node and its included child LCC-Nodes are interested for users. Moreover, we also define the *Near Similarity Criterion* to decide when to stop the searching process. Therefore, if the similarity between the query and the LCC-Node in the higher stage satisfies the definition of *Near Similarity Criterion*, it is not necessary to search its included child LCC-Nodes which may be too specific to use for users. The *Near Similarity Criterion* is defined as follows:

Definition 6.4: Near Similarity Criterion

Assume that the similarity threshold ***T*** for clustering is less than the similarity threshold ***S*** for searching. Because similarity function is the *cosine* function, the threshold can be represented in the form of the angle. The angle of ***T*** is denoted as $\theta_T = \cos^{-1} T$ and the angle of ***S*** is denoted as $\theta_S = \cos^{-1} S$. When the angle between the query vector and the cluster center (CC) in LCC-Node is lower than $\theta_S - \theta_T$, we define that the LCC-Node is *near* similar for the query. The diagram of Near Similarity is shown in Figure 6.7.

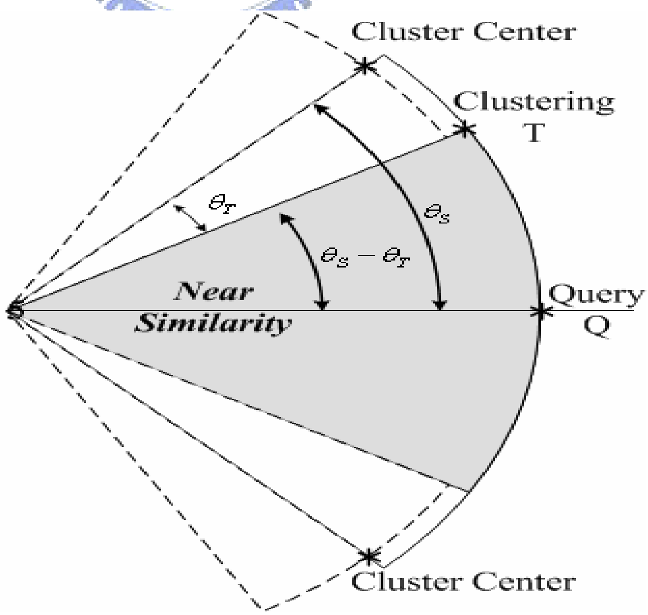


Figure 6.7: The Diagram of Near Similarity According to the Query Threshold Q and Clustering Threshold T

In other words, *Near Similarity Criterion* is that the similarity value between the query vector and the cluster center (CC) in LCC-Node is larger than $Cos(\theta_S - \theta_T)$, so that the *Near Similarity* can be defined again according to the similarity threshold **T** and **S**.

$$\begin{aligned}\text{Near Similarity} &> Cos\,(\theta_S - \theta_T) = Cos\,\theta_S Cos\,\theta_T + Sin\,\theta_S Sin\,\theta_T \\ &= \mathrm{S} \times \mathrm{T} + \left(\sqrt{1 - S^2}\right)\left(\sqrt{1 - T^2}\right)\end{aligned}$$

By the *Near Similarity Criterion*, the algorithm of the LCCG Content Searching Algorithm (LCCG-CSAlg) is proposed as follows.

Algorithm 6.5: LCCG Content Searching Algorithm (LCCG-CSAlg)

Symbols Definition:

Q: is the *query vector* whose dimension is the same as the *feature vector* of content node (CN)

D: is the number of the stage in an LCCG.

S_0~S_{D-1}: denotes the stage of an LCCG from the top stage to the lowest stage.

ResultSet, **DataSet**, and **NearSimilaritySet:** denote the sets of LCC-Nodes.

Input: The query vector **Q**, search threshold **T** and the destination stage S_{DES} where $S_0 \leq S_{DES} \leq S_{D-1}$.

Output: the **ResultSet** contains the set of similar clusters stored in LCC-Nodes.

Step 1: Initiate the **DataSet**= ϕ and **NearSimilaritySet** = ϕ.

Step 2: For each stage **S_i** ∈**LCCG**, repeatedly execute the following steps until **S_i**≧S_{DES}

  2.1: DataSet = **DataSet** ∪ LCC-Nodes in stage **S_i**, and **ResultSet**= ϕ.

  2.2: For each ***N_j*** ∈ **DataSet**,

   { **If *N_j*** is near similar with **Q**

   Then insert ***N_j*** into **NearSimilaritySet**.

   Else If (the similarity between ***N_j*** and **Q)** ≥ **T**

    Then insert ***N_j*** into **ResultSet**. }

  2.3: DataSet = ResultSet. *//for searching more precise LCC-Nodes in next stage in LCCG*

Step 3: Output the **ResultSet = ResultSet** ∪ **NearSimilaritySet**.

Chapter 7 Knowledge Controller (KC)

When the learners login to start learning, the **Knowledge Controller Module** of ILCMS is responsible for initiating a learning activity from the LAR and delivering the suitable learning contents and activities to learners according to their learning results. Therefore, a Learning Activity Controller (LAC) module implemented in KC module, including **System Coordinator (SC)** and **Inference Engine (IE)** [78], will retrieve the appropriate learning objects in LOR, testing sheets in Testing Item Bank (TIB), or application programs (AP) in APR according the personalized learning activity in LAR. Then, it delivers them to learners for adaptive learning with teaching strategy.

7.1 The Rule Inference Process in KC Module

Due to the pedagogical needs, the teachers can use OOLA model to edit the learning activity as directed graph that includes learning objects, applications and assessment quizzes. The OOLA is a rule-based model and the rule representation of OOLA model is based on the New Object oriented Rule Model (NORM) architecture [78] [123] [124] [125] [139] which modularize the rules as rule classes [11] [92]. While the rule set of specific domain is acquired, the rule classes are instantiated as rule objects and can be executed on the NORM inference engine, DRAMA [78].

Figure 7.1 illustrates a leaning process and associated rule inference process. While an OOLA based leaning activity starts, System Coordinator (CO) in LAC will load a suitable OOLA model and then Inference Engine (IE) will infer a suitable learning node as next node for learning according to the rule definitions within OOLA and learners' learning results after learners finished each node or time of node ran out. The SC will give learners a learning content or test sheet, or run an application program as learning service according to different node types which is selected by IE.

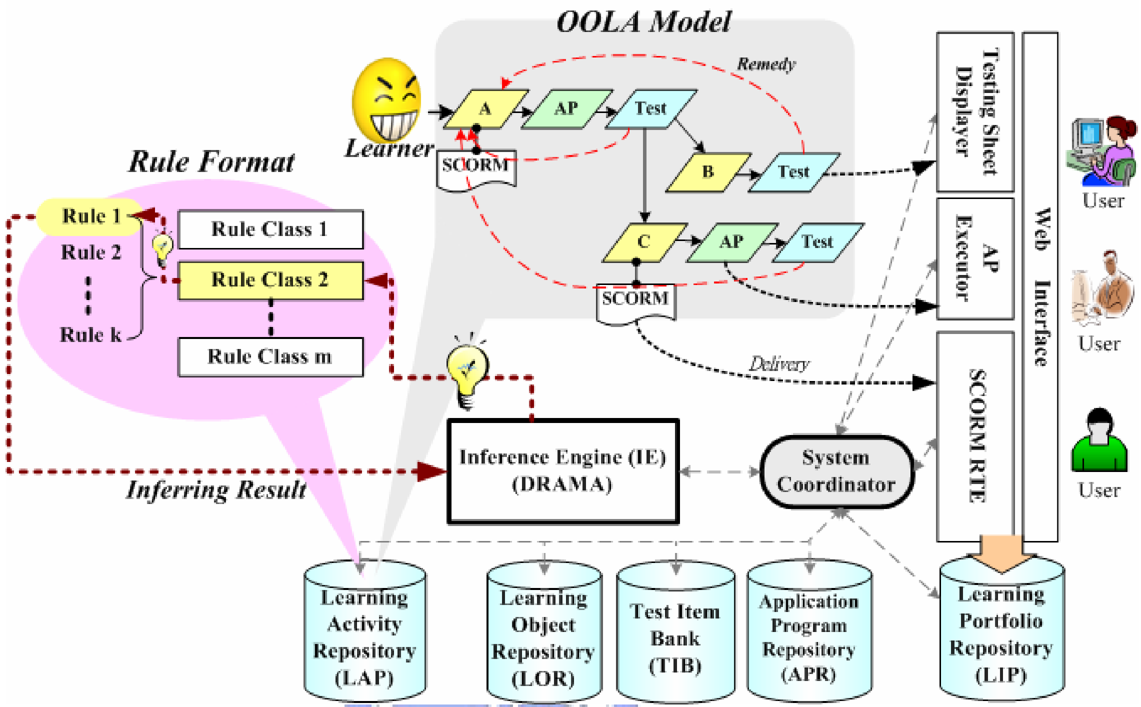


Figure 7.1: The Diagram of Rule Inference Process in KC Module

7.2 The Learning Process of OOLA based Learning Activity

As stated previously, in KA, System Coordinator (SC) is responsible for communicating with IE and controlling all related system modules to execute their jobs, such as displaying learning content or running application program. Accordingly, in this dissertation, we propose a learning process algorithm of running OOLA, as shown in Figure 7.2.

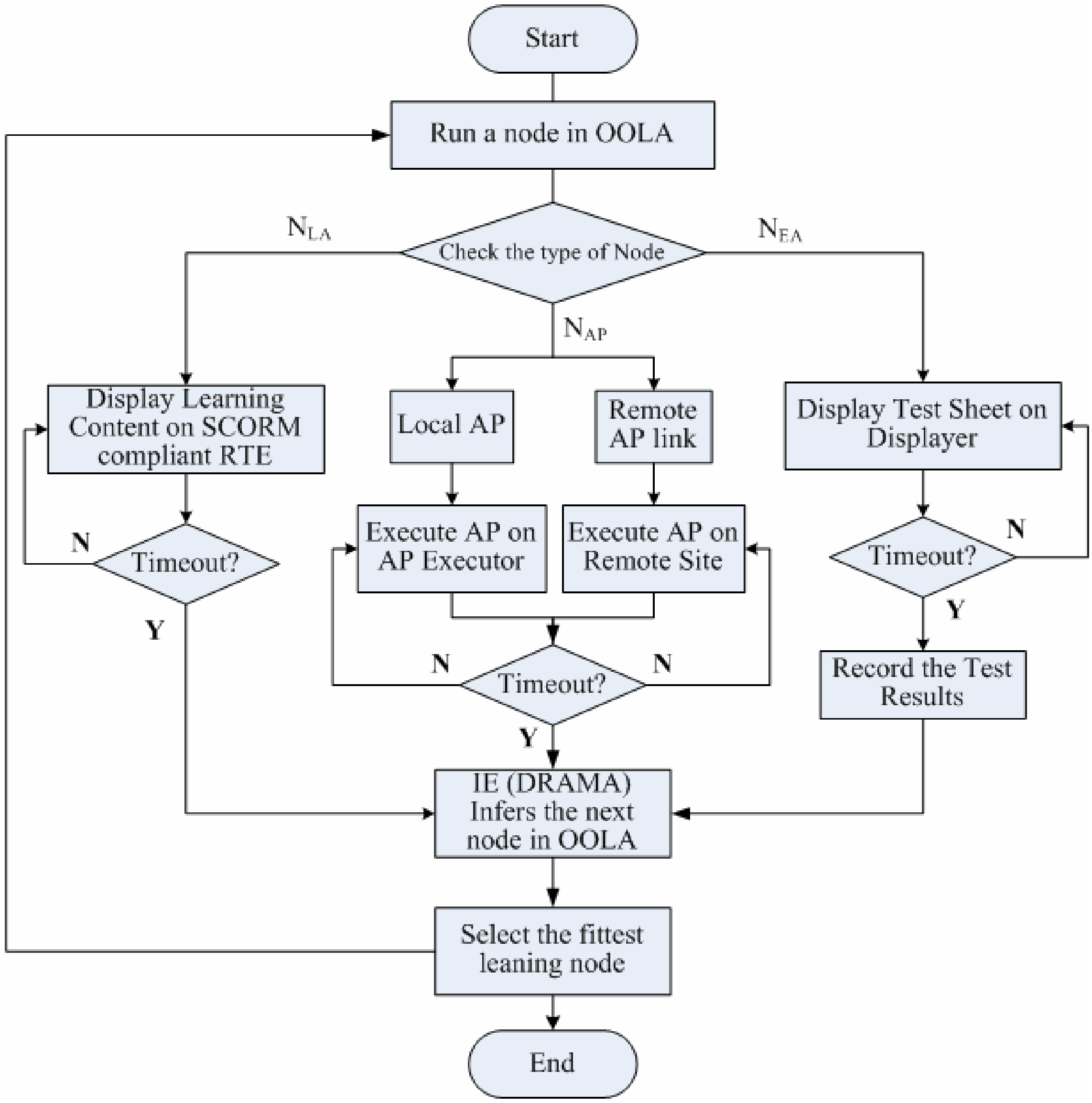


Figure 7.2: The Learning Process of Running OOLA Model

Chapter 8 Knowledge Miner (KMin)

Knowledge Miner Module includes a **Learning Portfolio Analyzer (LPA)**, which consists of **Learning Portfolio Mining (LPM)** [118] and **Two-Phase Concept Map Construction (TP-CMC)** [110] algorithm. According to the learners' characteristics, the former applies the clustering and decision tree approach to analyze the learning behaviors of learners with high learning performance. The latter applies Fuzzy Set Theory and Data Mining approach to automatically construct the concept map by learners' historical testing records. Therefore, after the learners finished the learning activities, teachers can use LPA to analyze the learning portfolios of learners for refining their teaching strategies and contents.

8.1 Learning Portfolio Analysis Using Data Mining Approach

Several articles [9] [36] [65] [112] [140] have proposed that a new learner will get the similar learning performance if providing the learning guidance extracted from previous similar learners. The concept is the same as the adage of Chinese, "Good companions have good influence while bad ones have bad influence." Therefore, we conclude that a new learner could get the high learning performance if s/he follows the effective learning experience of similar learners. However, this conclusion results in the following three issues should be solved: (1) how to acquire the learning characteristics of learners, (2) how to group learners into several groups according to her/his individual learning characteristics, and (3) how to assign a new learner to a suitable group for offering her/him personalized learning materials.

8.1.1 The Process of Learning Portfolio

During learning activity, learning behaviors of learners can be recorded in the database, called *learning portfolio*, including the learning path, preferred learning course, grade of course, and learning time, etc., in the e-learning environment. Articles [4] [15] [29] [95] [104] [108] have proved that the information of learning portfolio can help teacher analyze the learning behaviors of learners and discover the learning rules for understanding the reason why a learner got high or low grade.

Therefore, based upon the learning portfolio with the predefined data format, we can apply sequential pattern mining approach to extract frequent learning patterns of learners. Then, according to these mined learning patterns, these learners can be grouped into several groups with the similar learning behaviors using clustering approach. By using the questionnaires including the Learning Style Indicator [77], Group Embedded Figures Test (GEFT) [137], etc. to acquire the learning characteristics of learners, we can acquire the learning characteristics of learners as learner profile which can be used to create a decision tree to predict which group a new learner belongs to.

Thus, in this dissertation, we propose a four phase Learning Portfolio Mining (LPM) Approach using sequential pattern mining, clustering approach, and decision tree creation sequentially. Then, in the last Phase, we also propose an algorithm to create personalized activity tree which can be used in SCORM compliant learning environment.

The Framework of Learning Portfolio Mining (LPM):

As mentioned above, we propose a Learning Portfolio Mining (LPM) approach to extract learning features from learning portfolio and then adaptively construct personalized activity tree with associated sequencing rules for learners.

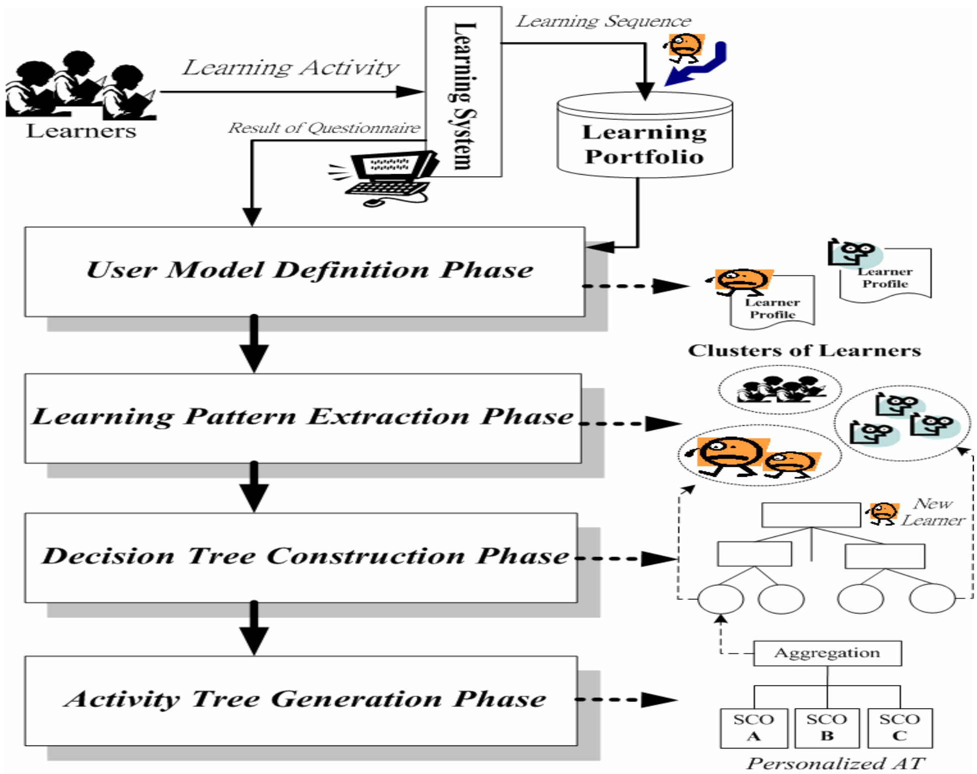


Figure 8.1: The Flowchart of LPM

As shown in Figure 8.1, LPM includes four phases described as follows:

1. **User Model Definition Phase:** we define firstly the learner profile including gender, learning style, and learning experience, etc. based upon existing articles and pedagogical theory, and the definitions of what we are going to discover in database.
2. **Learning Pattern Extraction Phase:** we apply sequential pattern mining technique to extract the maximal frequent learning patterns from the learning sequence within learning portfolio. Thus, original learning sequence of a learner

can be mapped into a bit vector where the value of each bit is set as 1 if the corresponding learning pattern is contained, and distance based clustering approach can be used to group learners with good learning performance into several clusters.

3. **Decision Tree Construction Phase:** after extraction phase, every created cluster will be tagged with a cluster labels. Thus, two third of the learner profiles with corresponding cluster label are used as training data to create a decision tree, and the remainings are the testing data which can be used to evaluate the created decision tree.
4. **Activity Tree Generation Phase:** finally, each created cluster including several learning patterns as sequencing rules can be used to generate personalized activity tree with associated sequencing rules of Sequencing and Navigation (SN).

8.1.2 The Clustering Process of Learner

In this section, we will describe the User Model Definition Phase and Learning Pattern Extraction Phase in LPM.

User Model Definition Phase:

Before extracting the learning features, we have to define a user model as learner profile, which will be recorded in database, to represent every learner. The definition is described as follows:

Learner L= (*ID*, *LC*, *LS*), where

- ***ID*:** denotes the unique identification of a learner.
- ***LC*** **=** $<c_1c_2\ldots c_m>$**:** denotes the sequence of learning characteristics of a learner.
- ***LS*** **=** $<s_1s_2\ldots s_n>$**:** denotes the learning sequence of a learner during learning activity, where s_i is an item of learning content.

In this dissertation, how to efficiently apply the existing pedagogical theories and how to further propose an efficient approach to solve personalized learning problem are our main concerns. Therefore, we only survey several related articles [9] [25] [29] [43] [66] [65] [83] [93] [104] [108] [135] [137], which investigated about 1) Learner Model, 2) Learning Style and Motivation, 3) course module category, 4) Learning Style, 5) Cognitive Styles, 6) Gender Difference, and 7) Student Characteristics, and then define the frequent learning characteristics for representing a learner by integrate their proposed leaning characteristics. The defined user model can also be extended if necessary. As shown in Table 8.1, the values of *Gender*, *Age*, *Education Status*, *Computer Experience*, and *Media Preference* can be inputted by learners directly and the values of *Learning Motivation*, *Cognitive Style*, *Learning Style*, and *Social Status* can be acquired by questionnaire, where we use the Learning Style Indicator [77] and Group Embedded Figures Test (GEFT) [137] to acquire the Kolb's Learning Style [66] and the information about field dependence/independence in Cognitive style, respectively. Here, the numeric value of *Age* can be transformed into symbolic with {L, M, H}. The transformation principle is described as follows:

*In all learners, ℓ and μ are the minimal and maximal values of age, respectively. Let $\Delta=(\ell-\mu)/3$, and then a numeric value of age can be mapped into symbolic value with **L** in $[\ell, \ell+\Delta)$, **M** in $[\ell+\Delta, \ell+2\Delta)$, and **H** in $[\ell+2\Delta, \ell+3\Delta]$.*

For example, **LC** = <F, M, S Y, H, FD, D, T, H> denotes that a learner is a *Female*, Age is *Medium* among all learners, Education Status is *Senior*, and etc. Nevertheless, the learning characteristics in user model can be modified for the real needs. In addition, **LS** denotes a learning sequence of a learner. For example, in Figure 2.2a, **LS** = <A, AA, AAA, AAB, AB> denotes that a learner studies the learning content A first and then studies the learning content AB. Therefore, based upon the user model, the learner can be represented as **L**=(35, <F, M, S Y, H, FD, D, T, H>, < A, AA, AAA, AAB, AB>).

Table 8.1: The Learning Characteristics of Learners

Attribute	Value
Gender	**F:** *Female*, **M:** *Men*
Age	**L:** $[\ell, \ell+\Delta)$, **M:** $[\ell+\Delta, \ell+2\Delta)$, **H:** $[\ell+2\Delta, \ell+3\Delta]$
Education Status	**E:** *Elementary*, **J:** *Junior,* **S:** *Senior*, **U:** *Undergraduate,* **G:** *Graduate*
Computer Experience	**Y:** *Yes*, **N:** *No*
Learning Motivation	**L:** *Low*, **M:** *Medium*, **H:** *High*
Cognitive Style	**FD:** *Field Dependence,* **FI:** *Field Independence*
Learning Style	**D: Doer (***Concrete Experience & Active Experimentation*) **W: Watcher** (*Reflective Observation & Concrete Experience*) **T: Thinker (***Abstract Conceptualization & Reflective Observation*) **F: Feeler** (*Active Experience & Abstract Conceptualization*)
Media Preference	**A:** *Audio*, **V:** *Video*, **T:** *Text*, **P:** *Picture*, **M:** *Picture & Text*
Social Status	**L:** *Low*, **M:** *Medium*, **H:** *High*

Learning Pattern Extraction Phase:

After defining the user model, we can apply sequential pattern mining technique to extract the maximal frequent learning patterns from the learning sequence within learning portfolio. Because we want to provide the new learner with effective learning guidance, we collect the learning sequences of learners with high learning performance, e.g., testing grade, from database, as shown in Table 8.2. For extracting the frequent learning pattern, the *Learning Pattern Extraction Phase* includes three processes shown in Figure 8.2: (1) Sequential Pattern Mining Process, (2) Feature Transforming Process, and (3) Learner Clustering Process.

Table 8.2: The Learning Sequences of 10 Learners

ID	Learning Sequence (LS)
1	<B, C, A, D, E, F, G, H, I, J>
2	<A, B, H, D, E, F, C, G, I, J>
3	<A, D, F, G, H, B, C, I, J>
4	<A, B, D, E, C, F, G, H>
5	<A, C, J, F, B, H, D, E, G>
6	<B, H, F, D, E, A, G, C, I>
7	<A, J, E, H, B, C, I, D, G>
8	<B, C, G, E, A, H, D, J, F>
9	<C, E, G, F, J, B, H, A, D>
10	<B, C, A, J, D, E, G, H, F>

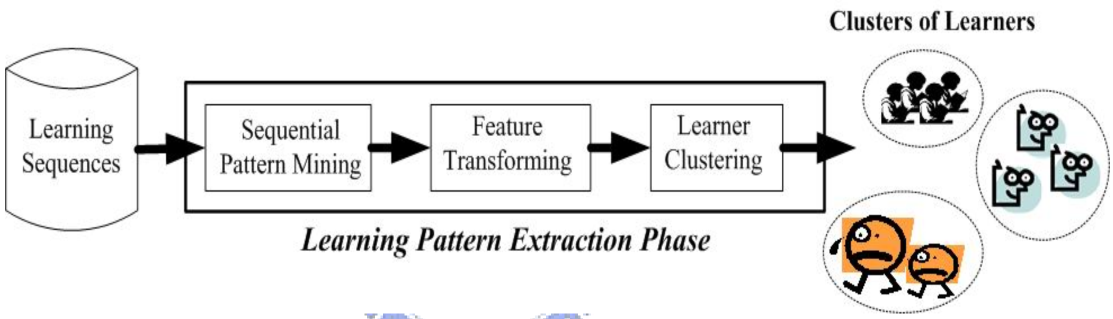


Figure 8.2: Learning Pattern Extraction Phase

Sequential Pattern Mining Process:

In this dissertation, we modify a sequential pattern mining approach, called GSP algorithm [4] [97], to extract the frequent learning patterns from learning portfolio because we use the maximal frequent learning pattern to represent the learning features of learners, shown in Figure 8.3.

Algorithm 8.1: Modified GSP Algorithm

Symbol Definition:

α: The minimum support threshold.

C_ℓ : The ℓ-Candidate itemset.

L_ℓ : The ℓ-large itemset

support(x) : it estimates the number of x in C_ℓ .

Input: Learning Sequence(*LS*) of learner, Minimal Support (α)

Output: The set of maximal frequent learning patterns (***MF***).

Step1: Generate and insert the 1-itemset into C_1

Step2: L_1={x |support(x)≧α, for x∈C_1}

Step3: Repeatedly execute this step until $C_\ell = NULL$.

3.1: $C_\ell = L_{\ell-1}$ JOIN $L_{\ell-1}$

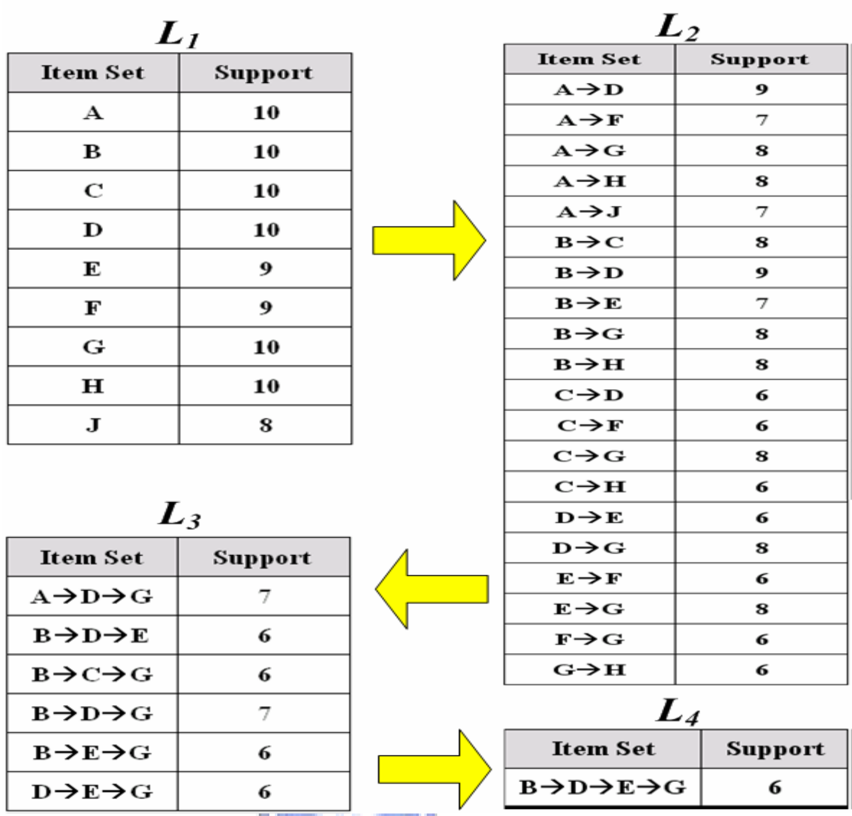
3.2: L_ℓ ={x |support(x)≧α, for $x \in C_\ell$ }

3.3: Insert x∈ L_ℓ into ***MF***, if ∃subsequence $y \subset x$ in ***MF*** then delete it.

Step5: **output the** *MF*

Figure 8.3: Maximal frequent sequential pattern mining algorithm

In Figure 8.3, the subsequence definition and JOIN process **(Step 3.1)** which are borrowed from GSP algorithm are described as follows. A sequence s_1 joins with s_2 if the subsequence obtained by dropping the first item of s_1 is the same as the subsequence obtained by dropping the last item of s_2. The candidate sequence generated by joining s_1 with s_2 is the sequence s_1 extended with the last item of s_2. For example, in L_3, sequence <A, B, C> joins with <B, C, D> to generate <A, B, C, D> for generating the C_4. In addition, in **MF**, a subsequence <A, B> and <B, C> will be deleted if a sequence <A, B, C> is generated in L_3 and <A, B, C> is the maximal frequent learning patterns (**Step 3.3**). Figure 8.4 shows the mining process of Modified GSP Algorithm with minimal support threshold α=6. Therefore, after applying the Modified GSP Algorithm for the learning sequences in Table 8.2, we can get the maximal frequent learning patterns as shown in Table 8.3.



L_1

Item Set	Support
A	10
B	10
C	10
D	10
E	9
F	9
G	10
H	10
J	8

L_2

Item Set	Support
A→D	9
A→F	7
A→G	8
A→H	8
A→J	7
B→C	8
B→D	9
B→E	7
B→G	8
B→H	8
C→D	6
C→F	6
C→G	8
C→H	6
D→E	6
D→G	8
E→F	6
E→G	8
F→G	6
G→H	6

L_3

Item Set	Support
A→D→G	7
B→D→E	6
B→C→G	6
B→D→G	7
B→E→G	6
D→E→G	6

L_4

Item Set	Support
B→D→E→G	6

Figure 8.4: Mining process of modified GSP algorithm withα= 6

Table 8.3: The set of maximal frequent learning patterns (MF)

Large Itemset	Maximal Frequent Learning Patterns									
***L*₂**	A→F	A→H	A→J	B→H	C→D	C→F	C→H	E→F	F→G	G→H
***L*₃**	A→D→G	B→C→G								
***L*₄**	B→D→E→G									

Feature Transforming Process:

The generated maximal frequent learning patterns can be used to represent learning features of learners, which denotes that a learner would get high learning performance if s/he follows these learning patterns. Thus, based upon maximal learning patterns in Table 8.3, the original learning sequences of every learner can be mapped into a bit vector where the value of each bit is set as 1 if the mined maximal learning pattern is a subsequence of original learning sequence. For example, in Table 8.3, the frequent learning pattern <B→D→E→G> is a subsequence of learning sequence <A, B, H, D, E,

F, C, G, I, J> of second learner and the <C→D> is not. Therefore, we can get the bit vector of every learner according to feature transforming process [44] as shown in Table 8.4.

Table 8.4: The result of feature transforming process

TID	B→D→E→G	A→D→G	B→C→G	A→F	A→H	A→J	B→H	C→D	C→F	C→H	E→F	F→G	G→H
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0	0	0	1	1	0
3	0	1	0	1	1	1	0	0	0	0	0	1	1
4	1	1	1	1	1	0	1	0	1	1	1	1	1
5	1	1	0	1	1	1	1	1	1	1	0	1	0
6	1	0	0	0	0	0	1	0	0	0	0	1	0
7	0	1	1	0	1	1	0	1	0	0	0	0	0
8	0	0	1	1	1	1	1	1	1	1	1	0	1
9	0	0	0	0	0	0	1	1	1	1	1	0	1
10	1	1	1	1	1	1	1	1	1	1	1	0	1

Learner Clustering Process:

As mentioned above, every learner can be represented by mined frequent patterns. Therefore, we can apply clustering algorithm to group learners into several clusters according to learning features of learners. In the same cluster, every learner with high learning performance has the similar learning behaviors. However, it is difficult to determine the number of clusters for applying clustering approach like K-means algorithm. A clustering algorithm, called ISODATA [48], can dynamically change the number of clusters by lumping and splitting procedures and iteratively change the number of clusters for better result. Therefore, in this paper, we apply the ISODATA clustering approach to group learners into different clusters. The Table 8.5 shows the result after applying ISODATA Clustering Algorithm for Table 8.4. The bit vector in Cluster Centroid Field denotes the representative learning patterns set in a cluster, which will be used to generate the sequencing rules of SCORM later.

Table 8.5: The result of applying ISODATA clustering algorithm

Cluster Label	ID of Learner	Cluster Centroids
1	{1, 4, 5, 8, 10}	<1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1>
2	{7}	<0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0>
3	{2, 3, 9}	<1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0>
4	{6}	<1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0>

8.1.3 The Prediction and Construction of Learning Guidance:

In this section, we will describe the Decision Tree Construction Phase and Activity Tree Generation Phase in LPM.

Decision Tree Construction Phase:

After learner pattern extraction phase, every created cluster will be tagged with a cluster label as shown in Table 8.5. However, how to assign a new learner to a suitable cluster according to her/his learning characteristics and capabilities is an issue to be solved. Fortunately, the decision tree approach can solve this issue. Thus, based upon the Learner Profiles with cluster labels in Table 8.6, we can apply decision tree induction algorithm, ID3 [95], to create a decision tree. In this paper, two third of the learner profiles with associated cluster label are used as training data to create a decision tree, and the remainings are the testing data. The result of applying ID3 algorithm is shown in Figure 8.5.

Table 8.6: The learner profiles with cluster labels

ID	Gender	Age	Education Status	Computer Experience	Learning Motivation	Cognitive Style	Learning Style	Preferred Media	Social Status	Cluster Label
1	F	M	U	Y	M	FI	D	A	H	**1**
2	F	L	S	N	H	FI	W	A	M	**3**
3	M	L	U	N	L	FI	D	T	M	**3**
4	M	M	S	Y	H	FI	W	G	L	**1**
5	F	M	U	Y	H	FI	T	A	M	**1**
6	M	M	U	N	L	FD	W	G	L	**4**
7	F	H	S	Y	H	FI	W	T	M	**2**
8	M	L	S	N	M	FD	T	T	H	**1**
9	F	M	H	Y	H	FI	F	G	M	**3**
10	M	H	H	Y	L	FD	D	G	M	**1**

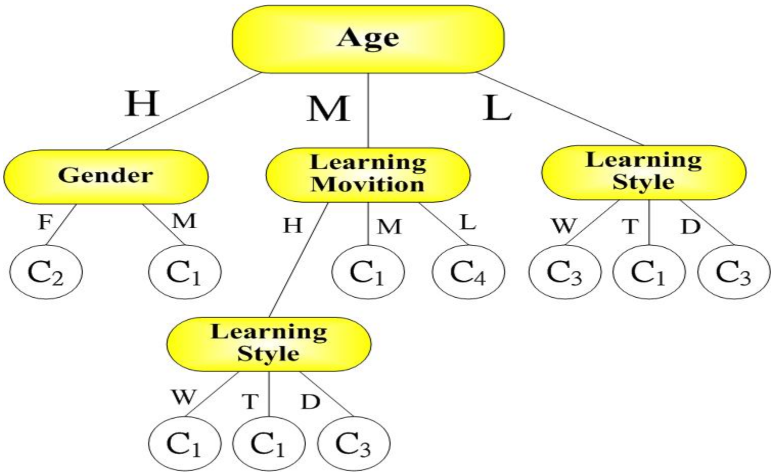


Figure 8.5: The decision tree based upon the learner profiles in Table 6

Activity Tree Generation Phase:

Finally, based upon the created decision tree, we can assign a new learner to a suitable cluster which contains several learning guidance. Each cluster contains a cluster centroid which corresponds to several learning patterns as sequencing rules in sequencing and navigation (SN) of SCORM 2004. Therefore, in this dissertation, we propose an algorithm to transform learning patterns of cluster into sequencing rules and then create the personalized activity tree, as shown in Figure 8.6.

Algorithm 8.2: Personalized Activity Tree Creation (PATC) Algorithm

Symbol Definition:

***LI*:** The set of learning items in a learning activity.

***CC*:** The corresponding learning patterns in Cluster Centroid

***LP*:** The set of Learning Patterns

***VA*:** Virtual Aggregation Node

SCO: The Sharable Content Object (SCO) of SCORM standard

Input: Learning Items (***LI***) and corresponding learning patterns (***CC***)

Output: Personalized Activity Tree (***PAT***)

Step 1: ***LP*** = { *lp* | for $lp \in$ CC}

Step 2: For each lp_i

1. Create a ***VA*** with sequencing rules: "*Flow=true*", "*Forward Only=true*", and "*Rollup Rule=All*".
2. The ***VA*** links every item as SCO in lp_i in order.
3. Set All SCOs with Rule,"if NOT complete, Deny Forward Process".

Step 3: If $\exists$ the same SCO in different VA,

Then create a ***Learning Objective*** to link these SCOs.

Step 4: ***ItemSet*** = { *x* | for $(x \in LI) \cap (x \notin CC)$}

Step 5: Create a ***VA*** with sequencing rules, "*choice=true*" and "*choice exit=True*", to link all items as SCO in ***ItemSet***.

Step 6: Create a *Root Aggregation* **node with sequencing rule, "*Flow=true*" and "*Choice=true*", to link all** *VAs*.

Figure 8.6: The algorithm of personalized activity tree creation (PATC)

For the data of Cluster 2 in Table 8.5, the results of PATC algorithm are shown in Figure 8.7. Firstly, in Step 1, the ***LP*** will be inserted five learning patterns according to the centroid of cluster 2, i.e., ***LP***={A→D→G, B→C→G,C→D, A→H, A→J}. In Step 2, because a learning pattern, which contains several items as SCO in SCORM, e.g., the item A in pattern A→H, represents an effective learning sequence, we can create a virtual aggregation node as a sub-activity to aggregate all items in learning pattern in order. Here, A Sharable Content Object (SCO) denotes "*a set of related resources that comprise a complete unit of learning content compatible with SCORM run-time* *requirements*" (SCORM, 2004). Moreover, in each SCO, we set its sequencing rule with "*if NOT complete, Deny Forward Process*" for controlling the navigation order. In order to make learners complete all learning objects (SCO) and satisfy the pass condition, we set the Rollup rule as "*All*". The rules, *Flow=true*" and "*Forward Only=true"*, can forbid learners to learn backward. In addition, a *learning objective* is created to link the same items appeared in different learning patterns. By setting the value of *learning objective*, we can forbid to learn an item repeatedly. For example, in Figure 8.7, the *learning objective*, called OBJ-A, links the SCO A in Aggregations 1, 3, and 4. After learner satisfied the SCO A, the OBJ-A is set and then the SCO A in Aggregations 3 and 4 will be skipped. In addition, the frequent learning patterns may not contain all learning items in the learning activity. Thus, we also create an aggregation node as referable learning activity to link these items which are not contained in learning patterns, e.g., in Figure 8.7, the Aggregation 6 contains {E, F, I} and rules, "*choice=true*" and "*choice exit=True*", for free navigation. Finally, the root aggregation node is used to link all aggregation nodes.

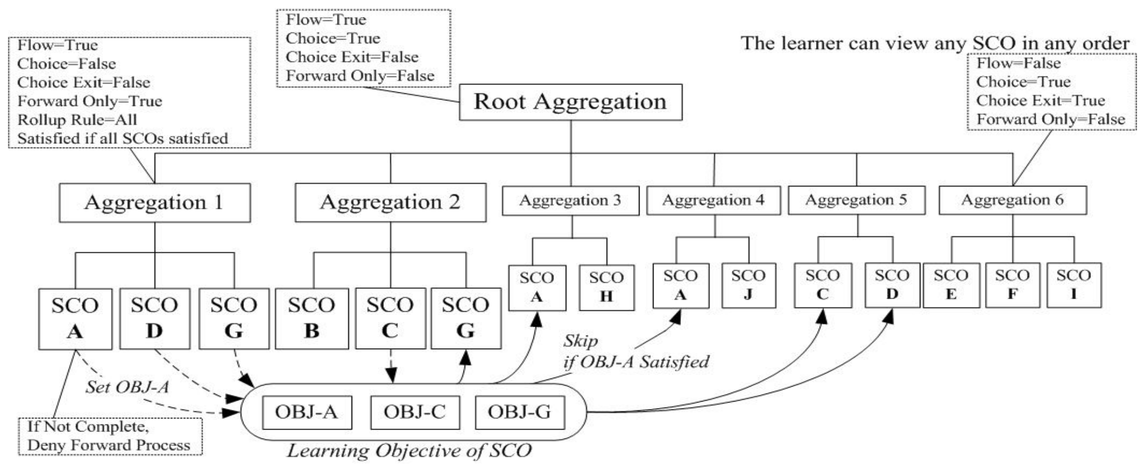


Figure 8.7: The result of PATC algorithm based upon cluster 2

8.2 Two-Phase Concept Map Construction (TP-CMC)

In the last five years, many adaptive learning and testing systems have been proposed to offer learners customized courses in accordance with their aptitudes and learning results [5] [21] [22] [38] [41] [49] [51] [54] [122] [126]. For achieving the adaptive learning, a predefined concept map of a course, which provides teachers for further analyzing and refining the teaching strategies, is often used to generate adaptive learning guidance. However, it is difficult and time consuming to create the concept map of a course. Thus, how to automatically create a correct concept map of a course becomes an interesting issue.

Therefore, in this dissertation, we propose a Two-Phase Concept Map Construction (TP-CMC) algorithm to automatically construct a concept map of a course by historical testing records. In TP-CMC, the Test item-Concept Mapping Table records the related learning concepts of each test item. As shown in Table 8.7, five quizzes contain these related learning concepts A, B, C, D and E, where "1" indicates the quiz contains this concept, and "0" indicates not. Moreover, a concept set of quiz *i* is denoted as CS_{Qi}, e.g., CS_{Q5}={B, D, E}. The main idea of our approach is to extract the prerequisite relationships among concepts of test items and construct the concept map. Based upon assumptions, for each record of learners, each test item has a grade.

Table 8.7: Test Item–Concept Mapping Table

	A	B	C	D	E
Q_1	0	0	0	1	0
Q_2	1	0	1	0	0
Q_3	1	0	0	0	0
Q_4	0	1	1	0	0
Q_5	0	1	0	1	1

As shown in Figure 8.8, our Concept Map Construction includes two phases: ***Grade Fuzzy Association Rule Mining Process Phase*** and ***Concept Map Constructing Process Phase***. The first phase applies fuzzy theory, education theory, and data mining approach to find four fuzzy grade association rule types, L-L, L-H, H-H, H-L, among test items. The second phase further analyzes the mined rules based upon our observation in real learning situation. Even based upon our assumptions, constructing a correct concept map is still a hard issue. Accordingly, we propose a heuristic algorithm which can help construct the concept map.

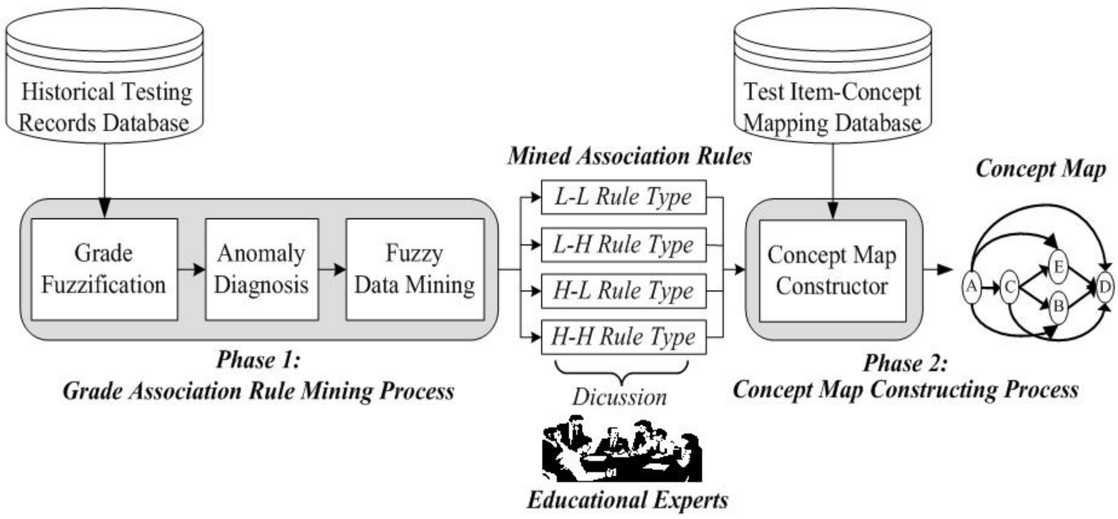


Figure 8.8: The Flowchart of Two-Phase Concept Map Construction (TP-CMC)

8.2.1 Grade fuzzy association rule mining process

In [126], the Look Ahead Fuzzy Association Rule Miming Algorithm (LFMAlg) has been used to find the associated relationship information embedded in the testing records of learners. In this phase, we propose an anomaly diagnosis process to improve LFMAlg and reduce the input data before the mining process.

1. Grade Fuzzification:

Firstly, because the numeric testing data are hard to analyze by association rule mining approach, we apply Fuzzy Set Theory to transform these into symbolic. Thus, after the fuzzification, the grade on each test item will be labeled as high (H), middle (M), and low (L) degree, which can be used as an objective judgment of learner's performance.

2. Anomaly diagnosis:

Based upon Item Analysis for Norm-Referencing of Educational Theory [89], the discrimination of item can tell us how good a test item is, i.e., item with high degree of discrimination denotes that the item is well designed. If the discrimination of the test item is too low (most students get high score or low score), this item as redundant data will have no contribution to construct the concept map. For decreasing the redundancy of test data, we propose a fuzzy item analysis, called Anomaly Diagnosis, to refine the test data.

3. Fuzzy Data Mining:

Then, we can apply LFMAlg [126] to find the grade fuzzy association rules of test items from the historical testing data. In this dissertation, we analyze the prerequisite relationships among learning concepts of quizzes according to 4 association rule types, ***L-L, L-H, H-L, H-H,*** generated from Large 2 Itemset. Q_i.L notation denotes that the *i*th question (Q) was tagged with low (L) degree, e.g., $Q_2.L \rightarrow Q_3.L$ means that learners get low grade on Q_2 implies that they may also get low grade on Q_3.

8.2.2 Concept map constructing process

1. Concept map constructor:

Firstly, the result of analyzing four association rule types, L–L, L–H, H–H, and H–L, are used to construct the prerequisite relationships between concept sets, which are used to define the edge between nodes of concept set and provide teachers with information for further refining the test sheet, of learning concepts of test items. Then, based on the prerequisite relationships of concept set and the Test item-Concept Mapping Table, we propose a Concept Map Constructing (CMC) Algorithm to find the corresponding learning concepts of concept set to construct the concept map according to the join principles of concept-pair.

8.2.3 Grade fuzzy association rule mining process

1. Grade fuzzification:

As described in Section 8.2.1, we apply fuzzy concept to transform numeric grade data into symbolic, called Grade Fuzzification. Three membership functions of each quiz's grade are shown in Figure 8.9. In the fuzzification result, "Low", "Mid", and "High" denote "Low Grade", "Middle Grade", and "High Grade" respectively. $Q_i.L$, $Q_i.M$, and $Q_i.H$ denote the value of LOW fuzzy function, MIDDLE fuzzy function, and HIGH fuzzy function for the quiz i, respectively. By given membership functions, the fuzzification of testing records is described in Example 8.1.

Example 8.1:

In Figure 8.10, assume there are 10 testing records with 5 quizzes of learners and the highest grade on each quiz is 20.

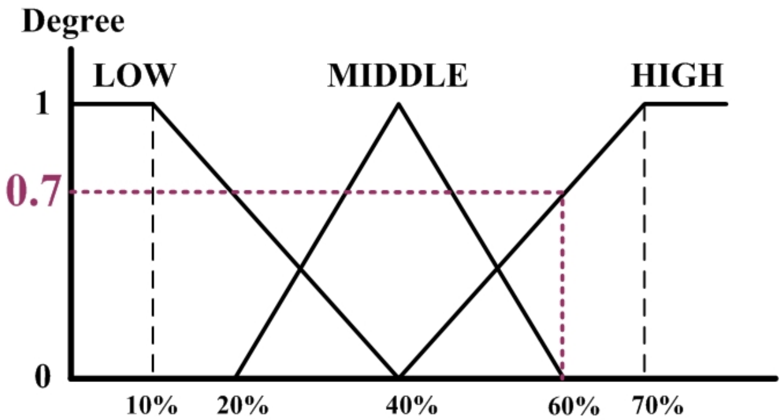


Figure 8.9: The given membership functions of each quiz's grade.

Student ID	Q_1	Q_2	Q_3	Q_4	Q_5	Total
1	12	18	20	20	7	77/100
2	12	14	18	3	7	54/100
3	12	16	14	4	7	53/100
4	2	8	12	6	20	48/100
5	2	8	12	2	12	36/100
6	2	10	8	2	20	44/100
7	20	5	5	4	1	35/100
8	10	6	6	1	5	28/100
9	10	5	5	1	5	26/100
10	10	3	4	0	5	21/100

Student ID	Q_1			Q_2			Q_3			Q_4			Q_5		
	Low	Mid	High	Low	Mid	High	Low	Mid	High	Low	Mid	High	Low	Mid	High
1	0.0	0.0	0.7	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.2	0.8	0.0
2	0.0	0.0	0.7	0.0	0.0	1.0	0.0	0.0	1.0	0.8	0.0	0.0	0.2	0.8	0.0
3	0.0	0.0	0.7	0.0	0.0	1.0	0.0	0.0	1.0	0.7	0.0	0.0	0.2	0.8	0.0
4	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.7	0.3	0.5	0.0	0.0	0.0	1.0
5	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.7	1.0	0.0	0.0	0.0	0.0	0.7
6	1.0	0.0	0.0	0.0	0.5	0.3	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
7	0.0	0.0	1.0	0.5	0.3	0.0	0.5	0.3	0.0	0.7	0.0	0.0	1.0	0.0	0.0
8	0.0	0.5	0.3	0.3	0.5	0.0	0.3	0.5	0.0	1.0	0.0	0.0	0.5	0.3	0.0
9	0.0	0.5	0.3	0.5	0.3	0.0	0.5	0.3	0.0	1.0	0.0	0.0	0.5	0.3	0.0
10	0.0	0.5	0.3	0.8	0.0	0.0	0.7	0.0	0.0	1.0	0.0	0.0	0.5	0.3	0.0
Sum	3.0	1.5	4.0	2.1	3.6	3.3	2.0	2.1	4.4	7.5	.5	1.0	3.4	3.3	2.7

Figure 8.10: The Fuzzification of Learners' Testing Records

2. Anomaly diagnosis:

For refining the input testing data, we propose the anomaly diagnosis, called Fuzzy Item Analysis for Norm-Referencing (FIA-NR) by applying Item Analysis for Norm-Referencing of Educational Theory, shown in Figure 8.11. A test item will be deleted if it has low discrimination.

Algorithm: Fuzzy Item Analysis for Norm-Referencing (FIA-NR)

Symbol Definition:

R_{iH} /R_{iL}: The sum of the fuzzy grades (H=1, M=0.5, L=0) on test item i for each learner in the high(H)/Low(L) group.

N_{iH} /N_{iL}: The number of learners in high/low group.

P_{iH} & P_{iL}: the ratios of R_{iH} to N_{iH} and of R_{iL} to N_{iL}, respectively.

Input : Fuzzified testing records of learners

Output : the ***Difficulty index*** (***P_i***) and the ***Discrimination index*** (D_i) of each test item

Step1: Sort Scores in descending order and divide it into **High**, **Middle**, and **Low** groups, each has 1/3 learners.

Step2: Let $P_{iH} = \frac{R_{iH}}{N_{iH}}$ and $P_{iL} = \frac{R_{iL}}{N_{iL}}$.

Step 3: Compute $P_i = 1 - \frac{P_{iH} + P_{iL}}{2}$ and $D_i = P_{iH} - P_{iL}$, for $i=1, \ldots, k$.

Step 4: Delete the test items with low Discrimination (<0.5).

Figure 8.11: Fuzzy Item Analysis for Norm-Referencing (FIA-NR)

Example 8.2:

Table 8.8 shows the fuzzified testing grades of learners on Q_4 sorted in the descending order of each learner's total score in the test sheet. For example, in Figure 8.10, because the result of fuzzification of learner ID 4 is (0.3, 0.5, 0.0), her/his Grade Level can be tagged with *M* by the Max(L, M, H) function.

Table 8.8: Sorted Fuzzified Testing Grade on Q_4

Group	**High**			**Middle**				**Low**		
Learner ID	1	2	3	4	6	5	7	8	9	10
Total (100)	77	54	53	48	44	36	35	28	26	21
Grade Level =Max(L,M,H)	H	L	L	M	L	L	L	L	L	L

Then, by applying FIA-NR algorithm, we can get the *Difficulty* and *Discrimination* of every quiz. For example, the P_{4H} and P_{4L} of Q_4 are

$P_{4H} = \frac{R_{4H}}{N_{4H}} = \frac{H + L + L}{3} = \frac{1 + 0 + 0}{3} = \frac{1}{3}$ and $P_{4L} = \frac{0}{3} = 0$, respectively. Therefore, its *Difficulty* P_4 and *Discrimination* D_4 are $P_4 = 1 - \frac{P_{4H} + P_{4L}}{2} = 1 - \frac{1/3+0}{2} = \frac{5}{6} = 0.83$ and 0.33 respectively. Thus, learners' grade on Q_4 will be deleted because its ***Discrimination*** is too low to use during the mining process and the construction of the concept map. Accordingly, the test sheet can be redesigned. All evaluated results are shown in Table 8.9.

Table 8.9: *Difficulty* and *Discrimination* Degree of Each Quiz

	Q_1	Q_2	Q_3	Q_4	Q_5
Difficulty (0 to 1)	0.25	0.42	0.42	0.83	0.75
Discrimination (-1 to 1)	0.5	0.83	0.83	0.33	0.5

3. Fuzzy Data Mining:

After filtering out these useless quizzes, we can apply Look Ahead Fuzzy Association Rule Mining Algorithm [126] as shown in Figure 8.12 to find the fuzzy association rules of test items. In LFMAlg Algorithm, the support value of every itemset x in candidate C_ℓ can be evaluated by the *support*(x) function, where x={A, B}$\subseteq C_{\ell-1}$, $A \cap B = \phi$. Then, the *support*(x) = *support*($A \cup B$) = $\sum_1^n Min(A, \mathrm{B})$, where n is the number of learners. For example, in Figure 8.10, *support*(Q_1.L, Q_3.H)=*Min*(1.0, 0.7)+*Min*(1.0, 0.7) =1.4.

Algorithm: LFMAlg Algorithm
Symbol Definition:
α_ℓ : The minimum support threshold in the ℓ -large itemset.
C_ℓ : The ℓ -Candidate itemset.
L_ℓ : The ℓ -large itemset
λ : The minimum confidence threshold.
Input: The test records of learners after Fuzzification and Anomaly Diagnosis. The minimum support threshold α_1 and λ.
Output : The fuzzy association rules of test records of learners.

Step1: Repeatedly execute this step until C_ℓ = NULL.
  1.1: Generate and insert the ℓ -itemset into C_ℓ
  1.2: $\alpha_\ell = \max(\frac{\alpha_1}{2} \quad , \quad \alpha_{\ell-1} - \frac{\alpha_1}{(\ell-1)\times c})$, where ℓ >1 and c is constant.
  1.3: $L_\ell = \{\ x \mid \text{support(x)} \geq \alpha_\ell, \quad for \quad x \in C_\ell \ \}$
  1.4: $\ell = \ell + 1$
Step2: Generate the association rules according to the given λ in L_ℓ .

Figure 8.12: Look ahead Fuzzy Association Rule Mining Algorithm (LFMAlg)

Example 8.3:

For the data shown in Examples 8.1 and 8.2, Figure 8.13 shows the process of finding the association rules with large 2 itemset by LFMAlg algorithm.

$\alpha_1 = 2.0$

L$_1$

Itemset	Support
Q1.L	3.0
Q1.H	4.0
Q2.L	2.1
Q2.H	3.3
Q3.L	2.0
Q3.H	4.4
Q5.L	3.4
Q5.H	2.7

$\alpha_2 = 1.6$

C$_2$

Itemset	Support
Q1.L ∩ Q2.L	0.0
Q1.L ∩ Q2.H	0.3
Q1.L ∩ Q3.L	0.0
Q1.L ∩ Q3.H	1.4
……	
Q3.L ∩ Q5.L	1.8
Q3.L ∩ Q5.H	0.0
Q3.H ∩ Q5.L	0.9
Q3.H ∩ Q5.H	1.4

Support >= 1.6

L$_2$

Itemset	Support
Q1.L ∩ Q5.H	2.7
Q1.H ∩ Q2.H	2.1
Q1.H ∩ Q3.H	2.1
Q1.H ∩ Q5.L	2.8
Q2.L ∩ Q3.L	2.0
Q2.L ∩ Q5.L	1.8
Q2.H ∩ Q3.H	3.0
Q3.L ∩ Q5.L	1.8

Figure 8.13: The Mining Process of Large 2 Itemset

Thus, Table 8.10 shows the grade fuzzy association rules with minimum confidence 0.8 generated from large 2 itemset into L-L, L-H, H-H, and H-L types. The $Conf_i$ (Confidence) is used to indicate the important degree of *i*th mined association rule. For example, the Confidence ($Conf_1$) of rule $Q_2.L \rightarrow Q_3.L$ can be obtained as follows.

$$Q_2.L \rightarrow Q_3.L : Confidence = \frac{support(Q_2.L \cup Q_3.L)}{support(Q_2.L)} = 0.95$$

Table 8.10: The Mining Results ($Conf_i > 0.8$)

Rule Types	***Mined Rules***	***Confi***
L-L	Q2.L→Q3.L	0.95
	Q3.L→Q2.L	1.00
	Q2.L→Q5.L	0.86
	Q3.L→Q5.L	0.90
L-H	Q1.L→Q5.H	0.90
	Q5.L→Q1.H	0.82
H-H	Q2.H→Q3.H	0.91
H-L	Q5.H→Q1.L	1.00

8.2.4 Concept Map Constructing Process

1. Concept Map Constructor:

Before constructing the concept map, we can get the prerequisite relationship among concepts of quiz from analyzing four association rule types, L-L, L-H, H-L, and H-H, based upon our observation obtained by interviewing the educational experts, in real learning situation. Therefore, we can conclude the *Heuristic 1*: given two quizzes Q_1 and Q_2, if concepts of Q_1 are the prerequisite of concepts of Q_2, Learner gets low grade on Q_1 implies that s/he may also get low grade on Q_2 or Learner gets high grade on Q_2 implies that her/his grade on Q_1 is high. As shown in Table 8.11, for each rule type, we use *Heuristic 1* to get its prerequisite relationships among concept sets of quizzes with parameterized possibility weight, which are used to construct the concept map. The definition of the symbols used in Table 8.11 is described as follows.

Symbol Definition:

CS_{Qi} **:** indicate concept set of quiz *i*

W_i **:** indicate the possibility of the possible scenario of the rule

Table 8.11: Prerequisite Relationship of Association Rule

Rule	W*i*	Prerequisite Relationship
$Q_i.L \rightarrow Q_j.L$	1.0	$CS_{Qi} \xrightarrow{pre.} CS_{Qj}$
$Q_i.L \rightarrow Q_j.H$	0.8	$CS_{Qj} \xrightarrow{pre.} CS_{Qi}$
$Q_i.H \rightarrow Q_j.H$	1.0	$CS_{Qi} \xrightarrow{pre.} CS_{Qj}$
$Q_i.H \rightarrow Q_j.L$	0.8	$CS_{Qi} \xrightarrow{pre.} CS_{Qj}$

In this dissertation, association rules generated from Large 2 Itemset are firstly used to analyze the prerequisite relationships between learning concepts of quizzes.

Therefore, by looking up Table 8.11, we can obtain the prerequisite relationships of concept set of quizzes with the possibility weight (W_i) for each mined rule in Table 8.10. The possibility W_i is a heuristic parameter of CMC algorithm because it can be modified according to different domains and learners' background. Moreover, the related explanations of the analysis in Table 8.11 are shown in Table 8.12. Table 8.13 shows the result of transforming association rules in Table 8.10 by analyzing the prerequisite relationships in Table 8.11.

Table 8.12: The Explanations of Rule Types

Rule	Description of Learning Scenario
L→L	If the association rule Q_i.L→Q_j.L is mined, it means that the CS_{Qi} is the prerequisite of CS_{Qj}, represented as $CS_{Qi} \xrightarrow{pre.} CS_{Qj}$. That is why getting low grade on Q_i might imply getting low grade on Q_j.
H→H	If the association rule Q_i.H→Q_j.H is mined, it means that the CS_{Qi} is the prerequisite of CS_{Qj}.
L→H	If the association rule Q_i.L→Q_j.H is mined, it means that the CS_{Qj} is the prerequisite of CS_{Qi} because CS_{Qi} may be not learned well resulting from CS_{Qj}.
H→L	If the association rule Q_i.H→Q_j.L is mined, it means that the CS_{Qi} is the prerequisite of CS_{Qj}.

Table 8.13: Result by Analyzing the Prerequisite Relationships in Table 8.11

Rule Type	***Association rules of quiz***	***Prerequisite relationship of Concept Set***	***Conf*** $_i$	W_i
L-L	Q2.L→Q3.L	$CS_{Q2} \xrightarrow{pre.} CS_{Q3}$	0.95	1.0
	Q3.L→Q2.L	$CS_{Q3} \xrightarrow{pre.} CS_{Q2}$	1.00	1.0
	Q2.L→Q5.L	$CS_{Q2} \xrightarrow{pre.} CS_{Q5}$	0.86	1.0
	Q3.L→Q5.L	$CS_{Q3} \xrightarrow{pre.} CS_{Q5}$	0.90	1.0
L-H	Q1.L→Q5.H	$CS_{Q5} \xrightarrow{pre.} CS_{Q1}$	0.90	0.8
	Q5.L→Q1.H	$CS_{Q1} \xrightarrow{pre.} CS_{Q5}$	0.82	0.8
H-H	Q2.H→Q3.H	$CS_{Q2} \xrightarrow{pre.} CS_{Q3}$	0.91	1.0
H-L	Q5.H→Q1.L	$CS_{Q5} \xrightarrow{pre.} CS_{Q1}$	1.00	0.8

For example, in Figure 8.14, the mined rules, $Q_1.L \rightarrow Q_2.H$ and $Q_1.H \rightarrow Q_2.L$, can be transformed into corresponding prerequisite relationship of concept set, resulting in a confused relation as a cycle between concept sets, called circularity. That is to say, concepts of Q_1 and concepts of Q_2 are prerequisite of each other, which is a conflict in our analysis. Therefore, during creating the concept map, we have to detect whether a cycle exists or not, e.g., $CS_{Q1} \rightarrow CS_{Q2} \rightarrow CS_{Q1}$.

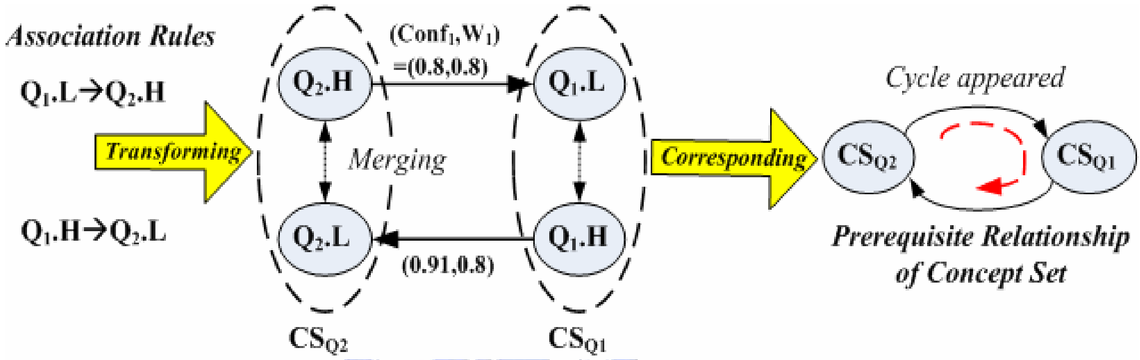


Figure 8.14: The Transforming of Association Rules.

Because each concept set may contain one or more learning concepts, we further define a principle of joining two concept sets and then generate corresponding concept-pair, (C_i, C_j), that is, *if* $CS_{Q1}=\{\bigcup_1^n a_i\}$ *and* $CS_{Q2}=\{\bigcup_1^m b_j\}$, *the set of concept-pair is* CS_{Q1} *JOIN* CS_{Q2} $=\{\bigcup_1^k (a_i,b_j)\}$, *where* $a_i \neq b_j$ and $k \leqq n \times m$. For example, if $CS_{Q1}=\{a_1,a_2\}$ and $CS_{Q2}=\{b_1,b_2\}$, CS_{Q1} JOIN CS_{Q2} $=\{(a_1,b_1),(a_1,b_2),(a_2,b_1)\}$, where $a_2=b_2$ is deleted. The related definition used in creating the concept map is given as follows:

Concept Map CM = (V, E), where

- **V** = $\{C_i \mid$ the node is unique for each $i\}$
- **E** = $\{ \overrightarrow{C_iC_j} \mid i \neq j \}$

The node, C_i, denotes the learning concept and the edge, $\overrightarrow{C_iC_j}$, which connects C_i and C_j, denotes that C_i is the prerequisite of C_j. The $\overrightarrow{C_iC_j}$ has an *Influence Weight*, IW_k, denotes the degree of relationship between learning concepts. The formulation of IW_k is $((k-1)\times IW_{k-1}+W_k\times Conf_k)/k$, $1\leqq k\leqq n$, where n is the amount of $\overrightarrow{C_iC_j}$.

The proposed Concept Map Constructing (CMC) algorithm is shown in Figure 8.15.

Algorithm : Concept Map Constructing (CMC) Algorithm
Input : Association rules of quiz with its grade level
Output : Concept Map

Step 1 : ***CM*** = {*Null*}.
Step 2 : For each association rule
  2.1: Insert CS_{Qi} for Q_i.L or Q_i.H and CS_{Qj} for Q_j.L or Q_j.H.
  2.2: Create a new *prerequisite edge* with confidence ($Conf_i$) and weight (W_i) of rule to connect the CS_{Qi} and CS_{Qj} according to Table 6.
Step 3 : In *CM*, repeatedly execute this step until no cycle is found.
  3.1: Do *Cycle Detection Process*.
  3.2: If a cycle is found
    Then Remove the edge with the lowest confidence.
Step 4 : Delete all independent nodes without connected edges.
Step 5 : For each node CS_{Qi} in ***CM***
  { Insert node's corresponding learning concepts according to ***Test item–Concept Table*** }
Step 6 : For each edge $\overline{CS_{Qi}CS_{Qj}}$ in ***CM***, perform the following substeps.
  Step 6.1: Use **JOIN** principle to join two connected concept sets for generating the concept-pair, (C_i, C_j) which replaces the original concept sets.
  Step 6.2: Re-compute the Influence Weight IW_k of the *kth* concept-pair (C_i,C_j) according to weight (W_k) and confidence ($Conf_k$) of $\overline{CS_{Qi}CS_{Qj}}$.
Step 4: Output the ***CM***.

Figure 8.15: Concept Map Constructing (CMC) Algorithm

For the CMC algorithm shown in Figure 8.15, the main purpose of *Cycle Detection Process* is to detect the unreasonable prerequisite relationship as a cycle among concept sets. It should be noted that the prerequisite relationship in the concept set map also fulfills the indicator $f_{1\bar{2}} > f_{\bar{1}2}$ in Table 8.14, which is an extension of [5] after cycle detection. The indicator denotes that if concepts of Q_1 are prerequisite of concepts of Q_2, it is reasonable that $f_{1\bar{2}} > f_{\bar{1}2}$, where $f_{1\bar{2}} = Count(Q_1.H \cap Q_2.L)$ and $f_{\bar{1}2} = Count(Q_1.L \cap Q_2.H)$. In addition, the Influence Weight, IW_k, denotes the degree how the learning status of concept C_i influences C_j. Therefore, the number of $\overrightarrow{C_iC_j}$ will enhance the value of Influence Weight. In the formulation of influence weight, the W_i denotes the possibility of the learning scenario of the association rule in our analysis. Thus, the educational experts can assign different value of W_i to the algorithm according to different domains and learner's backgrounds.

Table 8.14: Relative Quizzes Frequency

	(Q_1) Higher	**(Q_1) Lower**
(Q_2) Higher	f_{12}	$f_{\bar{1}2}$
(Q_2) Lower	$f_{1\bar{2}}$	$f_{\bar{1}\bar{2}}$

For the association rules given in Table 8.13, the process of CMC algorithm is shown in Figure 8.16. In Figure 8.16b, the edges drawn as dash line have the lowest confidences in cycles will be deleted in Cycle Detection Process. Moreover, Table 8.15 shows the example of computing the Influence Weight of Concept-Pair (B, E) in Figure 8.16f. Because the Concept-Pair (B, E) has two edges between CS_{Q5} and CS_{Q1}, we have to compute the Influence Weight twice.

Table 8.15: The Result of Computing the Influence Weight of Concept-Pair (B, D) in Figure 8.16.f

Rule	***Prerequisite Relationship***	***$Conf_i$***	***W_i***	IW_i
$Q_1.L \rightarrow Q_5.H$	$CS_{Q5} \rightarrow CS_{Q1}$	0.90	0.8	$W_1 \times Conf_1 = 0.9 * 0.80 \cong 0.72$
$Q_5.H \rightarrow Q_1.L$	$CS_{Q5} \rightarrow CS_{Q1}$	1.00	0.8	$\frac{(2-1) \times IW_1 + W_2 \times Conf_2}{n} = \frac{(1) \times 0.72 + (0.8) \times 1.00}{2} \cong 0.76$

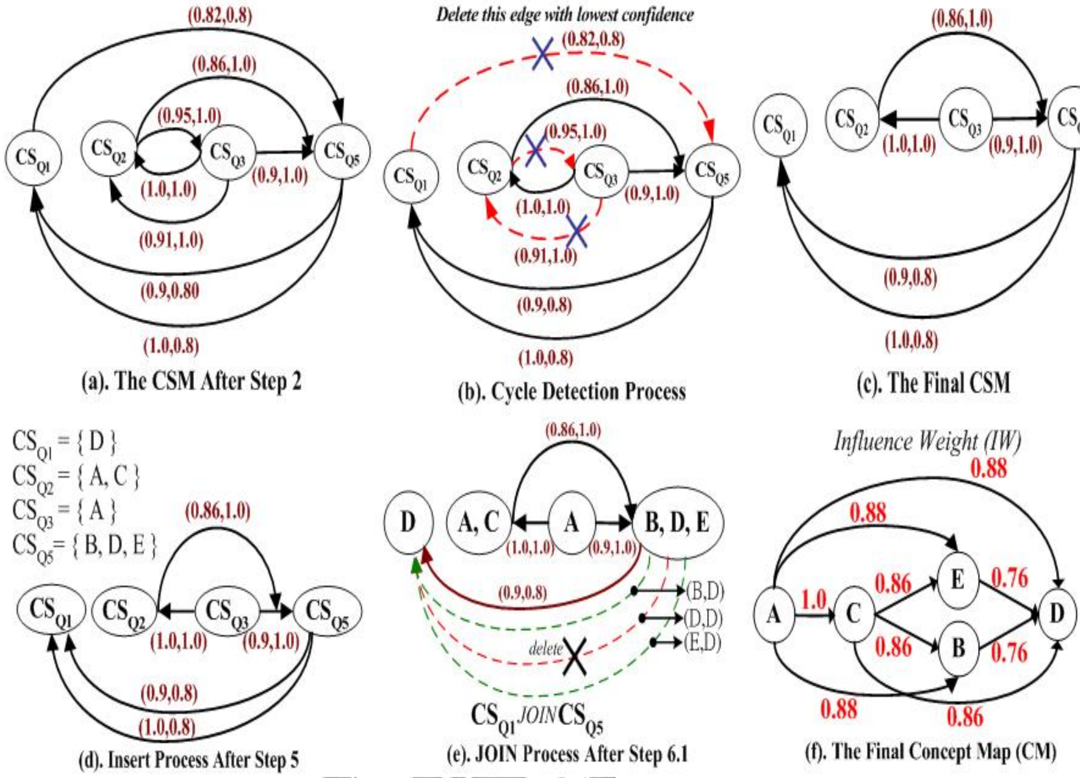


Figure 8.16: The Process of Concept Map Constructing Algorithm

8.2.5 Evaluating the redundancy and circularity of concept map

In this dissertation, creating a concept map without Redundancy and Circularity is our concern. As shown in Figure 8.17, we create three concept maps by using different approaches and evaluate their difference in terms of Redundancy and Circularity. Thus, we use three processing steps including anomaly diagnosis, the prerequisite relationship based upon analyzing L-L or L-L, L-H, H-L, H-H rule types, and cycle detection to create different concept maps. As shown in Figure 8.17, the prerequisite relationship between concept sets in Figure 8.17a is created based upon analyzing L-L rule type only, and Figure 8.17c is created based upon analyzing L-L rule type and anomaly diagnosis we proposed. Then, the concept maps as Figure 8.17b and d are transformed according to the **Test Item-Concept Mapping Table**. Figure 8.17e and f are created by our proposed approach.

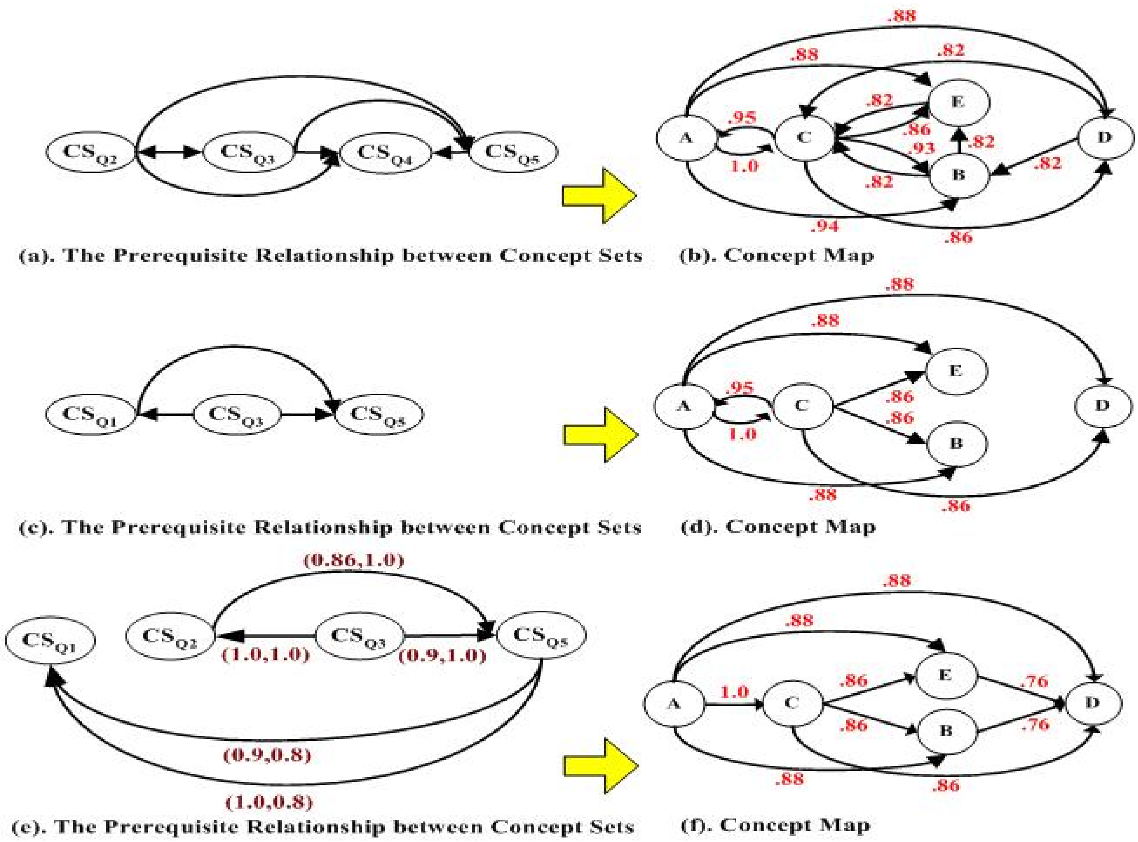


Figure 8.17: The (a) and (b) created based up analyzing L-L rule type only. The (c) and (d) are created based upon Anomaly Diagnosis and analyzing L-L rule type only. The (e) and (f) created by our approach.

Based upon these results of different approaches, the characteristics of approach are concluded as follows.

- **Non-redundancy:** the anomaly diagnosis can filter many useless test items with low discrimination for refining the input data. For example, in Figure 8.17a, the Q_4 with low discrimination results in generating many co-prerequisite links as a cycle in Figure 8.17b.

- **Non-circularity:** the cycle detection process can delete these cycles, e.g., the cycle between A and C in Figure 8.17d, to make the concept map un-ambiguous. Moreover, analyzing association rule with L-L, L-H, H-L, and H-H types can refine the concept map, e.g., the edges $\overrightarrow{ED}$ and $\overrightarrow{BD}$ connect the node D only in Figure 8.17f.

Chapter 9 Implementation and Experimental Results

In order to evaluate proposed ILCMS, several system implementations and experiments have been done in terms of each knowledge module. Also, the experimental results shows that proposed knowledge modules of ILCMS are workable and beneficial for learners and teachers. Thus, the details of implementation and experimental results will be described in following sections.

9.1 Learning Content Editor (LCE) in KA Module

9.1.1 Implementation of Content Transformation Scheme (CTS)

Our content transformation engine was developed based upon the Client/Server architecture, and the related softwares including FreeBSD, MySQL, and Apache server. The index page of our proposed system is shown in Figure 9.1. This system provides teaching material transformation, searching, authoring, management and personal information management. Here, only the transformation result of PowerPoint file is shown in Figure 9.2. As stated previously, first, the authors have to fill the related information of SCORM metadata, and then they need to define the coverage of each section unit (learning object). After the author confirms the definition, the system will extract the learning objects from original PPT file and package these into SCORM compliant teaching material. As shown in the bottom of Figure 9.2, it displays the final teaching material including metadata, table of content (organization), and learning content. Besides, because we have segmented the single PPT file into several learning objects, we can easily retrieve the related media file, including image, audio, video, and hyperlink, etc., of every learning object as shown in right bottom corner of Figure 9.2.

Figure 9.1: The Index Page of CTS System

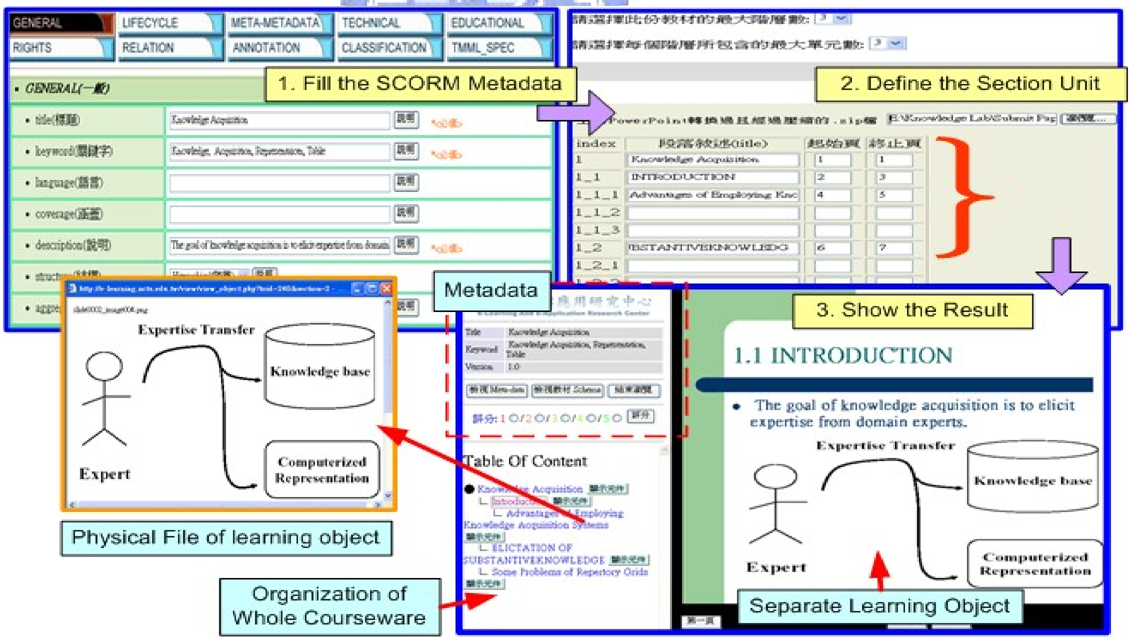


Figure 9.2: The Process of PowerPoint File Transformation.

9.1.2 The Implementation and Evaluation of OOCM Authoring Tool

In this section, based upon the OOCM scheme, a prototypical authoring tool is developed. It can provide users with graphical user interface (GUI) to efficiently construct the learning activity structure with desired sequencing behaviors and then transform learning activity into SCORM compliant course.

The Prototypical Framework of OOCM Authoring Tool:

As shown in Figure 9.3, for constructing a SCORM compliant course, the OOCM authoring tool including 3 functional components, an OOATs Library, and a Learning Object Pool are described as follows:

(1) **Learning Object Importer:** import the existing learning resource within SCORM course or user-defined learning objects into the learning object pool.

(2) **Course Sequencing Constructor:** provide the teacher/instructional designer to construct a complex graph based course structure by inserting OOAT selected from OOATs Library.

(3) **SCORM Content Package Transformer:** transform the graph based course structure into Activity Tree with related sequencing rules and then package its related learning resources into SCORM compliant course, based upon PN2AT and AT2CP algorithms.

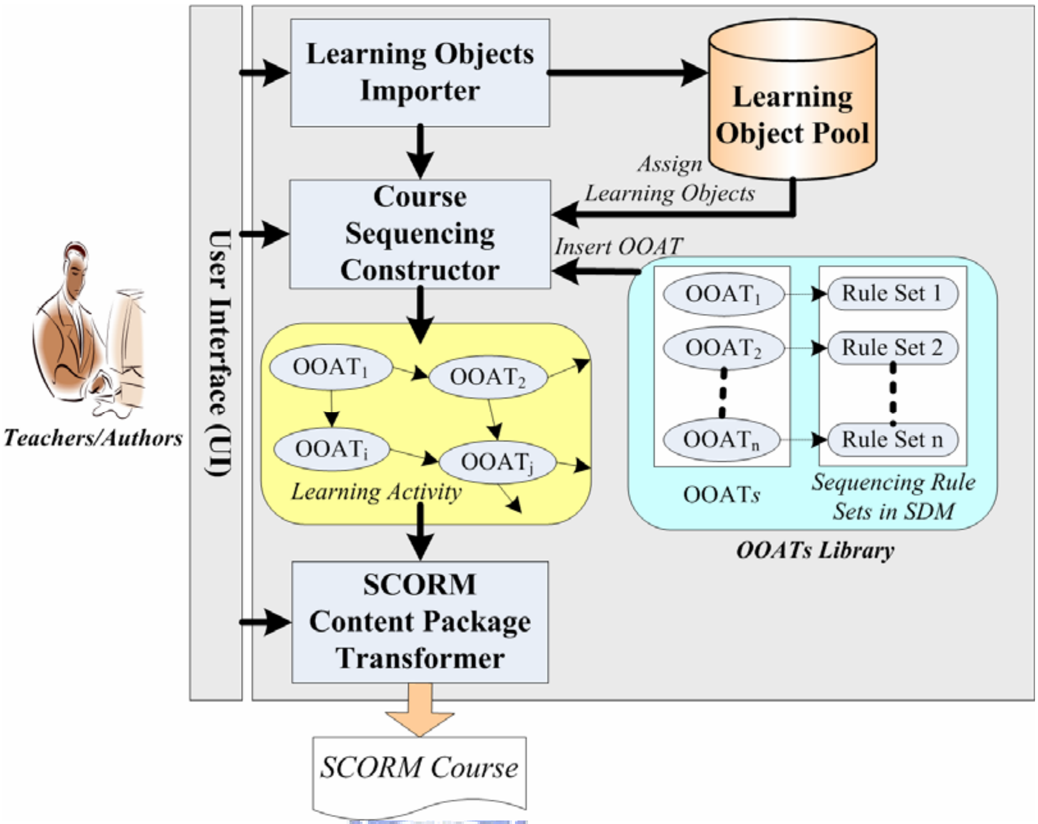


Figure 9.3: The Prototypical Architecture of OOCM Authoring Tool

Here, we describe and show the screenshot of OOCM authoring tool for constructing a SCORM compliant course by OOATs. The Authoring Tool is developed based on Java language and JGraph graphic tool [57] running on Windows operation system. The Figure 9.4 is the screenshot of OOCM authoring tool. The example course of "Photoshop" described in Section 4.2.7 was created by this OOCM authoring tool and executed on the SCORM RTE 1.3 as shown in Figure 9.5. As we see, the table of content in the left side of Figure 9.5 is consistent with the sequencing definition of HLPN in Figure 4.14.a. For example, the Course D (section 3) can not be selected until the test result in Course C satisfies the objective measure in *Global Objective* P_G.

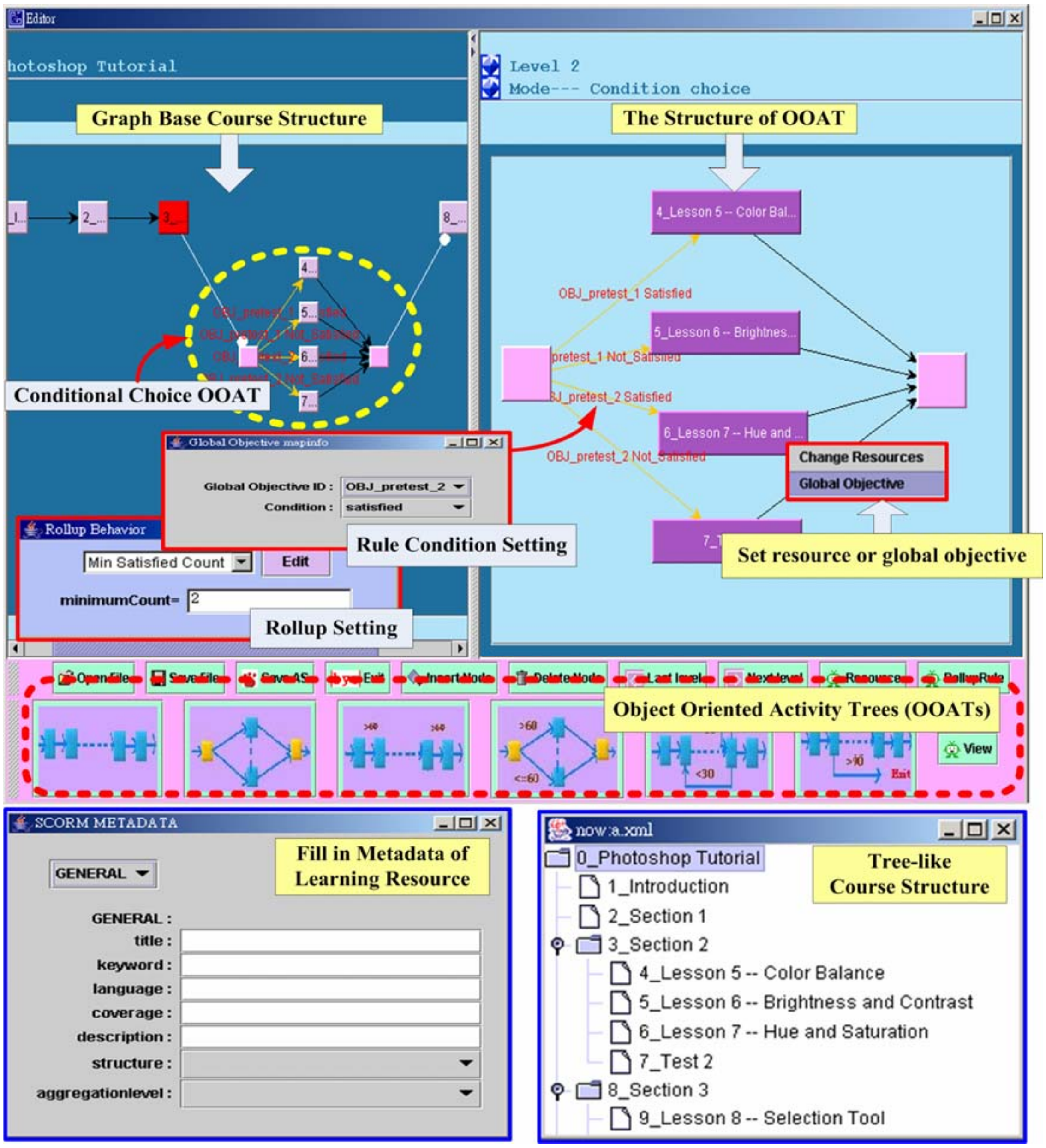


Figure 9.4: The Screenshot of the OOCM Authoring Tool

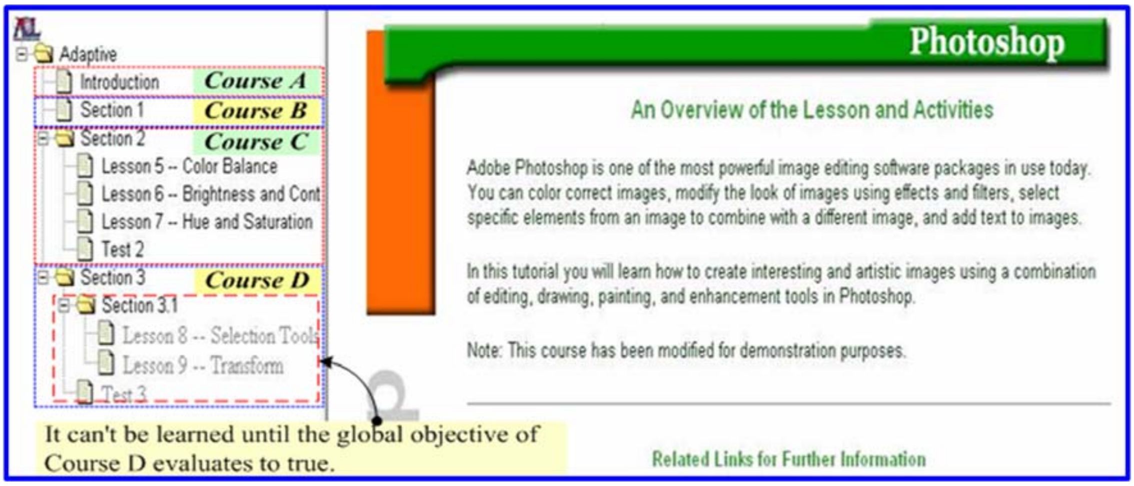


Figure 9.5: The Screenshot of Course “PhotoShop” Executed on SCORM RTE 1.3.

The Evaluation of OOCM approach:

For evaluating the efficiency of the OOCM approach compared with Reload Editor [94], an experiment has been done. The participants of experiment are eight Master students in educational college, which were divided into two groups: one (Experiment Group) used the OOCM authoring tool and the other (Comparison Group) used the Reload Editor. To begin with, everyone in two groups was given 30 minutes to be familiar with these tools and then given the same learning activity with desired sequencing behaviors to create the SCORM course by assigned tool for evaluating the **time cost**. Finally, two groups interchanged the assigned tool to create the same SCORM course for evaluating the **satisfaction degree** by questionnaire. The evaluation results are shown in Figure 9.6. The average time of using OOCM authoring tool is 14 minutes while the average time of using Reload Editor is 32 minutes. Moreover, according to the questionnaire, 1) learning the tool easily, 2) constructing the course without setting the complicated sequencing rules and 3) imagining the final course structure easily are the main advantages of OOCM authoring tool compared with Reload Editor. This shows that the OOCM approach is workable and beneficial for teachers/instructional designers.

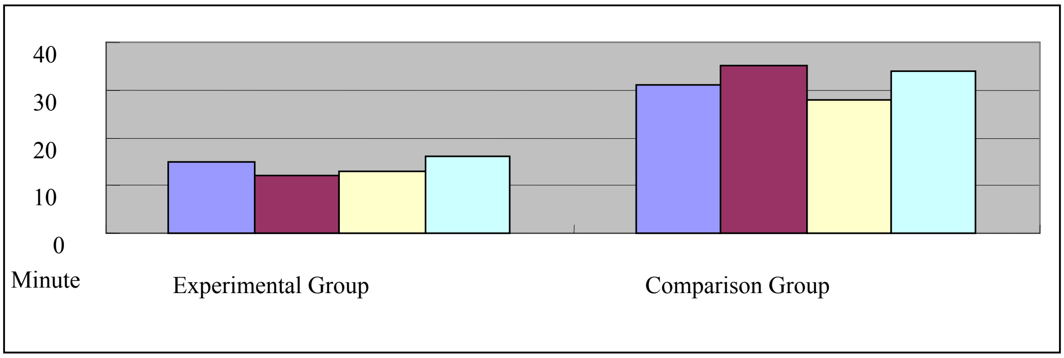


Figure 9.6: The Histogram of the Time Cost

9.2 OOLA Authoring Tool in KA Module

9.2.1 The Implementation of OOLA

The OOLA model is implemented as a user-friendly authoring tool which has several functions, such as adding course, editing rule, editing the item concept, editing item, etc. The learning system is implemented by JAVA and JSP platform. Each student needs to login the system when entering the OOLA system at the first time. Figure 9.7 shows the screenshots of OOLA authoring tool, which is been using to construct a learning activities, including login process, reading learning contents, searching data by the browser, having an examination, and discussing at the chat room. In addition, Figure 9.8 shows that OOLA authoring tool can be used to construct a complex learning activity.

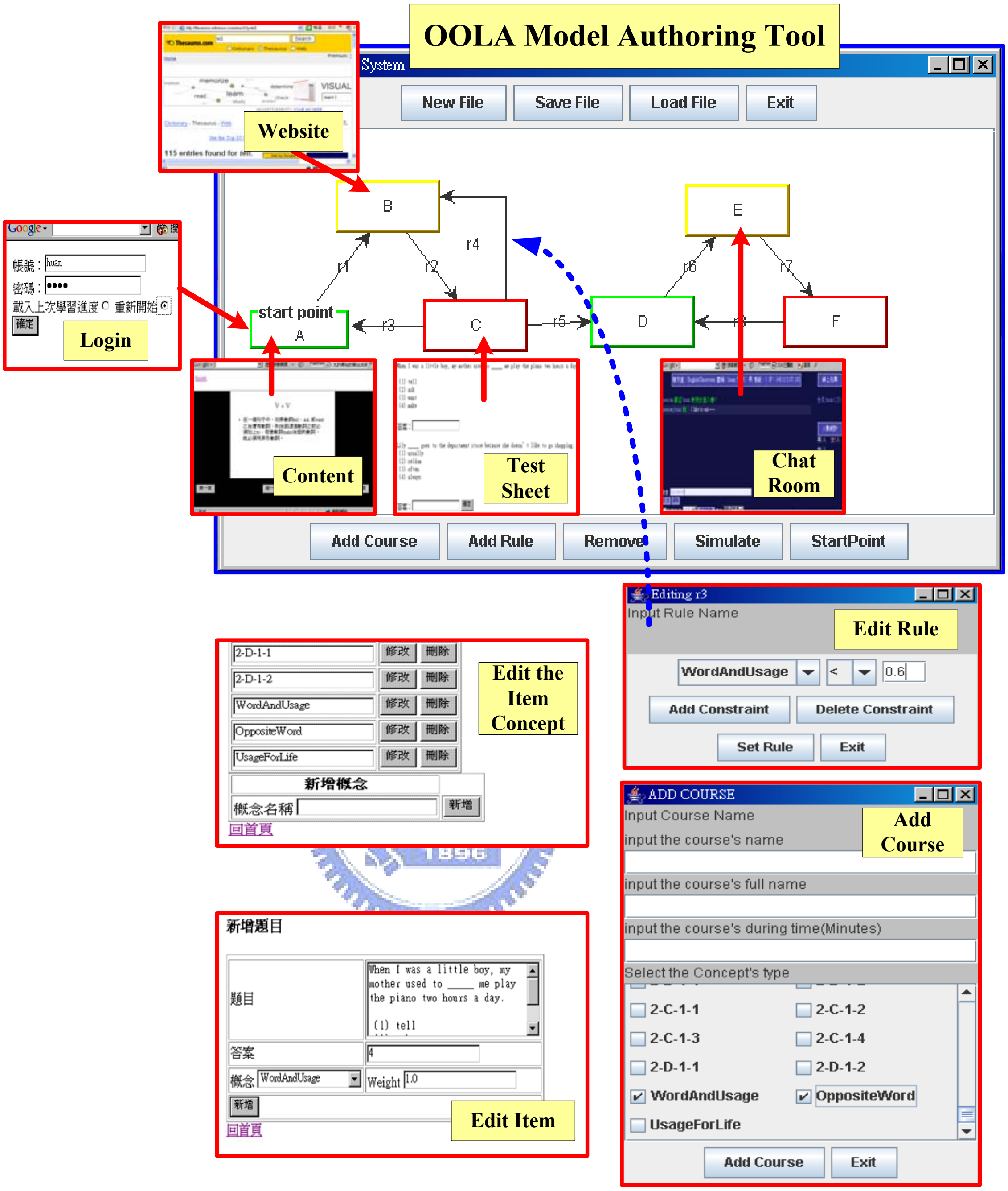


Figure 9.7: The Implementation of OOLA Authoring Tool

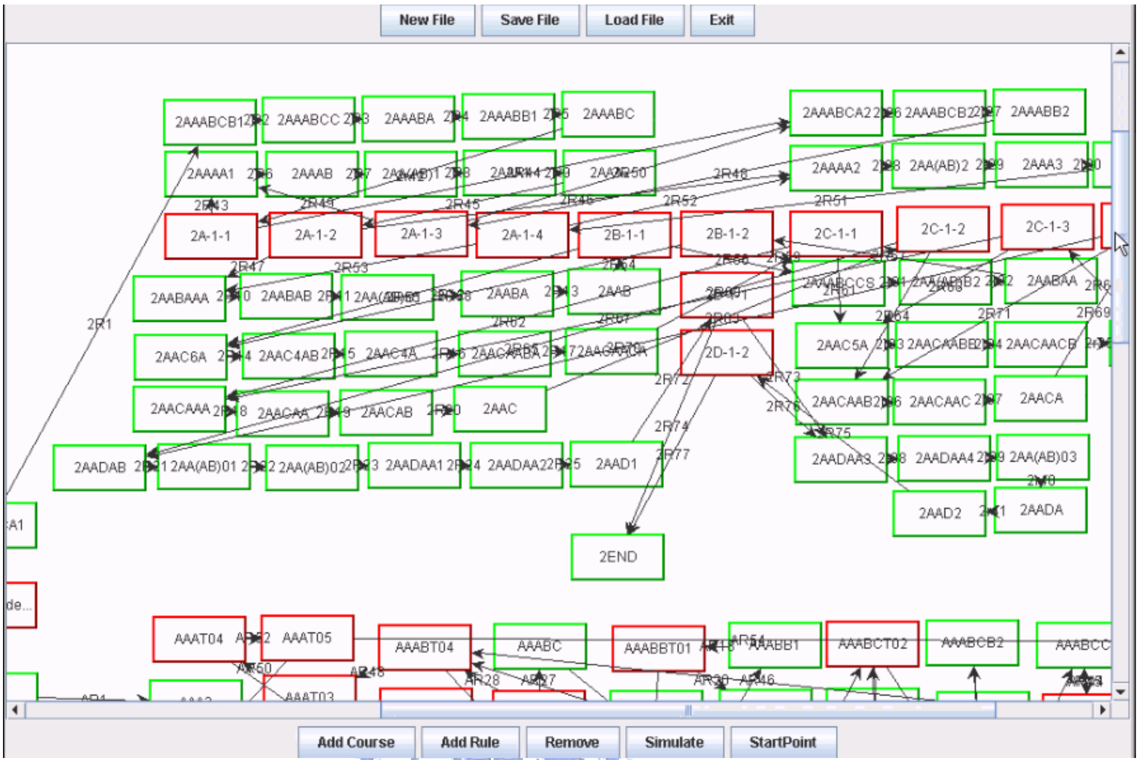


Figure 9.8: The Screenshot of Constructing a complex learning activity by OOLA Tool

9.2.2 The Experimental Results of OOLA

In order to evaluate OOLA model, the teaching strategy of Scaffolding Instruction [91] has been implemented. The scaffolding teaching strategy provides individualized support based on the learner's *zone of proximal development* (ZPD). "*The zone of proximal development is the distance between what children can do by themselves and the next learning that they can be helped to achieve with competent assistance*" [17] [91]. Experiments about learning topic "*The evaporation, condensation and boil of water*" among 4th, 5th, and 6th graders are investigated.

The OOLA design for Scaffolding Instruction:

The OOLA system replicates the actions of a teacher. Its role is to determine the learning achievement level of students by quizzes. With the assessment results, the

OOLA can give them the appropriate learning objects as *Scaffolding Instruction*. An example of learning activity is shown in Figure 9.9. For each learning concept, e.g., C_1, C_{1-1}, C_{1-2} and C_2, the *Scaffolding Instruction* learning activity is starting with the exam assessment activity (N_{EA}) as a pretest. If the student passes the quiz, the OOLA system will guide him/her to learn the next concept. Once the student failed in some exam assessment activities, s/he will receive the corresponding remedial learning object as *Scaffolding Instruction*. The online courses of the subject "*The evaporation, condensation, and boil of water*" are provided to 62 elementary students in Taiwan.

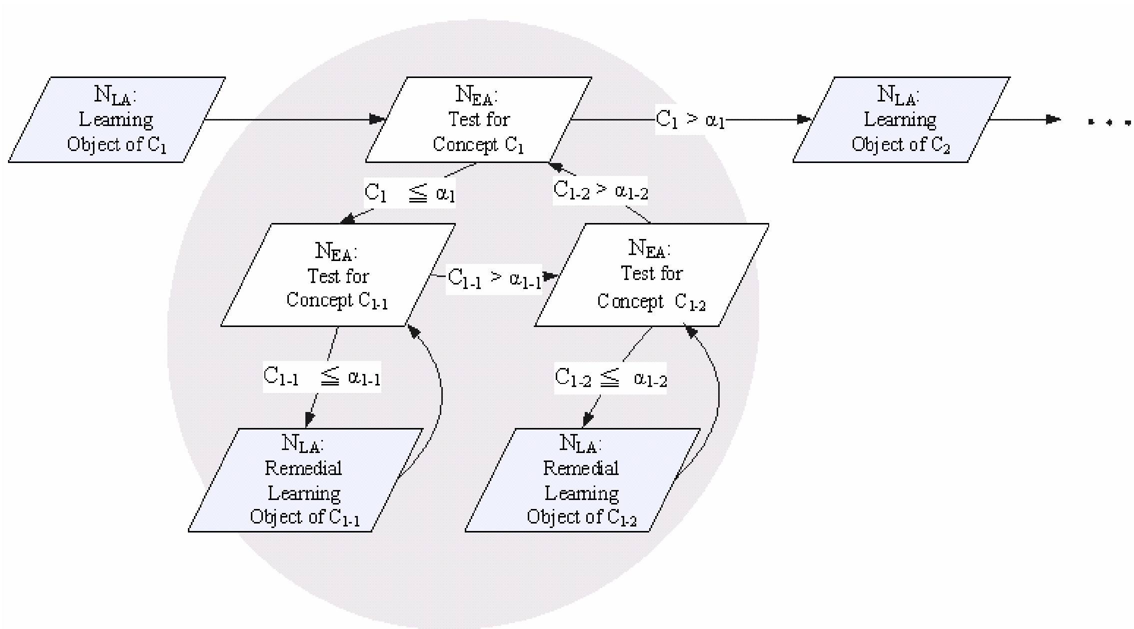


Figure 9.9: Scaffolding Instruction by OOLA

The Analysis of Experimental Results:

To evaluate the effectiveness of OOLA system, we apply the one-group pretest-posttest design for 62 students of 5th graders in a Taiwan elementary school. Firstly, the pretest examination score of concepts of "*The evaporation, condensation, and boil of water*" is the covariate variable. After one month learning with OOLA system, the posttest examination score of the same scope is the dependent variable. Referring to the pretest result, the students are partitioned into high grade group and low grade group. The pairwise t-test and discussion of all students, high grade group and low grade group are as follows.

The pairwise t-test of all students:

Table 9.1: The pretest-posttest of learning achievement

Student Group		Mean	Size	Standard Deviation	Mean difference
Learning	pretest	25.7419	62	3.1516	.4002
Achievement	posttest	28.1290	62	4.1429	.5261

Table 9.2: The one-group pretest-posttest t-test

Pairwise t-test	Variance of Paired Difference			t value	Sig. (2-tailed)
	Mean	Standard Deviation	Standard Error of Mean		
pretest-posttest	2.3871	3.9187	.4977	4.797	.000*

*P < .05

In Tables 9.1 and 9.2, the value t = 4.797 (p value = .000 < .05) shows that the pretest-posttest has significant difference. It deduced that the Scaffolding Instruction designed by OOLA system is effective for students.

The pairwise t-test of High Grade group:

Furthermore, referring to the pretest result, the students are partitioned into high grade group and low grade group. The pairwise t-test in each group is also investigated to analyze the pretest-posttest of learning achievement.

Table 9.3: The pretest-posttest of learning achievement of **high grade group**

Student Group		Mean	Size	Standard Deviation	Mean difference
Learning	pretest	28.3548	31	1.5822	.2842
Achievement	posttest	29.1290	31	3.5846	.6438

Table 9.4: The one-group pretest-posttest t-test of **high grade group**

Pairwise t-test	Variance of Paired Difference			t value	Sig. (2-tailed)
	Mean	Standard Deviation	Standard Error of Mean		
pretest-posttest	.7742	3.5657	.6404	1.209	.236

*P < .05

In Tables 9.3 and 9.4, the value t = 1.209 (p value = .236 > .05) shows that the pretest-posttest doesn't have significant difference. It deduced that the Scaffolding Instruction is not effective for high grade students.

The pairwise t-test of Low Grade group:

Table 9.5: The pretest-posttest of learning achievement of **low grade group**

Student Group		Mean	Size	Standard Deviation	Mean difference
Learning	pretest	23.1290	31	1.8928	.3400
Achievement	posttest	27.6452	31	3.3221	.5967

Table 9.6: The one-group pretest-posttest t-test of **low grade group**

Pairwise t-test	Variance of Paired Difference			t value	Sig. (2-tailed)
	Mean	Standard Deviation	Standard Error of Mean		
pretest-posttest	4.5161	3.6503	.6556	6.888	.000∗

*P < .05

In Tables 9.5 and 9.6, the value t = 6.888 (p value = .000 < .05) shows that the pretest-posttest has significant difference. It deduced that the Scaffolding Instruction designed by OOLA system is effective for low grade students.

After further discussion with students, we found that the high grade students tend to learning by interaction with other students or teachers. Therefore, the lack of instructor to discuss may cause their unobvious learning improvement. On the contrary, the low grade students tend to find the solutions from learning objects. It results in that the Scaffolding Instruction of OOLA system can assist them in finding the learning objects based on their misconception. Therefore, the Scaffolding Instruction of OOLA system is effective especially for low grade students. In the near future, we will enhance OOLA system to be able to support the Scaffold instruction for high grade students.

9.3 Learning Object Repository Manager in KM Module

For evaluating the performance of **Level-wise Content Management Scheme (LCMS)** of LOR Manager in KM Module, in this section, several experiments using synthetic and real SCORM compliant teaching materials have been done.

9.3.1 Synthetic Teaching Materials and Evaluation Criterion

Firstly, we use synthetic teaching materials (TM) to evaluate the performance of our proposed algorithms. All synthetic teaching materials are generated by three parameters: **1) V**: The dimension of feature vectors in teaching materials (TM), **2) D**: the depth of the content structure of TM, **3) B**: the upper bound and lower bound of included sub-section for each section in TM.

In Level-wise Content Clustering Algorithm (LCCAlg), the Single Level Clustering Algorithm (SLCAlg) can be seen as a kind of traditional clustering algorithms. To evaluate the performance, we compare the performance of LCCAlg with SLCAlg which uses the leaf-nodes as input in CTs and doesn't run the concept generation process. The resulted cluster quality is evaluated by the **F-measure** [67] which combines the *precision* and *recall* from the information retrieval. The F-measure is formulated as follows:

$$F = \frac{2 \times P \times R}{P + R}$$

, where P and R are *precision* and *recall* respectively. The range of F-measure is [0,1]. The higher the F-measure is, the better the clustering result is.

9.3.2 Experimental Results of Synthetic Teaching Materials

There are 500 synthetic teaching materials with V=15, D=3, and B = [5, 10] are generated. The clustering thresholds of LCCAlg and SLCAlg are 0.92. After clustering without refinement, there are 101, 104 and 2529 clusters generated from 500, 3664 and 27456 content nodes in the level L_0, L_1, and L_2 of content trees (CTs), respectively. Then, 30 queries generated randomly are used to compare the performance of two clustering algorithms. The **F-measure** of each query with threshold 0.85 is shown in Figure 9.10. Moreover, this experiment is run on AMD Athlon 1.13GHz processor with 512 MB DDR RAM under the Windows XP operating system.

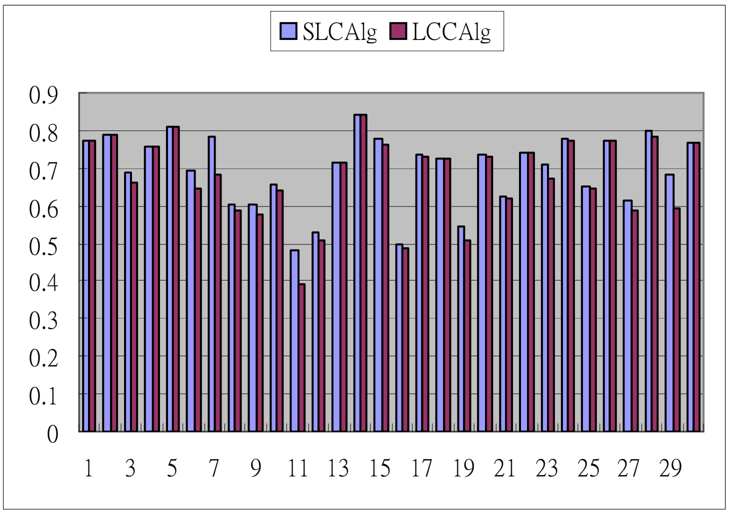


Figure 9.10: The F-measure of Each Query

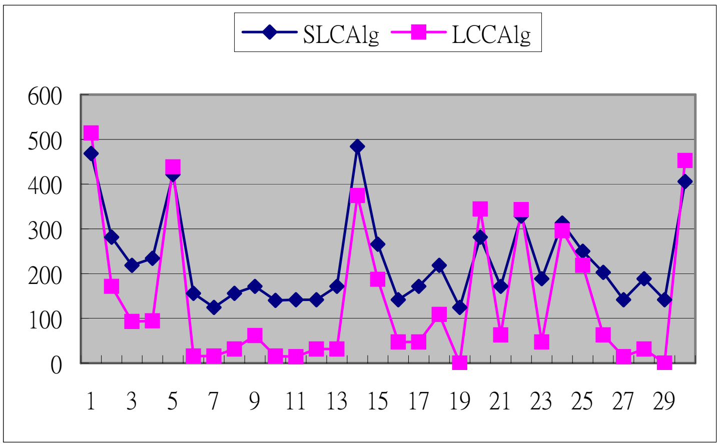


Figure 9.11: The Executing Time Using LCCG-CSAlg

As shown in Figure 9.10, the differences of the F-measures between LCCAlg and SLCAlg are small in most cases. Moreover, in Figure 9.11, the execution time using LCCG-CSAlg in LCCAlg is far less than the time needed in SLCAlg. Figure 9.12 shows that the clustering with clustering refinement can improve the accuracy of LCCG-CSAlg search.

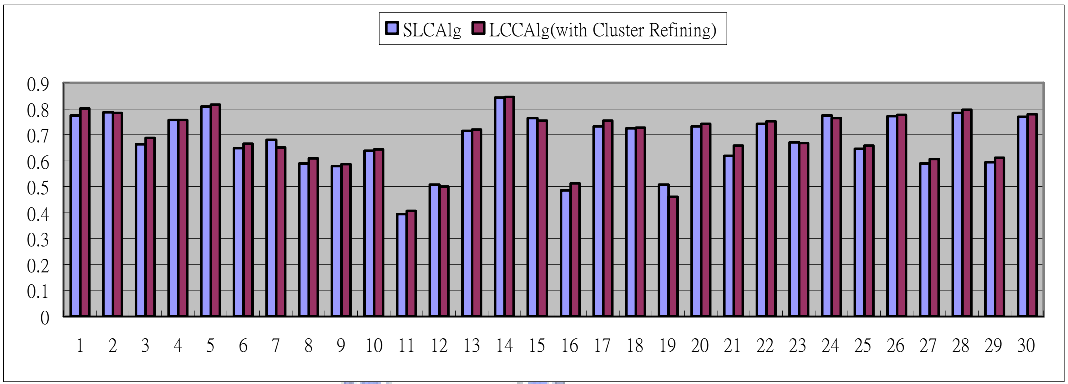


Figure 9.12: The Comparison of SLCAlg and LCCAlg with Cluster Refining

9.3.3 Experiment of Real SCORM Compliant Teaching Materials

As mentioned above, the performance of LCMS scheme evaluated using synthetic teaching materials is efficient. Besides, for evaluating the *Satisfied Degree* of searching results, we also do an experiment using the real SCORM compliant teaching materials. Therefore, we implement a prototype system of LCMS. As shown in Figure 9.13(1), users can first set the searching conditions to retrieve the desired learning contents. Then, all searching results with hierarchical relationships will be shown in Figure 9.13(2). Users can select the link to display the desired learning content as shown in Figure 9.13(3).

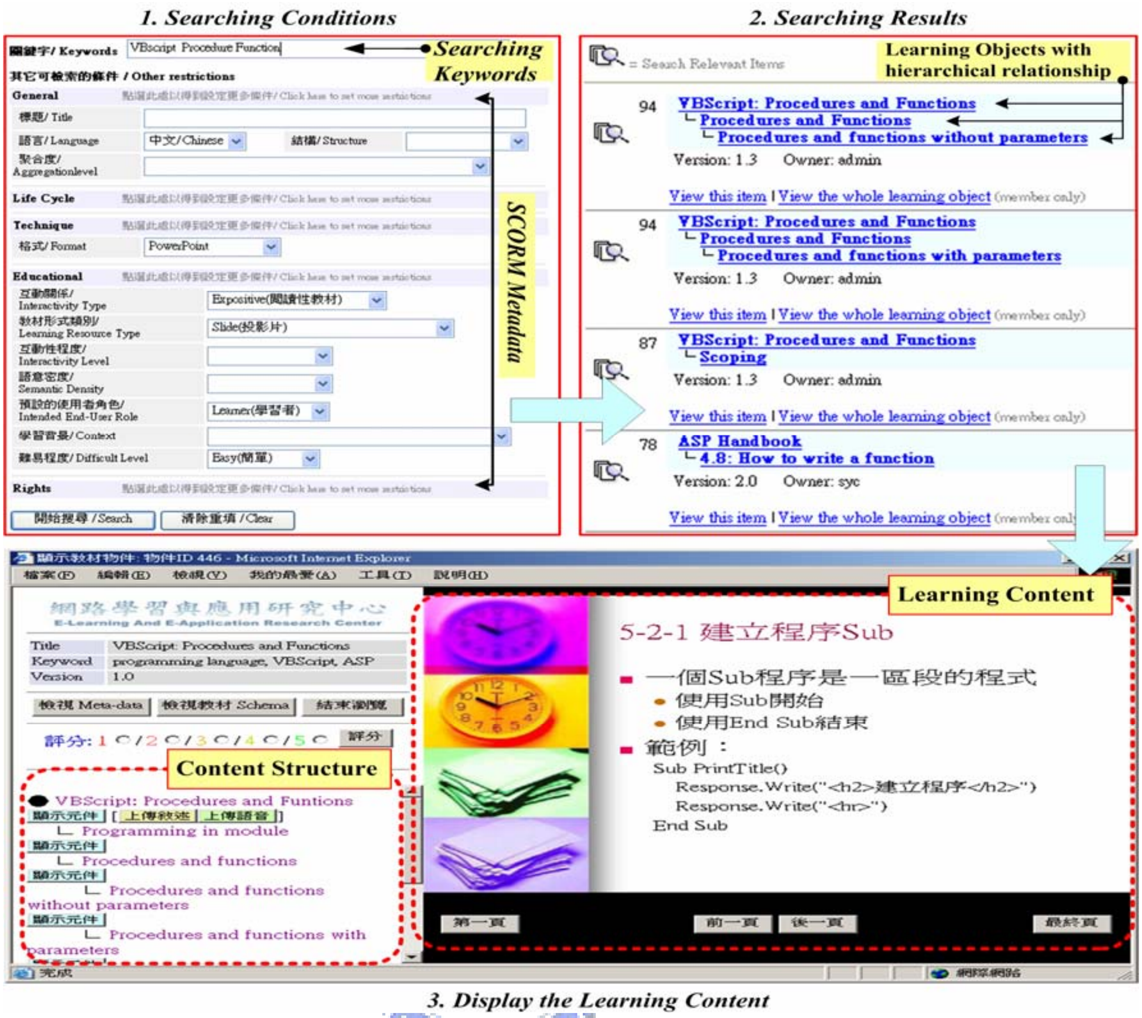


Figure 9.13: The Screenshot of LCMS Prototypical System in KM Module

Then, in this experiment, there are 100 articles with 5 specific topics: *concept learning*, *data mining*, *information retrieval*, *knowledge fusion,* and *intrusion detection*, where every topic contains 20 articles. Every article is transformed into SCORM compliant teaching materials and then imported into the prototype system of LCMS.

In addition, 15 participants, who are graduate students of *Knowledge Discovery and Engineering Lab of NCTU*, used the prototype system of LCMS to query the desired learning contents. Finally, a questionnaire is used to evaluate the performance of LCMS system for these participants. This questionnaire includes the following two questions: **1)** ***Accuracy degree***: "Are these learning objects desired?", **2)** ***Relevance***

degree: “Are the obtained learning objects with different topics related to your query?”. As shown in Figure 9.14, we can conclude that the LCMS scheme is workable and beneficial for users according to the results of questionnaire.

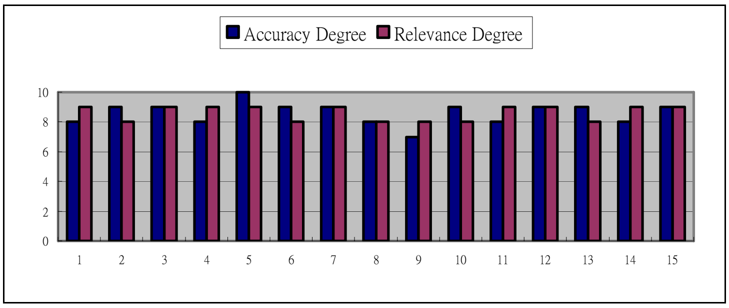


Figure 9.14: The Results of Accuracy and Relevance in Questionnaire (10 is the highest)

9.4 Learning Portfolio Analyzer (LPA) in KMin Module

9.4.1 The Implementation and Evaluation of LPM approach

For evaluating the LPM approach, we implement the LPM system and a training system developed by Java language to gain the training data of LPM system. As shown in Figure 9.15, the training system can let teachers import their learning contents which were organized into hierarchical structure and then display the index of learning contents for learners.

During learning, the training system will display the learning content and the index of its sub-learning contents after each learner chooses an interested content, where the index order is random because learners are apt to choose the top learning content in the index list. Afterward, according to the learning records obtained by training system, we can use the LPM system to generate several personalized SCORM compliant learning course. Figure 9.16 illustrates an example of generated SCORM learning course executed on SCORM Run Time Environment (RTE). In Figure 9.16, the right part shows the learning content and the left part shows the index of contents which a learner can select according to her/his current learning results. Namely, SCORM RTE will automatically control the display of contents according to the associated sequencing rules within the SCORM compliant course. Besides, learners can use the button in the top part to *continue*, *suspend*, or *quit* the learning activity. For example, in Figure 9.16, learners can choice any aggregation in left part to study because of the root aggregation with sequencing rules, "Forward Only=*false*" and "Choice=*true*". For an aggregation with sequencing rules, "Forward Only=*true*" and "Choice= *false*", its included content indexes will be hidden, such as aggregations 1 to 5. In other words, learners have to follow the personalized learning guidance and use the continue button to study next course. In addition, the included contents of aggregation 6 as stated in Activity Tree

Generation Phase can be viewed by learners in any order.

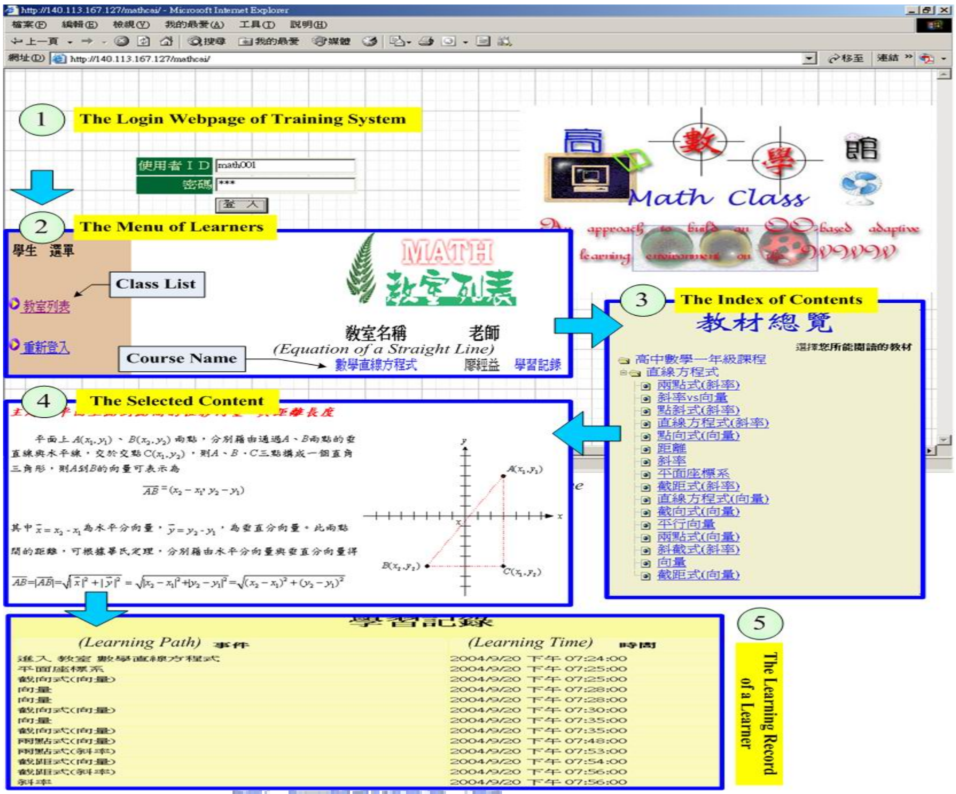


Figure 9.15: The learning process of training system to acquire learners' learning behavior

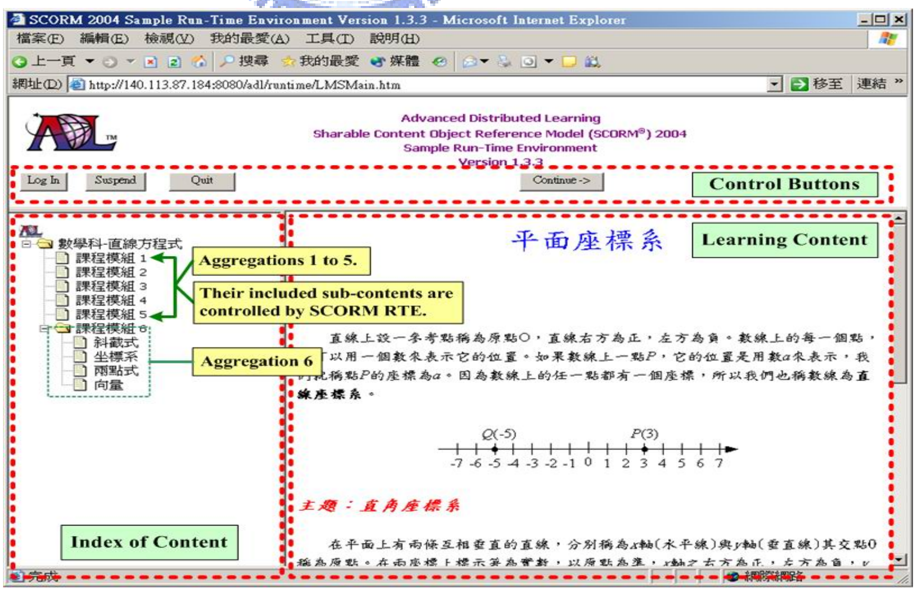


Figure 9.16: The SCORM learning course executed on SCROM run time environment (RTE)

The Experimental Results and Analysis:

To evaluate the efficacy of the LPM approach, an experiment was conducted from September 2004 to November 2004 on the *Equation of a Straight Line* course at a high school in Taiwan. The participants of the experiment are the Ninety students from two equal-sized classes, one is the *control group* A and the other is the *experimental group* B, taught by the same teacher. Before learning, the students in two groups filled out the questionnaire for acquiring their learning characteristics. Then, in Group A, the students use the training system to learn for gathering the training data. Thus, the learning sequences of students with high learning performance in Group A were used to extract the learning patterns, create the decision tree, and generate the activity trees by LPM system. Thus, we gained a decision tree with 5 clusters and 5 personalized activity trees in SCORM compliant course. In Group B, all students were partitioned into 5 groups by the created decision tree and then each group leaned the corresponding SCORM course in SCORM RTE 1.3 as shown in Figure 9.16. Finally, the testing results in Group B were analyzed by t-test approach.

The t-test values of the testing results are listed in Table 9.7. According to the mean value of the testing results, Group B performed better than Group A. By performing the t-test, it is deduced t=3.64, which implies a significant difference between the performance of Groups B and A in the testing Results (where the *t*-value of ‘‘Equal’’ variances is adopted because the ‘Pr>F ’’ value is 0.4302). Therefore, we can conclude that Group B achieved a significant improvement compared with Group A after receiving learning guidance by the personalized SCORM courses.

Table 9.7: *t*-Test of the test results (α=0.05)

Classes	N	Mean	S.D. (Standard Deviation)
Group A	45	66.11	10.86
Group B	45	75	12.24
Improvement (B-A)		8.89	

t-test

Variable	Variances	df	*t* Value
Grade	Equal	88	3.64
Grade	Unequal	86	3.64

Equality of variances

Variable	F Value	Pr>F
Grade	1.27	0.4302

9.4.2 The Experiment of TP-CMC in Physics Course

In this section, we describe our experiment results of the Two-Phase Concept Map Construction (TP-CMC) approach.

Experimental Results:

The participants of experiment are the 104 students of junior high school in Taiwan and the domain of examination is the Physics course. The related statistics of testing results and related concepts of testing paper are shown in Tables 9.8 and 9.9.

The prototype system of TP-CMC is developed based on PHP4 web language, MySQL database, and JGraph web graphic tool [57]. As shown in Figure 9.17a-c, the concept maps with Discrimination 0.0 and 0.3, and 0.5 are created by TP-CMC approach respectively. As mentioned in Section 8.2.3, Anomaly Diagnosis process in TP-CMC can refine the test data for decreasing its redundancy. As we see, the concept maps with low discrimination criteria in Figure 9.17a and b shows that the prerequisite relationships between learning concepts are very disordered and confused. However, with increasing the value of discrimination, the test data can be refined such that the clarity of concept map can be heightened, shown in Figure 9.17c. Moreover, the created concept map can provide the embedded learning information of students during learning Physics. For example, the relationship of concept-pair (6, 9) in Figure 9.17c represents that if students do not learn concept 6 (Speed and direction of motion) well, their learning performance of concept 9 (Change of speed and direction) are most likely bad. Therefore, teachers can modify their teaching strategies to enhance students_ learning performance of concept 6 for getting high performance of concept 9.

Table 9.8: The Related Statistics of Testing Results in Physics Course.

Subject	Information
Educational Degree	Junior High School
The Number of Students	104
Average Score of Exam	61.06
Standard deviation of scores	18.2
The Number of Test Items	50
The Number of Concepts	17

Table 9.9: Concepts List of Testing Paper in Physics Course

Concept ID	Learning Concept
1	Tools and Theories for Timing
2	Unit of Time
3	Isochronism of Pendulum
4	Change of Position
5	Movements
6	Speed and Direction of Motion
7	Average and Instant Speed
8	X- t Diagram
9	Change of Speed and Direction
10	Acceleration
11	Uniform Acceleration
12	Free Fall
13	V- t Diagram
14	The Resultant of Forces
15	Balance of Forces
16	Torque
17	Balance of Rotation

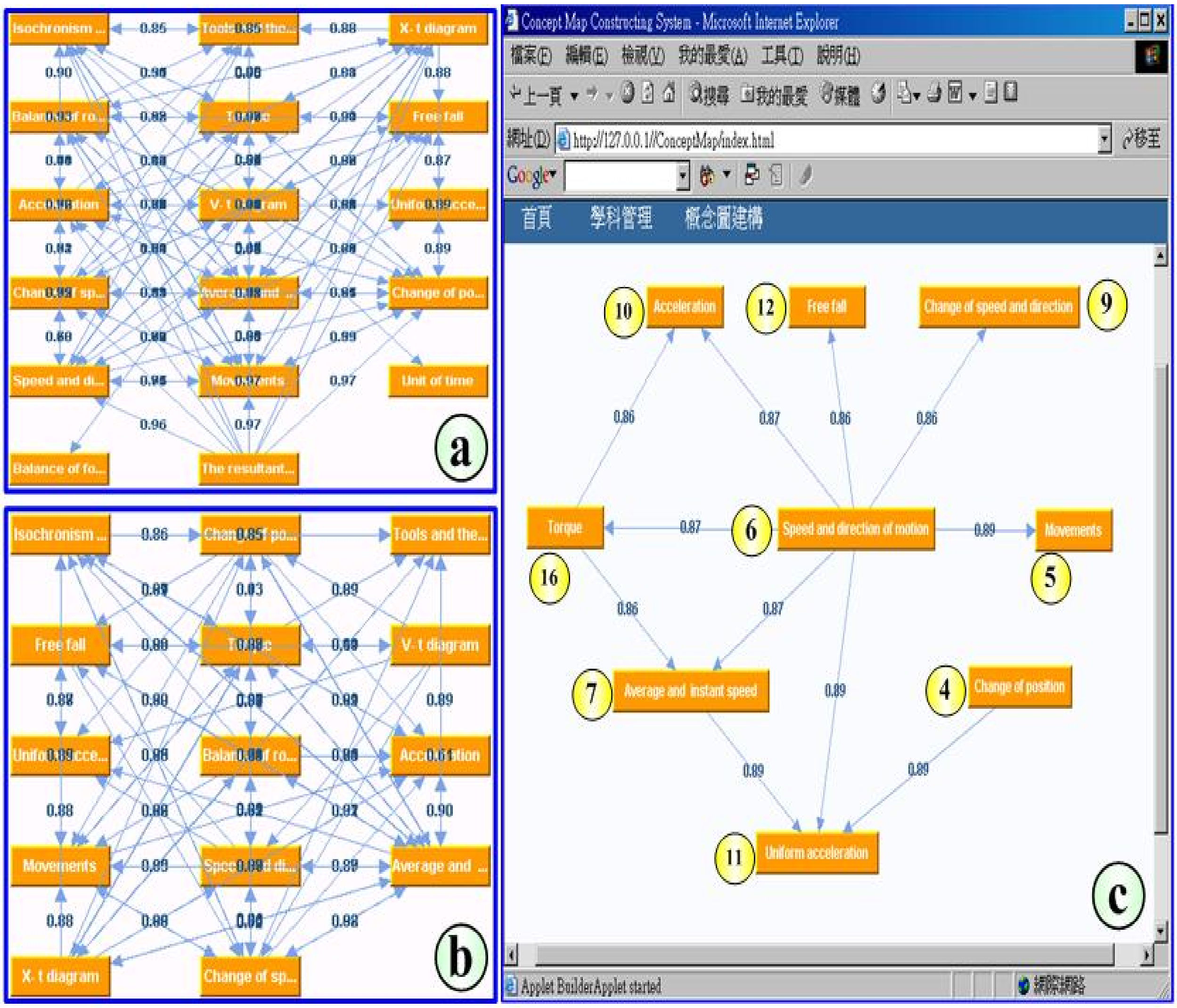


Figure 9.17: The concept maps (a), (b), and (c) with Discrimination 0.0, 0.3, and 0.5 are created by TP-CMC approach respectively. (Support=50, Confidence=0.85)

Chapter 10: Conclusion and Future Work

In this dissertation, based on Knowledge Management concept and LTSA with layering concept, an **Intelligent Learning Content Management System (ILCMS)** is proposed to intelligently manage a large number of learning contents and offer learners an adaptive learning strategy which can be refined by means of efficient learning portfolio analysis. The layered architecture of ILCMS consisting of six knowledge modules: **1) Knowledge Representation (KR)**, which uses SCORM standard, and new proposed Instructional Activity Model (IAM) and Object Oriented Learning Activity (OOLA) model to represent and manage the learning content and activity, **2) Knowledge Resources (KRes)**, which stores all related learning resources in repositories, **3) Knowledge Manager (KM)**, which efficiently manages a large number of learning resources in repositories, **4) Knowledge Acquirer (KA)**, which provides teachers with useful tools to create the SCORM and OOLA compliant learning content and activity, **5) Knowledge Controller (KC)**, which intelligently delivers the desired learning contents, services, test sheet to learners according to her/his learning results and performance, and **6) Knowledge Miner (KMin)**, which analyzes the learning portfolio for constructing the adaptive learning course and the learning concept map automatically.

The relationships of six knowledge modules in ILCMS are described as follows. First, **KRes Module** consists of learning resources: **Learning Activity**, **Learning Object**, **Test Item**, **Application Program**, and **Learning Portfolio**, which are described by data formats: SCORM and OOLA model defined in **KR Module**. Then, **KA** module includes a **Learning Content Editor (LCE)** and an **Object Oriented Learning Activity (OOLA) authoring tool**. In LCE, a **Content Transformation**

Scheme (CTS) has been proposed to divide a traditional teaching material into separate learning objects with SCORM metadata and then package them into one SCORM course. Moreover, an **Object Oriented Course Modeling (OOCM)** approach based upon High Level Petri Nets theory has been proposed to provide teachers to efficiently construct the SCORM compliant course with desired sequencing behaviors. Furthermore, OOLA authoring tool can help teachers construct an OOLA learning activity with desired teaching strategy. Moreover, **KM** module includes a **Learning Object Repository (LOR) Manager**, where we apply clustering approach and load balancing strategies to propose a **Level-wise Content Management Scheme (LCMS)** to efficiently maintain, search, and retrieve the learning contents in SCORM compliant LOR. When learners initiate a learning activity, the **Learning Activity Controller** in **KC** module will retrieve the appropriate learning objects, testing sheets, or application program (AP) according to the personalized learning activity in Learning Activity Repository (LAR) for learners. Furthermore, **KMin** module includes a Learning Portfolio Analyzer (LPA), which consists of **Learning Portfolio Mining** and **Two-Phase Concept Map Construction** algorithm. According to learners' characteristics, the former applies the clustering and decision tree approach to analyze the learning behavior of learners with high learning performance. The latter applies Fuzzy Set Theory and Data Mining approach to automatically construct the concept map by learners' historical testing records. Therefore, after learners finished the learning activities, teachers can use LPA to analyze learning portfolios for refining teaching strategies and contents.

Finally, In order to evaluate ILCMS, several system implementations and experiments have been done for each knowledge module. Also, the experimental results shows that proposed knowledge modules of ILCMS are workable and beneficial for learners and teachers.

Regarding the future work, it can be described in terms of each knowledge module of ILCMS as follows. First of all, for Knowledge Representation (KR), at present, although we have used SCORM to represent the learning content with associated learning objects, proposed a novel model, Instructional Activity Model (IAM), to efficiently manage the activity tree of SCORM, and proposed an Object Oriented Learning Activity (OOLA) model to describe a adaptive learning activity based on IAM concept, it is insufficient to represent the test item in test item bank and learners' information and portfolio in terms of standardization and interoperability. Therefore, in order to share and reuse these data information among different e-learning system, IMS also proposes Question & Test Interoperability (QTI) and Learner Information Package (IMS LIP) specification to support the interoperability of test item and learner information, respectively. In addition, IMS Learning Design (LD) is proposed to describe and design the standardized adaptive learning activity in support interoperability as well. However, although these standards can represent respective learning resource, e.g., QTI for quiz, SOCRM for learning content, LD for learning activity, LIP for learners' information, how to integrate these specifications into complete standard is an important issue. Accordingly, in the future, we can try to design and propose a complete data representation model which can efficiently integrate QTI, SCORM, LD, and LIP. Then, for, Knowledge Acquirer (KA), the functionalities of Learning Content Editor and OOLA authoring tool will be continuously enhanced so that user can use them more efficient and user-friendly. Furthermore, if we want to propose a complete data representation model mentioned above, it is necessary to develop a more powerful and user-friendly authoring tool in support of editing quiz, content, learning activity, learner information in standardized format. For Knowledge Manager (KM), in the future, when the number of learning activities grows quickly and become very large, how to efficiently manage the learning activity repository will become an important issue. Thus, to propose an efficient management scheme of learning activity will be our concerns. Furthermore, for Knowledge Controller (KC), how to propose a more powerful coordination scheme, which can efficiently control and coordinate a large number of concurrent system events and learning requests from system and learners, will get our much attention. Last but not least, for Knowledge Miner (KMin), we will deeply analyze the learning portfolio to find some interesting issues, e.g., learning behavior in collaborative learning environment, the relationship among concept, content, quiz, and teaching strategy, and then propose useful approach or algorithm to solve them. In addition, the layered architecture of ILCMS will be developed based on CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture) of SCORM [27], IMS TIF (Tools Interoperability Framework) [121], and IMS GWS Base Profile (General Web Service Base Profiles) [47] in order to support the interoperability and web services.

References

[1] Aviation Industry CBT Committee (AICC) 2004, AICC - Aviation Industry CBT Committee. http://www.aicc.org

[2] H. Avancini, A. Lavelli, B. Magnini, F. Sebastiani, and R. Zanoli, "Expanding Domain-Specific Lexicons by Term Categorization," *Proceedings of ACM Symposium on Applied Computing*, 2003, pp. 793-797.

[3] Alliance for Remote Instructional and Authoring and Distribution Networks for Europe (ARIADNE) 2004, ARIADNE: Foundation for The European Knowledge Pool. http://www.ariadne-eu.org

[4] R. Aggrawal and R. Srikant, "Mining Sequential Patterns," *Paper presented at the 11th International Conference on Data Engineering (ICDE)*, March 6-10, 1995, Taipei, Taiwan.

[5] J. Appleby, P. Samuels, and T.T. Jones, "Diagnosis–A Knowledge-based Diagnostic Test of Basic Mathematical Skills," *Computers & Education*, Vol. 28, No. 2, 1997, pp. 113-131.

[6] S.M. Alessi and S.R. Trollip, *Computer-based instruction: methods and development 2nd*, Englewood Cliffs, NJ: Prentice-Hall, 1991.

[7] N.L. Bassing, *Teaching in Secondary Schools*, Boston: Houghton Mifflin Company, 1963, pp. 51-55.

[8] E. Bruillard and G.L. Baron, "Computer-based Concept Mapping: A Review of A Cognitive Tools for Students," *In Proceedings of the International Conference on Educational Uses of Communication and Information Technologies*, Beijing, China, 2000, pp. 331-338.

[9] S. Beekhoven, U.D. Jong, and H.V. Hout, "Different Course, Different Students, Same Results? An Examination of Differences in Study Progress of Students in Different Courses," *Higher Education*, Vol. 46, 2003, pp. 37-59.

[10] B.S. Bloom, B.B. Mesia, and D. R. Krathwohl, *Taxonomy of Educational Objectives (two vols: The Affective Domain & The Cognitive Domain),* New York. David McKay, 1964.

[11] G. Booch, *Object-Oriented Design with Applications*, San Francisco, 1991.

[12] J.S. Bruner, *The Process of Education*, Cambridge, MA: Harvard University Press, 1960.

[13] J.J. Beishuizen and E.T. Stoutjesdijk, "Study strategies in a computer assisted study environment," *Int'l Journal of Learning and Instruction*, Vol. 9, 1999, pp. 281-301.

[14] P. Brusilovsky and J. Vassileva, "Course Sequencing Techniques for Large-Scale Web-based Education," *Journal of Engineering Education and Lifelong Learning,* Vol. 13, 2003, pp. 75-94.

[15] C.K. Chang, G.D. Chen and K.L. Ou, "Student Portfolio Analysis for Decision Support of Web Based Classroom Teacher by Data Cube Technology," *Journal of Educational Computing Research*, Vol. 19, No. 3, 1998.

[16] G.D. Chen, C.C. Liu, K.L. Ou and B.J. Liu, "Discovering Decision Knowledge From Web Log Portfolio for Managing Classroom Processes by Applying Decision Tree and Data Cube Technology," *Journal of Educational Computing Research*, Vol. 23, No. 3 , 2000, pp. 305-332.

[17] K. Chang, I. Chen, et al. (2002), "The effect of concept mapping to enhance text comprehension and summarization," *The Journal of Experimental Education*, Vol. 71, No. 1, pp. 5-23.

[18] CETIS 2004, 'ADL to make a 'repository SCORM'', The Centre for Educational Technology Interoperability Standards . http://www.cetis.ac.uk/content2/20040219153041

[19] S. Chen, "A Congitive Model for Non-Linear Learning in Hypermedia Programmes," *British Journal of Educational Technology*, Vol. 33, No. 4, 2002, pp.449-460.

[20] C. Chou, "A computer logging method for collecting use-reported inputs during formative evaluation of computer network-assisted distance learning," *Proc. of ED-Media'96*, 1996.

[21] K.E. Chang, S.H Liu, and S.W. Chen, "A Testing System for Diagnosing Misconceptions in DC Electric Circuits," *Computers & Education*, Vol. 31, No. 2, 1998, pp. 195-210.

[22] V. Carchiolo, A. Longheu, and M. Malgeri, "Adaptive Formative Paths in a Web-based Learning Environment," *Educational Technology & Society*, Vol. 5, No. 4, 2002.

[23] Click2Learn 2004, Click2learn Unveils SCORM 1.2 Resource Kit', SumTotal. Retrieved 4 September 2004 from http://www.sumtotalsystems.com/archived/pressrelease_c2l.html?cid=40

[24] D.R. Cutting, D.R. Karger, J.O. Predersen, J.W. Tukey, "Scatter/Gather: A

Cluster-based Approach to Browsing Large Document Collections," *Proceedings of the Fifteenth Interntional Conference on Research and Development in Information Retrieval*, 1992, pp. 318-329.

[25] W. Chen, and R. Mizoguchi., "Communication Content Ontology for Learner Model Agent in Multi-agent Architecture," *Advanced Research in Computers and Communications in Education*, *Proc. ICCE'99*, 1999, pp. 95-102.

[26] L. Curry, "Patterns of learning styles across selected medical specialties," *Educational Psychology*, Vol. 11, 1991, pp. 247-277.

[27] Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) (2005), CORDRA: Content Object Repository Discovery and Registration/Resolution Architecture: Learning Systems Architecture Laboratory, http://cordra.lsal.cmu.edu/cordra/

[28] DLTS (Distance Learning Technology Standards), http://www.xauat.edu.cn/dlts/

[29] D.G. Dewhurst, H.A. Macleod, and T.A.M. Norris, "Independent Student Learning Aided by Computers: An Acceptable Alternative to Lectures?," *Computer and Education*, Vol.35, 2000, pp. 223-241.

[30] F. Debole and F. Sebastiani, "Supervised Term Weighting for Automated Text Categorization," *Proceedings of ACM Symposium on Applied Computing*, 2003, pp. 784-788.

[31] Elicitus Content Publisher, (ECP) 2004, Elicitus Content Publisher – Create more learning with less authoring. Retrieved 4 September 2004 from http://www.elicitus.com/home.htm

[32] e-learning Centre, 2004, e-learning Centre: Course Authoring Tools. Retrieved 4 September 2004 from http://www.e-learningcentre.co.uk/eclipse/vendors/authoring.htm

[33] EDUCAUSE, http://www.educause.edu/

[34] A.D. Efland, "The Spiral and the Lattice: Changes in Cognitive Learning Theory with Implications," *Studies in Art Education*, Vol. 36, No. 3, 1995, pp. 134-153.

[35] E-Learning Framework (2005), The E-Learning Framework, http://www.elframework.org/

[36] U. Fayyad, "Data Mining and Knowledge Discovery in Database: Implications for Scientific Databases," *Proceedings of Ninth International Conference on Scientific and Statistical Database Management*, 1997, pp. 2-11.

[37] S. Fischer, "Course and exercise sequencing using metadata in adaptive hypermedia learning systems," *ACM Journal of Educational Resources in Computing*, Vol. 1, No. 1, Spring 2001.

[38] G. Frosini, B. Lazzerini, and F. Marcelloni, "Perform Automatic Exams," *Computers & Education*, 31(3), 1998, pp. 281-300.

[39] G. Fischer and J. Ostwald, "Knowledge management: problems, promises, realities, and challenges," *IEEE Intelligent Systems*, Vol. 16, No. 1, 2001.

[40] R.M. Gagne, "Learning Outcomes and Their Effects: Useful Categories of Human Performance," *American Psychologist*, Vol. 39, No. 4, 1984, pp. 377-385.

[41] H. Gamboa, "Designing Intelligent Tutoring Systems: A Bayesian Approach," *In Proceedings of the Ana Fred 3rd International Conference on Enterprise Information Systems (ICEIS'2001)*, 2001, pp. 452-458.

[42] R.M. Gagné, L. Briggs and W. Wager, *Principles of Instructional Design*, 4th ed. Harcourt Brace Jovanovich, 1992.

[43] J.E. Gilbert and C.Y. Han, "Adapting Instruction in Search of 'A Significant Difference'," *Journal of Network and Computer Application*, Vol. 22, 1999, pp. 149-160.

[44] V. Guralnik and G. Karypis, "A scalable algorithm for clustering sequential data," *Paper presented at the 2001 IEEE International Conference on Data Mining*, November 29 - December 2, 2001, San Jose, California, USA.

[45] J.L. Gordon, "Creating Knowledge Maps by Exploiting Dependent Relationships," *Knowledge Based System*, Vol. 13, 2000, pp. 71-79.

[46] B.R. Gaines and M.L.G. Shaw, "Concept Maps as Hypermedia Components," *International Journal of Human-Computer Studies*, Vol. 43, No. 3, 1995, pp. 323-361.

[47] General Web Services (2005), IMS General Web Services Public Draft Specification, http://www.imsglobal.org/gws/index.html

[48] D.J. Hall and G.B. Ball, "ISODATA: a novel method of data analysis and pattern classification. *Technical report, Stanford Research Institute*, Menlo Park, CA, USA, 1965.

[49] 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, No. 2, 2003, pp. 229-248.

[50] P. Honey and A. Mumford, *The Manual of Learning Styles*, 10 Linden Avenue,

Maidenhead: Peter Honey, 1986.

[51] C.S. Hsu, S.F. Tu, and G.J. Hwang, "A Concept Inheritance Method for Learning Diagnosis of a Network-based Testing and Evaluation System," *In Proceedings of the 7th International Conference on Computer-Assisted Instructions*, 1998, pp. 602-609.

[52] G.J. Hwang, "A tutoring strategy supporting system for distance learning on computer networks," *IEEE Trans. on Education*, Vol. 41, No. 4, pp. 9, 1998.

[53] G.J. Hwang, "A knowledge-based system as an intelligent learning advisor on computer networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 2, Tokyo, Japan, 1999, pp. 153–158.

[54] G.J. Hwang, "A Conceptual Map Model For Developing Intelligent Tutoring System," *Computers & Education*, Vol. 40, No. 3, 2003, pp. 217-235.

[55] IMS Abstract Framework (IAF) (2005), IMS Abstract Framework (IAF), http://www.imsglobal.org/af/index.html

[56] Instructional Management System (IMS) 2004, IMS Global Learning Consortium. http://www.imsproject.org/

[57] JGraph (2004), Open-Source Graph Component Available for Java, Retrieved April 21, 2004 form http://www.jgraph.com/.

[58] E.R. Jones 2004, Dr. Ed's SCORM Course, http://www.scormcourse.jcasolutions.com/index.php

[59] K. Jensen, "*Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use,"* Monographs in Theoretical Computer Science, Springer-Verlag, 1997.

[60] K. Jensen, "*An Introduction to the Theoretical Aspects of Coloured Petri Nets*," In: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (eds.): A Decade of Concurrency, Lecture Notes in Computer Science, Vol. 803, Springer-Verlag, 1994, pp. 230-272.

[61] W.B. Johnson, L.O. Neste, and P.C. Duncan, "An authoring environment for intelligent tutoring systems," *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, 1989, pp. 761-765.

[62] K. Jensen and G. Rozenberg, "*High-level Petri Nets," Theory and Application*. Springer-Verlag Publishers, 1991.

[63] S.K. Ko and Y.C. Choy, "A Structured Documents Retrieval Method supporting Attribute-based Structure Information,“ *Proceedings of the 2002 ACM symposium on Applied computing*, 2002, pp. 668-674.

[64] L. Kause and C. Fallon, 2004, 'Creating E-Learning Content in Authorware 7 for

SCORM1.2-Compliant LMSs and LCMSs', *Macromedia-Solutions: Creating E-Learning Content in Authorware 7 for SCORM1.2-Compliant LMSs*. Retrieved 4 September 2004 from http://www.macromedia.com/resources/elearning/article/lo_packager01/

[65] D.A. Kolb, "Learning Style Inventory: Technical Manual", Boston, Mass: McBer originally published in 1976, 1985.

[66] David Kolb's learning styles model 2004, Kolb's learning styles, Retrieved September 22, 2005 from http://www.businessballs.com/kolblearningstyles.htm.

[67] B. Larsen and C. Aone, "Fast and Effective Text Mining Using Linear-Time Document Clustering," *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 16-22.

[68] Y.C. Lei, A Level-wise Clustering Algorithm on Structured Documents, *Master Thesis*, Department of Computer Information Science, National Chiao Tung University, Taiwan, 2003.

[69] S. Labidi and J. S. Ferreira, "Technology-assisted instruction applied to cooperative learning: The SHIECC project," in *Proc. 28th Ann. Frontiers Educ. Conf.*, vol. 1, Tempe, AZ, Nov. 4–7, 1999, pp. 286–291.

[70] F.H. Lin, "Modeling Online Instruction Knowledge for Virtual Training Systems using Petri Nets," *Proceedings of IEEE Pacific Rim Conference on Communications*, Computers and Signal Processing, Victoria, B.C., Canada, Vol. 1, ISBN: 0-7803-7080-5, 2001, pp. 212-215.

[71] X.Q. Liu, et al, "Knowledge Aggregation And Navigation High-Level Petri Nets-based in E-learning," *Proceedings of the First International Conference on Mache Learning and Cybernetics,* Beijing, China, 2002, pp. 420-425.

[72] M. Liu, "The Relationship Between The Learning Strategies and Learning Styles in a Hypermedia Environment," *Computers in Human Behaviour*, Vol. 10, No. 4, 1994, pp. 419-434.

[73] J. Lee and L.F. Lai, "A High-Level Petri Nets Based Approach to Verifying Task Structures," *IEEE Transactions on Knowldge and Data Engineering*, Vol. 14, No. 2, 2002, pp. 316-335.

[74] LMML (Learning material markup language), http://daisy.fmi.uni-passau.de/pakmas/lmml/

[75] H.V. Leong, D. MeLeod, A. Si, and S.M.T. Yau, "On Supporting

Weakly-Connected Browsing in a Mobile Web Environment," *Proceedings of ICDCS2000*, 2000, pp: 538-546.

[76] LSAL 2003, 'SCORM Best Practices Guide for Content Developers', *Learning Systems Architecture Laboratory: Carnegie Mellon LSAL.* http://www.lsal.cmu.edu/lsal/expertise/papers/index.html

[77] LSI (2004). *Kolb's learning styles: Learning Style Indicator (LSI)*, retrieved September 22, 2005 from http://chat.carleton.ca/~tblouin/Kolb's%20Leaning%20Styles%20Model/kolbs%20test.htm.

[78] 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, 2003, pp. 369-385.

[79] IEEE Learning Technology Standards Committee (LTSC) 2004, IEEE LTSC | WG12. http://ltsc.ieee.org/wg12/

[80] Learning Technology Systems Architecture (LTSA) 2001, " LTSA specification LTSA Draft 9, "IEEE LTSC | WG1, Retrieved 4 September May 2005 from http://ltsc.ieee.org/wg1/materials.html

[81] H.Y. Lin, S.S. Tseng, J.M. Su, and J.F. Weng, "Design and Implementation of an Object Oriented Learning Activity System," to appear in t*he 10th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI) 2006*, Orlando, Florida, USA, July, 2006.

[82] X. Li and W. Yu, "Object Oriented Fuzzy Petri Net for Complex Knowledge System Modeling," *Proceedings of the 2001 IEEE International Conference on Control Applications,* 2001, pp. 476-481.

[83] D. McIlroy, B. Bunting, K. Tierney and M. Gordon, "The Relation of Gender and Background Experience to Self-reported Computing Anxieties and Cognitions," *Computer in Human Behavior*, Vol.17, 2001, pp. 21-33.

[84] T. Murata, "*Petri Nets: Properties, Analysis and Applications*," *Proceedings of the IEEE,* Vol. 77, No. 4, 1989, pp. 541-580.

[85] J.D. Novak, (1998). *Learning, Creating, and Using Knowledge: Concept Maps As Facilitative Tools in Schools and Corporations.* Lawrence Erlbaum Assoc.

[86] B. Oakley, "A virtual classroom approach to teaching circuit analysis," *IEEE Trans. on Education*, Vol. 39, No. 3, 1996, pp. 287-296.

[87] K.A. Papanikolaou, M. Grigoriadou, G.D. Magoulas, and H. Kornilakis, "Towards

New Forms of Knowledge Communication: the Adaptive Dimension of a Web-based Learning Environment," *Computers & Education*, Vol. 39, Issue 4, December 2002, pp. 333-360.

[88] J. B. Pugliesi and S.O. Rezende, "Intelligent hybrid system for a training and teaching environment," in *Proc. 3rd Int. Conf. Comput. Intell. Multimedia Applicat.*, 1999, pp. 148–152.

[89] W.J. Popham, "*Classroom Assessment: What Teachers Need to Know*," Pearson Allyn & Bacon, 1999, pp. 222-227.

[90] G.J. Posner, *Analyzing the Curriculum*, N.Y.: McGraw-Hill, 1992.

[91] E. Raymond, " *Cognitive Characteristics. Learners with mild disabilities : a characteristics approach*," Allyn & Bacon, 2000, pp. 169-201.

[92] J.R. Rumbaugh, M.R. Blaha, et al., "*Object-Oriented Modeling and Design," 1990.*

[93] R. Riding and I. Cheema, "Cognitive Styles—An Overview and Integration," *Educational Psychology*, Vol. 11, No. 3-4, 1991, pp.193-215.

[94] Reload Editor (Reload) 2004, Reload Project, http://www.reload.ac.uk

[95] J. Quinlan, "Induction of Decision Trees," *Machine learning*, Vol.1, 1986, pp.81-106.

[96] V.V. Raghavan and S.K.M. Wong, "A Critical Analysis of Vector Space Model in Information Retrieval," *Journal of the American Soczety for Information Science*, 37, 1986, pp. 279-287.

[97] R. Srikant and R. Aggrawal, "Mining Sequential Patterns: generalizations and performance improvements," *Paper presented at the Fifth International Conference on Extending Database Technology*, March 25-29, 1996, Avignon, France.

[98] L. Sheremetov and A.G. Arenas, "EVA: An Interactive Web-based Collaborative Learning Environment," *Computers & Education*, Vol. 39, Issue 2, 2002, pp. 161-182.

[99] C.T. Sun and C. Chou, "Experiencing CORAL: design and implementation of distance cooperative learning," *IEEE Transactions on Education*, Vol. 39, No. 3, 1996, pp. 357-366.

[100] Sharable Content Object Reference Model (SCORM) 2004, Advanced Distributed Learning. http://www.adlnet.org/

[101] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, Vol. 34, No. 1, 2002, pp. 1-47.

[102] R. Sternberg and E. Grigorenko, *Styles of Thinking in the School*, Branco Weiss Institute, May 1998.

[103] T.K. Shih, J.C.S. Hung, W.C. Ko, W.C. Chang, and N.H. Lin, "Collaborative Courseware Authoring Based On SCORM Metadata," *Proceedings of the IEEE International Conference on Multimedia & Expo 2003 (ICME 2003)*, Taipei, Taiwan, July, 2003. Retrieved 4 September 2004 from http://www.mine.tku.edu.tw/scorm/

[104] L. Shashaani and A. Khalili, "Gender and Computers: Similarities and Differences in Iranian College Students' Attitudes Toward Computers," *Computers & Education*, Vol.37, 2001, pp. 363-375.

[105] Seminar Learning System, (SLS) 2004**,** Seminar Learning System - Information Transfer. Retrieved 4 September 2004 from http://www.seminar.co.uk/author/author.cfm

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

[107] G. Shaw, N. Marlow, "The Role of Student Learning Styles, Gender, Attitudes and Perceptions on Information and Communication Technology Assisted Learning," *Computers & Education*, Vol. 33, Issue 4, December 1999, pp. 223-234.

[108] E. Sadler-Smith, "The Relationship between Learning Style and Cognitive Style," *Personality and Individual Differences*, Vol.30, 2001, pp. 609-616.

[109] Sequencing and Navigation (SN) 2004, 'Sharable Content Object Reference Model (SCORM) Sequencing and Navigation (SN) Version 1.3', Advanced Distributed Learning. http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false

[110] S.S. Tseng, P.C. Sue, J.M. Su, J.F. Weng, and W.N. Tsai, "A New Approach for Constructing the Concept Map," to appear in *Journal of Computers & Education*. (Article In Press) **(SSCI, SCI, EI)**

[111] Y. Shang, H.C. Shi, and S.S. Chen, "An Intelligent Distributed Environment for Active Learning," *ACM Journal of Educational Resources in Computing*, Vol. 1, Issue 2, 2001, pp. 4-17.

[112] K. Smith and H. Tillema, "Evaluating Portfolio Use As A Learning Tool For Professionals," Scandinavian Journal research, Vol. 42, No. 2, 1998, pp.193-205.

[113] G.H. Su, S.S. Tseng, C.J. Tsai, and J. R. Cheng, “Building an object-oriented and individualized learning environment on the WWW,” *Proc. of ICCE'99*, pp. 728-735, Chiba, Japan, Nov., 1999.

[114] J.M. Su, S.S. Tseng, C.H. Chen, Y.C. Sung, T.H. Su, and W.N. Tsai (2003), “A Study of Standardization of Traditional Teaching Materials,” *Proc. of ICEE2003*, International Conference on Engineering Education, Valencia, Spain, July 2003.

[115] J.M. Su, S.S. Tseng, C.T. Chen, and W.N. Tsai, “Adaptive Learning Environment for Pedagogical Needs,” *Journal of Information Science and Engineering (JISE)*, Vol. 20, No. 6, November, 2004, pp.1057-1077. **(SCI)**

[116] 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 (JISE)*, Vol. 21, No. 5, September, 2005, pp. 1053-1075. (SCI)

[117] J.M. Su, S.S. Tseng, C.Y. Chen, J.F. Weng, and W.N. Tsai, “Constructing SCORM Compliant Course Based on High Level Petri Nets,” *International Journal Computer Standards & Interfaces*, Vol. 28, Issue 3, 2006, pp. 336-355. **(SCI)**

[118] J.M. Su, S.S. Tseng, W. Wang, J.F. Weng, J.T.D. Yang, and W.N. Tsai, “Learning Portfolio Analysis and Mining for SCORM Compliant Environment,” *Journal of Educational Technology & Society (ETS)*, Vol. 9, Issue 1, 2006, pp. 262-275. **(SSCI)**

[119] E. Triantafillou, A. Pomportsis, and S. Demetriadis, “The Design and The Formative Evaluation of An Adaptive Educational System Based On Cognitive Styles,” *Computers & Education*, Vol. 41, No. 1, Aug. 2003, pp. 87-103.

[120] S.R. Terrell, “The Effect of Learning Style on Doctoral Course Completion in a Web-based Learning Environment,” *The Internet and Higher Education*, Vol. 5, Issue 4, 2002, pp. 345-352.

[121] Tool Interoperability Framework (TIF) (2005), Tools Interoperability Guidelines, http://www.imsglobal.org/ti/index.html

[122] E. Triantafllou, A. Poportsis, and S. Demetriadis, “The Design and The Formative Evaluation of An Adaptive Educational System Based on Cognitive Styles,” *Computers & Education*, Vol. 41, No. 1, 2003, pp. 87-103.

[123] C.J. Tsai, and S.S. Tseng, “Building A CAL expert system based upon two-phase knowledge acquisition,“ *Expert Systems with Applications* 22, 2002, pp. 235-248.

[124] C.F. Tsai. *Design and Implementation of New Object-Oriented Rule Base*

Management System. Master Thesis, Department of Computer and Information Science, National Chiao Tung University, 2002.

[125] C.J. Tsai, S.S. Tseng, and Y.C. Wu, "A New Architecture of Object-Oriented Rule Base Management System," *Proceeding of Tools Asia'99*, Nanjing, China, 1999.

[126] C.J. Tsai, S.S. Tseng, and C.Y. Lin, "A Two-Phase Fuzzy Mining and Learning Algorithm for Adaptive Learning Environment," *In Proceedings of the International Conference on Computational Science (ICCS'01), Lecture Notes in Computer Science (LNCS 2074)*, Vol. 2, CA, USA, 2001, pp. 429-438.

[127] ULF (Universal Learning Format), http://www.saba.com

[128] T.A. Urdan and C.C. Weggen (2000), "Corporate E-Learning: Exploring a New Frontier, " WR Hambrecht + Co, Retrieved May 2005 from www.e-learning.nl/publicaties/ marktonderzoek/New_Frontier.pdf

[129] J. Vassileva and R. Deters, "Dynamic Courseware Generation on The WWW," *British Journal of Educational Technology*, Vol. 29, Issue. 1, 1998, pp. 5-14.

[130] V. Vasandani and T. Govindaraj, "Intelligent diagnostic problem solving tutor: An experimental evaluation," in *Proc. Conf. Syst. Man, Cybern. Dec. Aiding Complex Syst.*, vol. 3, 1991, pp. 1739–1744.

[131] V. Vasandani and T. Govindaraj, "Knowledge organization in intelligent tutoring systems for diagnostic problem, solving in complex dynamic domains," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 7, 1995, pp.1077-1095.

[132] W3C (updated 9 Jun. 2004), World Wide Web Consortium. http://www.w3.org

[133] E.Y.C. Wong, A.T.S. Chan, and H.V. Leong, "Efficient Management of XML Contents over Wireless Environment by Xstream," *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 1122-1127.

[134] W.C. Wong and A. Fu, "Incremental Document Clustering for Web Page Classification," *Proceedings of IEEE Int. Conf. on Info. Society in the 21st century: emerging technologies and new challenges (IS2000)*, 2000.

[135] E.V. Wilson, "Student Characteristics and Computer-Mediated Communication," *Computers & Education*, Vol.34, 2000, pp. 67-76.

[136] C.Y. Wang, Y.C. Lei, P.C. Cheng, and S.S, Tseng, "A Level-wise Clustering Algorithm on Structured Documents," *Proceedings of NCS2003, Taiwan, 2003.*

[137] H.A. Witkin, P.T. Oltman, E. Raskin, and S.A. Karp, *A manual for the group embedded figures test*, Palo Alto: Consulting Psychologists Press, 1971.

[138] L.H. Wong, C. Quek, and C.K. Looi, "TAP: A software architecture for an inquiry dialogue-based tutoring system," *IEEE Trans. Syst., Man,Cybern. A*, vol. 28, May 1998, pp. 315–325.

[139] Y.C. Wu, *An Approach to Object-oriented Rule Base Management System.* Master Thesis, Department of Computer and Information Science, National Chiao Tung University, 1999.

[140] F.H. Wang and H.M. Shao, "Effective Personalized Recommendation Based on Time-Framed Navigation Clustering and Association Mining," Expert Systems with Applications, Vol. 27, 2004, pp. 365–377.

[141] eXtensible Markup Language (XML) (updated 26 Mar. 2004), Extensible Markup Language (XML). http://www.w3c.org/xml/

[142] S.M.T. Yau, H.V. Leong, D. MeLeod, and A. Si, "On Multi-Resolution Document Transmission in A Mobile Web," *the ACM SIGMOD record*, Vol. 28, Issue 3, Sep. 1999, pp:37-42.

[143] A. Yoshikawa, M. Shintani, and Y. Ohba, "Intelligent tutoring system for electric circuit exercising," *IEEE Trans. Educ.*, vol. 35, pp. 222–225, Aug. 2000.

[144] J.T. D. Yang, C.Y. Tsai, and T.H. Wu, "Visualized Online Simple Sequencing Authoring Tool for SCORM-compliant Content Package," *Proceedings of the 4th IEEE International Conference on Advanced Learning technologies (ICALT 2004)*, Finland, August, 2004.

[145] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of* ACM-SIGMOD Int. Conf. Management of Data, 1996, pp. 103-114.