แบบจำลองการแนะนำวัตถุการเรียนรู้สำหรับผู้เรียนโดยใช้รูปแบบการเรียนรู้

นางสาวนพมาศ ปักเข็ม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

LEARNING OBJECT RECOMMENDATION MODEL FOR LEARNERS BASED ON LEARNING STYLES

Miss Noppamas Pukkhem

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2009

Thesis Title	LEARNING OBJECT RECOMMENDATION MODEL FOR LEARNERS BASED ON LEARNING STYLES
By	Miss Noppamas Pukkhem
Field of Study	Computer Engineering
Thesis Advisor	Associate Professor Wiwat Vatanawood, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

.... Dean of the Faculty of Engineering

(Associate Professor Boonsom Lerdhirunwong, Dr. Ing.)

THESIS COMMITTEE

.... Chairman

(Associate Professor Pornsiri Muenchaisri, Ph.D.)

.... Thesis Advisor

(Associate Professor Wiwat Vatanawood, Ph.D.)

.... External Examiner

(Assistant Professor Pattarachai Lalitrojwong, Ph.D.)

.... Examiner

(Associate Professor Taratip Suwannasart, Ph.D.)

.... Examiner

(Assistant Professor Arthit Thongtak, Ph.D.)

นพมาศ ปักเข็ม : แบบจำลองการแนะนำวัตถุการเรียนรู้สำหรับผู้เรียนโดยใช้รูปแบบการเรียนรู้ (LEARNING OBJECT RECOMMENDATION MODEL FOR LEARNERS BASED ON LEARNING STYLES) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 194 หน้า.

โดยทั่วไปในระบบจัดการการเรียนรู้มักจะไม่มีส่วนในการทำความรู้จักรูปแบบการเรียนรู้ของผู้เรียน ทำให้มีการนำเสนอวัตถุการเรียนรู้เดียวกันกับผู้เรียนทุกคน ผลลัพธ์ที่ได้คือวัตถุการเรียนรู้ดังกล่าวอาจไม่สอดคล้องกับลักษณะของผู้เรียนและเป็นการเพิ่มความไม่บรรลุเป้าหมายการเลือกวัตถุการเรียนรู้ของผู้เรียนอีกด้วย

วัตถุประสงค์หลักของงานวิจัยนี้ คือ การนำเอารูปแบบการเรียนรู้มาประยุกต์เข้ากับระบบการแนะนำวัตถุการเรียนรู้ โดยจัดสร้างขึ้นมาในรูปแบบของกฎสำหรับการเชื่อมโยงความสอดคล้องระหว่างรูปแบบการเรียนรู้กับคุณลักษณะของวัตถุการเรียนรู้ด้วยเทคนิคการวิเคราะห์คำ เพื่อแนะนำวัตถุการเรียนรู้ที่สอดคล้องกับรูปแบบการเรียนรู้ให้กับผู้เรียน แบบจำลองการแนะนำวัตถุการเรียนนี้มีการเสนอวิธีการคัดกรองแนวคิดของการเรียนรู้ที่ไม่เหมาะสมด้วยอัลกอริทึมการสร้างแผนที่แนวคิดแบบร่วมกันของผู้สอน ในส่วนการแนะนำผู้เรียนได้มีการสร้างแบบจำลองผู้เรียนโดยใช้แบบจำลองการจำแนกลักษณะการเรียนรู้ของเฟลเดอร์และซิลเวอร์แมนในการจำแนกผู้เรียนออกเป็น 8 รูปแบบ และพัฒนาวิธีการคำนวณค่าความสอดคล้องของแต่ละวัตถุการเรียนรู้สำหรับแต่ละผู้เรียน โดยใช้อัลกอริทึมสามแบบ ได้แก่ 1) แบบไม่คำนึงถึงลักษณะส่วนบุคคล 2) แบบวิเคราะห์ความต้องการของผู้เรียนตามลักษณะการเรียนรู้กับเนื้อหาของวัตถุการเรียนรู้ 3) แบบวิเคราะห์ความเหมือนระหว่างผู้เรียนที่เจาะจงกับผู้เรียนอื่น ๆ จากผลการทดลอง พบความน่าสนใจของรูปแบบส่วนใหญ่ของผู้เรียนซึ่งมีประโยชน์ต่อการนำมาปรับปรุงการเรียนการสอนและการสร้างวัตถุการเรียนรู้ โดยพบว่าค่าความคลาดเคลื่อนระหว่างการเลือกวัตถุการเรียนรู้ที่ผู้เรียนชอบด้วยตนเองและการเลือกวัตถุการเรียนรู้ตามความสอดคล้องกับรูปแบบการเรียนรู้โดยการแนะนำจากระบบนั้น การให้คำแนะนำโดยการใช้อัลกอริทึมในแบบที่ 2 ให้ค่าความคลาดเคลื่อนน้อยกว่ากลุ่มอื่น.

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต.......................................

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.......................................

ปีการศึกษา 2552

##4771815621 : MAJOR COMPUTER ENGINEERING

KEY WORDS : LEARNING OBJECT/ LEANER MODELING/ LEARNING STYLE / RECOMMENDATION SYSTEM

NOPPAMAS PUKKEHM: LEARNING OBJECT RECOMMENDATION MODEL FOR LEARNERS BASED ON LEARNING STYLES. THESIS ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, Ph.D., 194 pp.

In general, Learning Management Systems (LMS) do not usually "know" the learner and simply present the same learning object to all learners without taking into learner learning styles or their preference. This gave result increased dissatisfied learning object to learners.

The main focus of this research is to apply the learning style in learning object recommendation system by using the mapping rules that are developed by word analysis technique. Based on learning style-based design, concept map combination model is proposed to filter unsuitable learning concepts for the course. In part of learning object recommendation, learner model based on Felder and Silverman learning style model is developed to classify learners into 8 styles and implement the compatible value computational methods, which includes three recommendations: i) non-personalized recommendation , ii) preferred feature-based recommendation, and iii) neighbor-based collaborative filtering recommendation. The results show the interesting patterns of the most learners are. It is useful to improve the learning process in the educational system and learning object development. The analysis of preference error (PE) is considered by comparison between actual preferred learning object and compatible prediction, the least error in experimental domain is the feature-based recommendation algorithm.

Department: Computer Engineering Student's signature:..

Field of study: Computer Engineering Advisor's signature:..

Academic year: 2009

ACKNOWLEDGEMENTS

First of all I would like to express my gratitude to my Ph.D. advisor, Associate Professor Dr.Wiwat Vatanawood, for his valuable guidance and continuous support throughout my thesis and for providing me with an excellent research environment.

I would also like to thank Professor Dr. Martha Evens, who was my advisor when I carried out my research overseas at the Department of Computer Science, College of Science and Letters, Illinois Institute of Technology, USA. She suggested me some interesting ideas which are useful for researching on the Artificial Intelligence area.

I would like to thank Associate Professor Dr.Pornsiri Muenchaisri, Assistant Professor Dr.Pattarachai Lalitrojwong, Associate Professor Dr.Taratip Suwannasart and Assistant Professor Dr. Arthit Thongtak for making me the honor of accepting to participate in my thesis committee, as well as, for their valuable feedback on my work.

I would also like to thank my students, who contributed to the implementation of parts of the proposed recommendation algorithm, as well as to all the participants in the experimental studies.

Special thank go to all my colleagues form the Software Engineering Laboratory and Computer Graphic and Computer Image Laboratory, especially grateful to my friends: Siwaporn Sorncharean and Santi Chotekeaw, for their support throughout the part my study time, as well as for their valuable comments on my research.

Moreover I am grateful to the Thai Government and Thaksin University for funding this research.

Last but not least, I would like to emphasize the loving care of my husband. My greatest thanks go to my parents for their endless support throughout my entire life, for being my mentors and my role models.

CONTENTS

Page

CHAPTER

Page

CHAPTER

Page

CHAPTER

Page

CHAPTER

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

The first section of this chapter describes the motivation, background and problem statement of the thesis. Next the objective issues, scope, contributions and research methodology are outlined. The chapter ends with the organization of the thesis.

1.1 Motivation

Today's e-learning is dominated by the Learning Management Systems (LMS), such as Blackboard (Backboard, 2008), Moodle (Moodle, 2008), ATutor (ATutor, 2008) or dotLRN (dotLRN, 2008); these LMS represent integrated systems which offer support for a wide area of activities in the e-learning process. These systems provide instructors to create the courses and test suites, to communicate with the learners, to monitor and evaluate their works. The learners can learn, communicate and collaborate by means of LMS.

Online digital learning resources are commonly referred to as learning objects in e-learning community. They offer a new way of thinking about learning content. Actually, learning objects can be educational components presented in any format. Learning objects are commonly stored in learning object repositories which facilitate various functions, such as learning object creation, submission, search, comment, review, etc. Rapidly evolving internet and web technologies have unlocked using learning objects in LMS, but the problem is that LMS does not offer personalized services and it dues to the "one-size-fit-all" problem. All learner being given access to the same set of learning objects and tools without taking into account the difference in interest, prior knowledge, experience, motivation and goals. This gives result in lack of learner information to perform accurate prediction of the most compatible learning objects. Researchers have tried to find out how learners learn? It is a part of this thesis to provide a pattern of learner with their learning style that can be used in the recommendation model.

Focus of this research is on building the learning object recommendation model. This model consists of the methods to provide a suitable concept map according to various experts' designs, and the recommendation methods on the basis of learner styles. Learner's learning style is used as the adaptation criterion that different learners have distinctive characteristics and learning styles, since it is one of the individual differences that play an important role in learning, according to educational field.

1.2 Objectives

The objectives of this thesis are as follows:

1. Develop methodologies for creating a concept map that provide suitable topics contained related learning objects.
2. Identify the significant metadata of learning object from existing metadata standards which give description of attributes of the learning object. The attribute will be used as input value of recommendation method.
3. Develop the learner model based on learning style dimensions.
4. Develop the recommendation algorithm that recommends the most compatible learning objects for learners based on learner model.

1.3 Scopes

In this work, the development of learning object recommendation methodologies that can be used to support individualized learning process for learner is proposed. The model architecture is designed based on multi-agent modeling and it provides the methodologies as follows:

1. An algorithm for building integrated concept map that combines the designs of various instructors. To combine the concept map from different instructors, we have assumptions:
    - The candidate concepts must inherit from the same learning goal hierarchy.
    - The concepts must be contained in the same course and in the same curriculum.
2. Organize and index the learning objects for proposed approach based on IEEE Learning Object Metadata (IEEE LOM).
3. Develop the learner model for providing the learner's value space of suitability of learning object calculation.
4. Develop a recommendation algorithm for calculating the compatibility of learning objects for learner.
5. Evaluate the algorithm by using experiments with groups of undergraduate learners in the university.

1.4 Contributions

This thesis provides a methodology for learning object recommendation that consists of two main works:

1. The methodology for combining the concept map from the various designs of experts that help the system to filter the unsuitable concepts for the course.
2. The generating of learner model based on learning styles.
3. The recommendation algorithms for selecting the personalized learning object to the learner that develop based on learner's learning style.

All of main works will support personalized learning object selection in learning management systems.

1.5 Research Methodology

1. Study instructional design theory, adaptive system structure, learning object concept, and learner learning style model.
2. Review existing researches on recommendation system in several fields.
3. Study fundamental theories of recommendation techniques, feature selection techniques, data mining techniques and evaluation methodology.
4. Design and develop the topic filtering method based on collaborative expert's designs.
5. Design and implement learning object model and learner model for collecting and preparing the initial learner and learning object datasets.
6. Set up experiments and test for learner style classifiers, each single recommendation algorithm (feature-based and collaborative filtering techniques).
7. Analyze the result of each algorithm.
8. Adjust the parameter of recommendation algorithm and retest with the same dataset.
9. Analyze the result and make conclusions.
10. Implement the web-based system prototype to demonstrate recommendation methodology.

1.6 Organization of the Thesis

This thesis is organized in six chapters as follows:

In chapter I, the motivation, objective, scopes and benefit of the work are presented.

Chapter II gives background and literature review. Several aspects are covered, including an overview of learning object concept, learning style theory, an adaptive hypermedia system, a basic of recommendation system and evaluation methodologies. The related works are also included.

Chapter III presents the analysis and design of learning object and learner model.

Chapter IV, the designing of learning object recommendation model is proposed. The course concept map combination model is presented in this chapter. Next, the detail of all proposed learning object recommendation algorithms are described.

Chapter V presents the experiments and results of each proposed recommendation algorithm and evaluation result comparisons.

Finally, chapter VI concludes the thesis, giving a summary of its main contribution, discussing its limitations and pointing towards future research directions.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

The recommendation systems for e-learning represent a continuously growing research domain, involving knowledge from several fields (context-collaborative filtering, adaptive system, learner modeling, learning management system, instructional science). This chapter deals mainly with the technical aspects of background knowledge for learning object recommendation systems. The related works are also included.

The first section presents an overview of learning object concept. Next in section 2.2, the details an adaptive system including its components are explained. The detailed description of each technique in recommendation systems is presented in Section 2.3. Section 2.4 describes evaluation methodologies for recommendation systems. Then, the chapter ends with some related works provided in Section 2.5.

2.1 Learning Objects and Learning Object Metadata

2.1.1 Learning Objects

Learning object is the term that is widely used to refer to educational materials. Some definitions for learning objects are summarized as follows:

- "Modular digital resources uniquely identified and meta-tagged, that can be used to support learning." – National Learning Infrastructure Initiative (Educause, 2007).
- "The main idea of learning objects is to break educational content down into small chunks that can be reused in various learning environments, in the spirit of object-oriented programming" –David A. Wiley (Wiley, 2002).
- "Any entity, digital or non-digital, that may be used for learning, education or training." –IEEE 1484.12.1-2002. July, 15 2002, Draft Standard for Learning

Object Metadata, IEEE Learning Technology Standards Committee (LTSC) (IEEE LOM, 2008).

According to this board and vague definition, almost everything could be considered as a learning object. A traditional text book, a web page, a piece of multimedia content, a software tool and even a person, an event, or a place can all be considered learning objects. The IEEE definition has been highly criticized. It fails to become an authentic and universally accepted definition. Consequently, various definitions, which narrow down the scope, have been created by different groups of practitioners. Wiley proposes a working general definition of a learning object – "any digital resource that can be reused to support learning" (Wiley, 2002). The learning object architecture separates content, display and navigation; but then seeks to bind the instructional materials into a coherent learning experience based on instructional strategy.

Learning objects are a new way of thinking about learning content design, development and delivery. Instead of providing all of the material for an entire course or lecture, a learning object only seeks to provide material for a single lesson or lesson-topic within a larger course. Examples of learning objects include simulations, interactive data sets, quizzes, surveys, annotated texts and adaptive learning modules. In general, learning objects have the following characteristics

- Self-contained – each learning object can be consumed independently
- Reusable – a single learning object may potentially be used in multiple contexts for multiple purposes on multiple campuses
- Can be aggregated – learning objects can be grouped into larger collections, allowing for their inclusion within a traditional course structure

- Tagged with metadata – every learning object has descriptive information allowing it to be easily found by a search -- which facilitates the object being used by others
- Just enough – if you need only part of a course, you can use only the Learning Objects you need
- Just in time – learning objects are searchable, you can instantly find and take the content you need
- Just for you – learning objects allow for easy customization of courses for a whole organization or even for each individual

A learning object does not have a predetermined size. Granularity of a learning object can extend from sub-topics to topics to lessons, and their associated media elements. Collections of learning object topics aggregate to form lessons, modules, courses, and curriculum libraries. Figure 2.1 shows the granularity of a learning object.

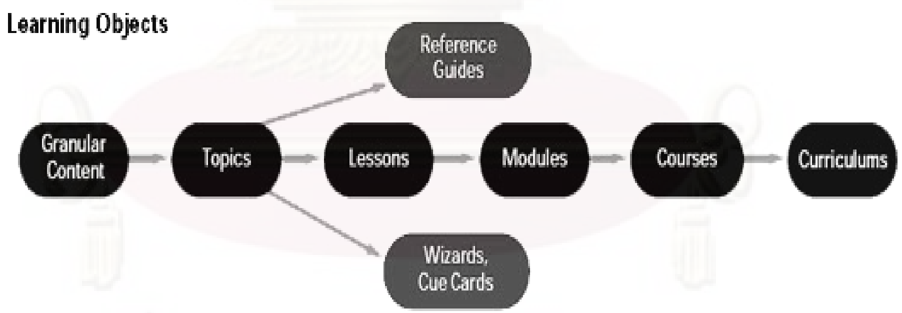


Figure 2.1: The granularity of a learning object (Alderman, 2002).

Figure 2.2 represents a common way of planning content organization. Topic level is a composition of digital media elements: text, graphics, animation, audio, video, and interactive user interface components.

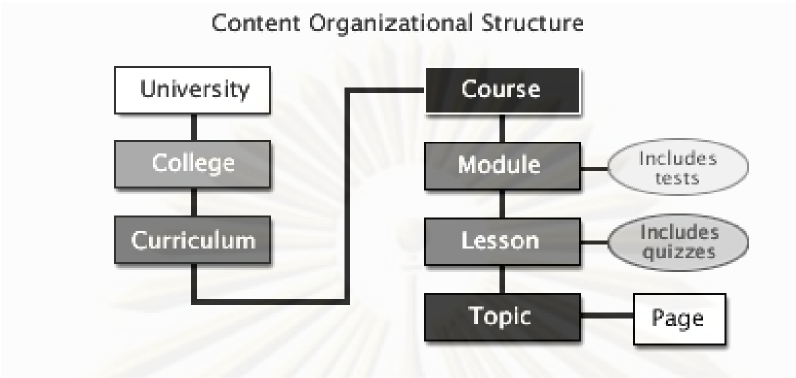


Figure 2.2: The content organization structure of Learning Object (Alderman, 2002).

The goal of this thesis is to develop an approach for learning object recommendation. A learning object has to be evaluated to decide its suitability. IEEE learning object structure appears rational and essential for this purpose. In the scope of this research, we assume that all learning objects consist of some elements as specified in IEEE definition.

2.1.2 Learning Object Metadata

International efforts have been made on developing standards and specifications about learning objects since late 1990's. IEEE Learning Technology Standards Committee, IMS Global Learning Consortium, Inc., and CanCore Initiative are organizations active in this area.

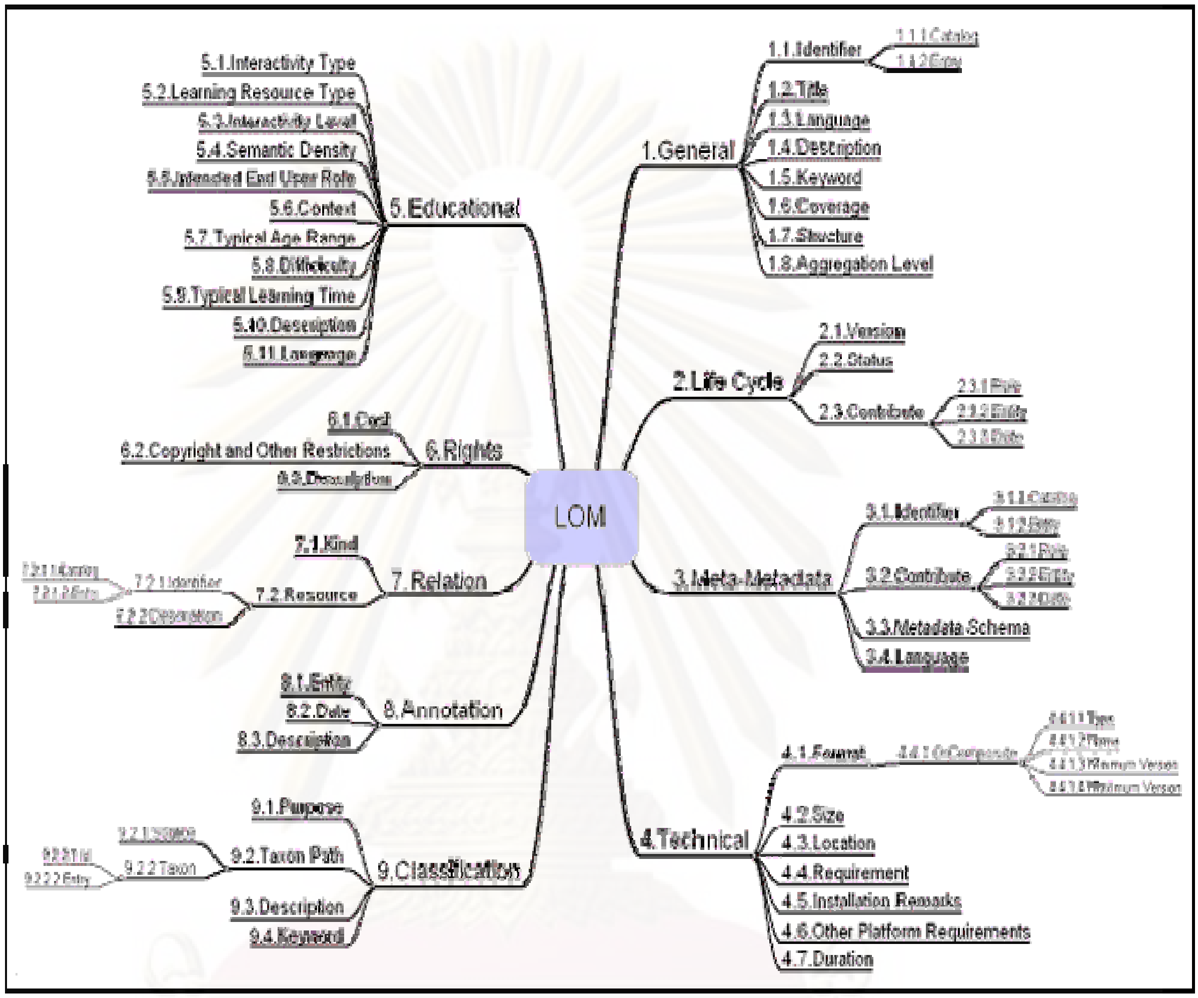


Figure 2.3: The structure of IEEE LOM metadata elements (IEEE LOM, 2008).

IEEE LOM Standard is a multipart standard, which is composed of Standard for Learning Object Metadata Data Model, Standard for XML Binding and Standard for RDF Binding. The first part of the standard, IEEE 1484.12.1 LOM Data Model standard (IEEE, 2008), has been accredited and released. The LOM Data Model is the core of existing metadata specifications. It defines a hierarchical structure for describing a learning object. In a LOM instance, relevant characteristics of learning object are represented by data elements that are grouped into nine categories. Figure 2.3 depicts the overall structure of LOM Data Model and the description of each top category is described in Table 2.1

Table 2.1: Top level of LOM categories.

Top Level	Description
General	The General category groups the general information that describes the resource as a whole.
Lifecycle	The Lifecycle category groups the features related to the history and current state of this resource and those who have affected this resource during its evolution.
Meta-metadata	The Meta-metadata category groups information about the meta-data record itself (rather than the resource that the record describes).
Technical	The Technical category groups the technical requirements and characteristics of the resource.
Educational	The Educational category groups the educational and pedagogic characteristics of resource.
Rights	The Rights category groups the intellectual property rights and conditions of use for the resource.
Relation	The Relation category groups features that define the relationship between this resource and other targeted resources.
Annotation	The Annotation category provides comments on the educational use of the resource and information on when and by whom the comments were created.
Classification	The Classification category describes where this resource falls within a particular classification system.

The metadata specification developed by IMS and ARIADNE was the origin of IEEE LOM Standard. Since then, IMS has released various versions of IMS specification based on updates of IEEE LOM Standard development. Besides IMS Learning Resource Meta-Data Information Model (IMS Metadata Specification) (IMS, 2001), current IMS specification includes documents defining other useful operations such as learning content packaging and simple sequencing.

The IEEE LOM standard and IMS specification are both complex and general. CanCore addresses this issue with its synthesis efforts that include guidelines for selecting elements, refinements of definitions, examples, technical implementation notes, and vocabulary recommendations (IEEE LOM, 2008). CanCore is an instantiation of the LOM standard that occupies the middle ground between this standard and the concrete work for building interoperable metadata records.

The Advanced Distributed Learning (ADL) Initiative is another organization working with IEEE and IMS closely. While CanCore focuses on semantics and interpretation, ADL puts efforts on technical issues. ADL's Sharable Content Object Reference Model (SCORM) bundles or integrates a collection of specifications and standards into a collection of "technical books", a set of interrelated technical standards, specifications and guidelines designed to meet high-level requirements for learning content and systems (ADL SCORM, 2008). It is often illustrated as a bookshelf holding nearly all of the specifications come from other organizations including IEEE, IMS, etc. The SCROM consists of three main topics, Content Aggregation (CAM), Run-time Environment (RTE), and Sequencing and Navigation (SN). The technology developments from those groups are integrated within a single reference model to specify consistent implementations, and additional detail and implementation guidance have been added.

Because of the promise of exchanging and sharing learning objects, however, this standardized metadata approach is well accepted around the world. To meet the requirements of learning object recommendation, extending existing standards and specifications to include more information such as contextual requirements and learner style and preference would be one direction to explore.

2.2 Learning Style

Learning style is an important criterion towards providing personalization, since they have a significant influence on the learning process. Attempting to represent the learning styles of learner and adapting the learning object so as the most suit them is a challenging research goal. The definitions are started in Section 2.2.1. Next, in Section 2.2.2 we present the examples of existing learning style model and Section 2.2.3 addresses the selected learning style in this research.

2.2.1. Learning Style Definitions

Learning style is one of the individual differences that play an important role in learning. Learning style designates everything that is characteristic to an individual when learner is learning, i.e. a specific manner of approaching a learning activity, the learning strategies activated in order to fulfill the task. There have been given several definitions:

- "the composite of characteristic cognitive, affective, and psychological factors that serve as relatively stable indicators of how a learner perceives, interacts with, and responds to the learning environment" (Keefe, 2003)
- "distinctive behaviors which serve as indicators of how a person learns from and adapts to his environment, and provide clues as to how a person's mind operates" (Gregorc, 1979)
- "a gestalt combining internal and external operations derived from the individual's neurobiology, personality and development, and reflected in learner behavior" (Keefe and Ferrell, 2002)

- "a predisposition on the part of some students to adopt a particular learning strategy regardless of the specific demands of the learning task" (Beshuizen and Stoutjesdijk, 1999)
- "an individual's preferred approach to organizing and presenting information" (Riding and Rayner, 1998)

As we can see, learning style has been attributed several connotations in the literature. Learning styles can be seen as applied cognitive styles, removed one more level from pure processing ability usually referring to learners' preferences on how they process information and not to actual ability, skill or processing tendency. According to (Riding and Rayner, 1998), the key elements in an individual's personal psychology which are structured and organized by an individual's cognitive style are affect or feeling, behavior or doing, and cognition or knowing, and this psychological process is reflected in the way that the person builds a generalized approach to learning. The building up of a repertoire of learning strategies that combine with cognitive style, contribute to an individual's learning style (Papanikolaou et al., 2006).

2.2.2. Example of Learning Style Models

Coffield identified 71 models of learning styles, among which 14 were categorized as major models, according to their theoretical importance, their widespread use and their influence on other learning style models (Coffield et al., 2004):

- Gregoric's Mind Styles Model and Style Delineator (Gregorc, 1985)
- Myers-Briggs Type Indicator (Myers and McCaulley, 1985)
- Felder-Silverman Learning Style Model (Felder and Silverman, 1988)
- Allinson and Hayes' Cognitive Style Index (Allinson and Hayes, 1996)
- Herrmann's Brain Dominance Instrument (HBDI) (Herrmann, 1996)

- Entwistle’s Approaches and Study Skills Inventory for Students (Entwistle, 1998)
- Riding’s Cognitive Styles Analysis (Riding and Rayner, 1998)
- Vermunt’s Inventory of Learning Styles (Vermunt, 1998)
- Kolb’s Learning Style Inventory (Kolb, 1999)
- Sternberg’s Thinking Styles Inventory (Sternberg, 1999)
- Honey and Mumford’s Learning Styles Questionnaire (Honey and Mumford, 2000)
- Apter’s Motivational Style Profile (Apter, 2001)
- Jackson’s Learning Styles Profiler (Jackson, 2002)
- Dunn and Dunn’s model and instruments of learning styles (Dunn and Griggs, 2003)

In Table 2.2, the learning styles theories and models are presented. For each model, the presentation includes: (i) the learner categorizations proposed by each model, (ii) the existence of an assessment instrument for categorizing each learner in the above categories, and (iii) indicative references for each model.

Table 2.2: The examples of learning style models and their assessment instrument.

Name	Learner’s Categorization	Assessment Instrument
Kolb Learning Style Inventory	Divergers(concrete, reflective), Assimilators(abstract, reflective), Convergers(abstract, active), Accommodators(concrete, active)	Learning Style Inventory (LSI), consisting of 12 items in which subjects are asked to rank 12 sentences describing how they best learn

Table 2.2: The examples of learning style models and their assessment instrument.(cont.)

Name	Learner's Categorization	Assessment Instrument
Dunn and Dunn Learning Style Assessment Instrument	Environmental, Emotional, Sociological, Physical factors	(i) Learning Style Inventory (LSI) designed for children grade 3-12; (ii) Productivity Environmental Preference survey (PEPS)-adult version of the LSI containing 100 items
Felder-Silverman Index of Learning Styles	Sensing-Intuitive, Visual-Verbal, Active-Reflective, Sequential-Global	ILS questionnaire, consisting of 44 questions
Riding-Cognitive Style Analysis	Wholists-Analytics, Verbalisers-Imagers	CSA (Cognitive Styles Analysis) test, consisting of three sub tests based on the comparison of the response time to different items
Honey and Mumford Learning Styles Questionnaire	Theorist, Activist, Reflector, Pragmatist	Honey&Mumford's Learning Styles Questionnaire (LSQ), consisting of 80 items with true/false answers

Table 2.2: The examples of learning style models and their assessment instrument.(cont.)

Name	Learner's Categorization	Assessment Instrument
Gregoric-Mind Styles and Gregoric Style Delineator	Abstract Sequential, Abstract Random, Concrete Sequential, Concrete Random	Gregoric Style Delineator containing 40 words arranged in 10 columns with 4 items each; the leaner is asked to rank the words in terms of personal preference
Hermann-Brain Dominance Model	Quadrant A (left brain, cerebral), Quadrant B (left brain, limbic), Quadrant C (right brain, limbic), Quadrant D (right brain, cerebral)	120 questions that refer to four profile preferences codes corresponding to each quadrant
Mayers-Briggs-Type Indicator	Extroversion, Introversion, Sensing, Intuition, Thinking, Feeling, Judgment, Perception	(i) MBTI(Myers-Briggs Type Indicator), (ii) Kiersey Temperament Sorter I, and (iii) Kiersey Character Sorter II

These models differ in the learning theories they are based on, the number and the description of the dimension they include. According to Curry's "Onion Model" (Curry, 1983), learning style models can be categorized into four layers:

1. Personality Models: this model focuses on the personality traits of the learner and the way they influence the learning process.

2. Information Processing Models: this model focuses on the processes of acquiring, ordering and engaging with information.
3. Social Interaction Models: this model focuses on the collaborative aspects of the learning process.
4. Instructional Preference Models: this model focuses on the environmental, emotional and sociological preferences of the learner.

According to (Coffield et al., 2004), learning style can be identified as five families as follows:

1. Genetic and constitutionally based factors
2. Cognitive structure family
3. Stable personality type
4. Flexible stable learning preferences
5. Learning approaches and strategies

2.2.3. Incorporating Learning Style in Proposed Approach

Felder-Silverman learning style model is the one of the most widely used learning style in adaptive hypermedia system. The suitable learning style models for finding the learning style of learners are concluded by Brown (Brown et al, 2007):

- The model should be able to quantify learning styles (computable condition)
- The model should display a good degree of validity and reliability/internal consistency and thus provide accurate evaluations of learning style
- The model should be suitable for use with multimedia
- The model should be suitable for use with adaptive web-based education system
- The model should be easily administered to university students

Another important reason noted by Sangineto (Sangineto et al, 2007), Felder-Silverman learning styles was widely experimented and validated on an engineering and science student population. Furthermore, this model contains useful pragmatic recommendations to customize teaching according to the students' profiles.

For this thesis, the reasons for selecting Felder and Silverman model are presented as follows:

- It is clearly in process of learning style assessment. The learner is classified into eight styles.
- This model provides the Index of Learning Style (ILS) questionnaire. The ILS questionnaire may be used at no cost for non-commercial purposes by individuals who wish to determine their own learning style profile and by educators who wish to use it for teaching, advising, or research. Moreover, the structure of sentence is easy to support the word analysis of mapping rules generating.
- A 44-item ILS questionnaire is suitable for learner intention to answer all questions.
- The same reason that mentioned by Sangineto that Felder-Silverman learning style is popular for an engineering and science learner supported by the experiment of validation in many educational researches.

A 44-item ILS questionnaire is designed to detect all psychological domains of learning style. The number of questions is verified to cover eight learning styles: active, reflective, sensing, intuitive, visual, verbal, sequential and global (Felder and Silverman, 1988). It is very important that learners have to answer every question to measure their learning style.

Next, the Felder-Silverman learning style model (Felder and Silverman, 1988) which will be used to reference in proposed approach is described in more detail. According to it, learners are characterized by their preferences in four dimensions and their observed criteria are presented in Table 2.3.

Table 2.3: The observed criteria of each learning style dimension.

Learning Style Categories	Learning Style Dimension	Observed Criteria
Perception	Sensing/Intuitive	- Time spent - Content's nature - Kinesthetic activity
Reception	Visual / Verbal	- Format (text, video, etc.)
Understanding	Active/ Reflective	- Kinesthetic activity - Material reviewing
Processing	Sequential / Global	- Navigation action

Active / Reflective learners: Active learners learn by trying things out and enjoy collaborative working, while reflective learners like to think about the material first and prefer working alone.

Sensing/ Intuitive learners: Sensing learners have a preference towards facts and details and they tend to be practical and careful, while intuitive learners prefer abstract material, they like to innovate, to discover possibilities and relationships.

Visual / Verbal learners: Visual learners remember best what they see (pictures, diagrams, schemas etc) while verbal learners get more out of works, either spoken or written.

Sequential / Global learners: Sequential learners tend to gain understanding in linear steps, while global learners learn in large leaps, being fuzzy about the details of the subject but being able to make rapid connections between subjects.

As all dimensions described above, we can categorize them into four groups of style: perception style, reception style, understanding style and processing style. Each style is presented as follows.

2.2.3.1 Perception Style

The learning experience starts with the learner's perception of the material. At this stage, the learner is either more sensing or intuitive. Sensing concentrate on information gathered through the five senses. They are interested in "just the facts" that they need and do not want to be bothered with any information or ideas that may confuse the issue. Alternatively, intuitive learners are much more interested in meaning and relationships than they are in the facts themselves. They are very good at reading between the lines and tend to anticipate future events. This dimension can be measured by the time spent, the level of activity involved, and the content's nature (theory or application).

2.2.3.2. Reception Style

Learners receive information through two primary channels: visual and auditory. Visual learners remember best what they see (pictures, diagrams, flow charts, time lines, films, and demonstrations). Verbal learners benefit more from words (written and spoken explanations). However, everyone learns more when information is presented both visually and verbally. This dimension can be measured by the format of the teaching material and the activity it involves from the learner.

2.2.3.3 Understanding Style

At the processing stage, active learners tend to retain and understand information best by doing something active with it (discussing, applying it or explaining it to others). They have a tendency to test and spend time experimenting with simulations, changing values of variables and observing the results. In addition, active learners tend to like group work. On the other hand, reflective observers spend more time on theoretical aspects of a subject to try to understand it thoroughly. They are motivated by reading the textbook, analyzing a diagram or a chart, and spending more time on the material than their active counterparts.

2.2.3.4 Processing Style

Another pair of learning style dimension is sequential and global. Sequential learners tend to gain understanding in linear steps, while global learners learn in large leaps, being fuzzy about the details of the subject but being able to make rapid connections between subjects.

2.3 Adaptive Educational Hypermedia System

Adaptive Educational Hypermedia System (AEHS), is a relatively new research direction, situated at the intersection of hypermedia and user modeling (Brusilovsky, 2001), offering an alternative to the traditional “one-size-fit-all” approach. Adaptive educational hypermedia system stores a user model that describes about goals, preferences, knowledge level of learner and use to interact with the learner in order to adapt to learner needs.

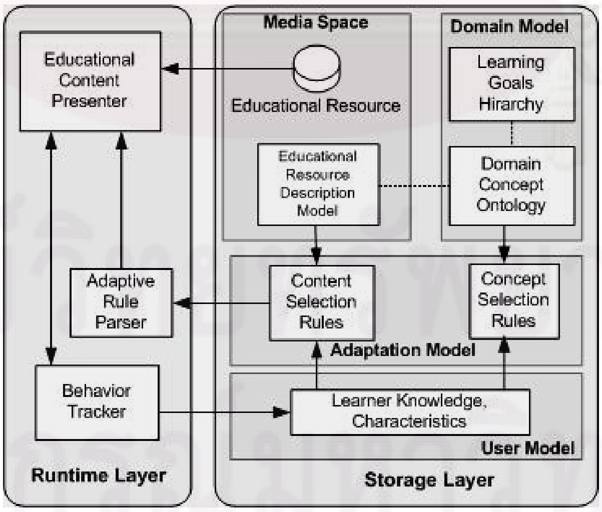


Figure 2.4: The generalized architecture of AEHS.

Figure 2.4 shows the generalized architecture of adaptive education hypermedia systems, which involves four key steps (Brusilovsky, 2005).

Designing the Domain Model; this process produces a design of the hierarchical learning goals, describing the course domain concepts.

Designing the User Model; this process produces the design of the model that defines the learner's cognitive characteristics and preferences.

Designing the Media Space; the process produces the design of the resource description model for representing the educational characteristics of the learning resources, e.g., the learning resource type, its complexity, the relationships among learning resources.

Designing the Adaptation Model; the process produces the design of the concept selection rules that are used for selecting appropriate resources from the Domain Model and the Media Space.

The common ground of system is reflected not only by their capability of adaptiveness, but also by their limitation:

- The adaptation can be achieved only among the local alternatives.
- Rules and conditions for learning resource selection and organization are predefined.
- The decision made in the system mainly relies on the built in virtual expert.

In this work, we focus on the concept selection rules in the adaptive hypermedia system that will be supported with learner model method and will be used in learning object mapping rule.

2.4 Recommendation Systems

Recommendation Systems (RSs) can be divided by three major types based on the technique used: Collaborative Filtering (CF), content-based filtering and hybrid filtering. Some researchers have added a fourth major type called "knowledge-based filtering" or “conversational” (Burke, 2002). In this thesis, we will discuss all of the three major types of recommendation systems. The key idea is personalization of the recommendation and at the core of personalization is the task of creating a model of the learner. Content-based approaches build user models that link the content of the information a user has consumed about the artifacts to be recommended to the preferences of the user concerning those artifacts; CF approaches build user models that link the information preferences of the user to those of other users with similar preference; hybrid approaches use a mixture of CF and content-based modeling; and knowledge-based approaches construct user profiles more gradually using many “interactive” forms of knowledge structure. In all approaches, the success to the item recommended is represented by the utility of the item, usually capture by a rating specified by the user based on how much the user liked the item (Adomavicius and Tuzhilin, 2005).

2.4.1 Recommendation Techniques

Content-based approaches recommend items based on the contents of the items a user has experienced before. Obviously, to ensure ‘high-quality’ recommendations, the system should conduct a rather delicate analysis on the content features of the target item in an attempt to establish the relationship between what the user likes and the target item.

Generally, the content-based recommendation approach has its roots in information retrieval (IR) and information filtering approaches. The IR researchers made the majority of current content-based techniques are able to associate the content aspect of items such as books, movies, documents, news articles etc, with the elements that are the most probably attractive to users (Woodruff et al., 2000). Content-based filtering in recommendation systems not only utilizes the content aspect of the items but also user profiles that contain information about users’ preferences. The user profile models are normally constructed explicitly from users’ own specified keywords from a list of pre-defined keywords on a topic;

or implicitly from the system's long-term observations of user behaviors (Woodruff et al., 2000).

Content-based recommendation systems overcome the limitations of the collaborative filtering by making suggestions based on the content of the items and target user's ratings. Two different content-based approaches have been proposed: feature-based and text categorization-based. Feature-based recommendation systems (Sebastiani, 2002) extract important features from the item descriptions and learn a user's profile using a set of pre-classified (according to the user's rating) feature vectors leading actor/actress in a movie recommender system. However, choosing representative features and appropriately encoding them, is not an easy task. Text categorization systems learn from thousands of features (words or phrases), but recent research has shown that it is possible to build effective classifiers (Sebastiani, 2002). Several systems using text categorization (TC) have been developed. They have been applied to recommend web pages, news documents and books.

Collaborative filtering (CF) makes recommendations by observing like-behavior groups. It starts with the assumption that users who enjoyed certain things in the past will enjoy similar things in the future. CF build user profiles by keeping user ratings on items without relying on the content of the items; a user-item rating matrix incorporating users and their rating maintains this information. CF remains the most commonly adopted technique in commercial recommendation systems (Herlocker et al., 2004), and the most studied in the academic community.

CF algorithms rely on similarity metrics computed between two users' ratings of items being recommended. The CF system has the potential to learn from a group of similar user and arrive at appropriate recommendations without the need to construct a complete profile for each user. Therefore, the key to CF is to apply similarity measurements to identify users with similar preferences to given user. A number of similarity measurements have been applied including Pearson correlation, mean squared difference, vector similarity

(Breese et al., 1998) and Euclidian distance which are used in this thesis. We can compute the distance between two scenarios using some similarity function sim(x,y) ranking from 0-1 by using equation (1), where x, y are scenarios composed of N features, such that $x = \{x_1, ..., x_n\}, y = \{y_1, ..., y_N\}$.

$$sim(x, y) = \frac{1 - \sum_{i=1}^{N} \sqrt{\left(x_i^2 - y_i^2\right)}}{\sum N} \qquad (1)$$

The disadvantage of collaborative recommendation systems is that they often require explicit user feedback. This produces problems as studies have shown that users are reluctant to provide any sort of conscious feedback without some form of incentive (Herlocker et al., 2004). This is particularly prevalent early on in system deployment as no recommendations can be given until users have first entered some ratings. This has become known as the cold start problem. Another disadvantage of these systems is that they can only suggest previously visited pages, and therefore designers have to engineer methods of pro-actively finding new resources and recommending them to their users. The third major problem with document recommendation occurs with certain individual users who have unique interests (Balabanovic and Shoham, 1997). Their trails fail to match any other group of users and this leads to poor recommendations (Mooney and Roy, 2000). This problem can be overcome by increasing the number of users or by using an alternative system for recommendation.

Because of the weaknesses of both content and collaborative recommendation techniques, some of the latest recommendation systems that have appeared in the 1990's are drawing on both techniques to provide recommendations. These new hybrid recommendation systems can use the strengths of both techniques to overcome their individual weaknesses (Claypool et al., 1999).

Hybrid recommendation mechanisms attempt to deal with some of these issues and smooth out the drawbacks of the collaborative filtering and content-based approaches. A purely content-based approach fails to consider community endorsement, and is concerned with only the significant features describing the content of an item, whereas, a purely collaborative filtering approach ignores the contents of item, and makes recommendations only based on comparing the user against clusters of other similar users. Consider, however, the possibility that item information can be obtained through a content-based approach, and user information can be obtained from collaborative filtering. By combining these two techniques, we can smooth out the drawbacks of both the pure content-based and pure CF approach and obtain both individual as well as collective experiences with respect to the items being recommended.

The majority of hybrid recommendation system combines collaborative and content-based approaches by learning and constructing a unified user profile for recommendations. For example, FAB (Balabanovic and Shoham 1997) can be regarded as two-layered filtering system. The first layer is created by a content-based approach, which ranks documents by topic, and then ranked documents are sent to user's personal filter. In the second layer, a user's relevance feedback is used to modify both the personal profile filter and the topic filter. It is obvious that only filtered documents are added to the list of candidate documents to be recommended it appropriately based on content filtering. (Claypool et al., 1999) and (Pazzani, 1999) attempt to build separate user profiles based on the content-based and collaborative mechanisms. Then, the outputs from these two approaches are incorporated either by a linear combination of ratings (Claypool et al., 1999) or a voting scheme (Pazzani, 1999).

2.4.2 Evaluation Methodologies

Since the first automated rating-based recommendation system was proposed in 1994, the accuracy of recommendation systems remained the ultimate evaluation goal in the research literature until early 2004 when several researchers began to explore other ways to evaluate the performance of recommendation systems (Herlocker,2004; McNee et al., 2006; Riedl and Dourish, 2005; Adomavicius and Tuzhilin, 2005).

Although the metrics adopted in previous recommendation systems differ, there are some commonly used metrics which have been acknowledged in the community. In this thesis, we will use the objective measures. Objective approaches are then sub-classified into two main categories: predictive accuracy metrics and classification accuracy metrics (Herlocker et al., 2004).

2.4.2.1 Predictive Accuracy Metrics

Predictive accuracy metrics examine how close the recommendation system's predicted ratings are to the true user ratings. Among the many flavors of these metrics, Mean Absolute Error or MAE is the most popular (e.g.(Melville et al., 2002, Shardanand and Maes, 1995, Sarwar et al., 1998, Claypool et al., 1999; Herlocker et al., 1999; Miller et al., 2004; Tang et al., 2005; Adomavicius and Tuzhilin, 2005). MAE measures the average absolution deviation between the user's true rating and the system's predicted rating. However, the accuracy of MAE depends heavily on how well and carefully 'true preference' is determined, that is, whether a rating of 3 or 4 should be regarded as 'good' by both the system and the user. This is especially true when the preference scale is small, say from 0 to 3. Errors will be inadvertently introduced into the system in erroneously classifying a 'good' item as a 'bad' one, or vice versa. For a larger scale, say 0-5 with 3.5 as the cut-off value differentiating good from bad items, then predicting a 4 as 5, or a 2 as 3, makes little difference to the users.

$$MAE = \frac{\sum_{i=1}^{N} |x - x_i|}{N} \qquad (2)$$

Obviously, the metric is of particular value for evaluating tasks where the predicted rating will be displayed o the user, in an attempt to establish a trust between the system and the user, so as to encourage the user to come to rely on the subsequent ratings given by the system (Herlocker et al., 2004). For instance, (Dahlen et al., 1998) make movie predictions and display them to the user (along with the number of the stars). Obviously, if the predicted ratings deviate from user's true ratings, it could compromise the credibility of the system.

2.4.2.2 Classification Accuracy Metrics

According to (Herlocker et al., 2004), classification accuracy metrics measure the ability of a recommendation system that makes correct or incorrect decisions to determine whether an item is good. Thus, this type of the measurement is usually regarded as a decision-support accuracy metric (Herlocker et al., 2004; Tang et al., 2005), which examines how well a recommendation system can make predictions that help users select high-quality items.

One assumption of these metrics is that user preferences in recommendation systems should be binary, that is, making recommendations is a binary classification process: either users will like it; or they will not. Suggested by Herlocker et al. (1999), and widely adopted in the research community (e.g. (Good et al., 1999; Meville et al., 2002; Tang et al., 2005) is ROC (Receiver Operating Characteristic) sensitivity, which was originally introduced into the IR community by Swets with the name 'relative operating characteristics'. Generally, the probability of a randomly selected good item being accepted by the user is referred to as sensitivity; while the probability of a randomly selected bad item being rejected by the user is referred to as specificity (Good et al., 1999). Thus, when adopted for a recommendation system, the ROC model measures the decision-support aspect of accuracy: how the system differentiates between 'good' items and 'bad' items. The metric can be represented in accuracy error (PE) and can compute by equation (3).

$$AE = 1 - \frac{\sum_{i=1}^{N}(X \cap Y)}{N} \tag{3}$$

In equation (3), X is the actual preference item, Y is the predicted item from the recommendation algorithm and N is the number of user. $\sum_{i=1}^{N}(X \cap Y)$ is the frequency of X and Y appearing together for user number i=1 to N.

2.5 Software Agents

The field of agents has many diverse researchers, approaches and ideas, which help to create one of the more dynamic research areas in recent years. This section introduces the field of agents, looking at the history behind their development and the characteristics that help define modern software agents. The huge popularities of agent research have arisen at the time when object-oriented programming language such as Java and C++ are proving such a success. This can be demonstrated by a quick visit to the popular search engine Google which will uncover over a hundred different agent frameworks, of which this section will describe only a select few.

2.5.1 Standardization

There are currently a wide range of different agent architectures, frameworks and systems developed for both research and industrial purpose. To unify these approaches three standardization efforts have appeared with the overall aim of increasing interoperability between agent systems.

- MASIF- The Mobile Agent System Interoperability Facility (MASIF, 2007) has been in development by the Object Management Group (OMG) since 1995 to promote interoperability and mobility among agent platforms.
- KQML- The knowledge Query Meta Language (KQML) (KQML, 2007) is one of the most popular and widely used protocols for defining agent-to-agent communication. KQML is the oldest project, developed in 1992 by the DARPA Knowledge Sharing Effort consortium.
- FIPA – The most recent addition is Foundation for Intelligent Physical Agents (FIPA, 2007), a non-profit organization created in 1996 aimed at developing software standards for maximizing interoperability within and across agent-based systems.

Of these three approaches, MASIF uses a procedure-oriented interaction model using Remote Procedure Calls (RPC) or Remote Method Invocation (RMI) technology, while both KQML and FIPA both specify a message-oriented Agent Communication Language (ACL). The ACL model used in both FIPA and KQML is based on speech act theory, a field of research aimed at analyzing the semantic content of vocalized messages.

These standards facilitate agent interaction across hardware platforms, operating systems, programming languages and agent platforms. Recent FIPA compatibility tests (FIPA, 2002) have already shown successful interoperability through the transfer of ACL message between several FIPA compliant frameworks.

2.5.2 Agent Frameworks

Agent Frameworks are programming tools for constructing agents. Examples of these are Voyager (Object Space), Aglets (IBM, 2007) and JADE (JADE, 2008) . Due to the numbers of agent frameworks available, an extensive analysis of them all would be out of the scope of this thesis. Instead, three systems will be examined in this section. On the most fundamental level, each framework supports three features for agent developers.

- Creation: Each framework provides the ability to quickly create and run agents within a supported environment.
- Communication: Each framework supports agent-to-agent communication using speech acts.
- Discovery: Each framework allows agents to find new agents using a service based discovery mechanism.

On top of this, each framework offers a unique set of additional features such as standards compliance, mobility, interoperability, knowledge-based ontologies and graphical interface.

2.6 Related Work

This section details a related work of this thesis. In Section 2.6.1, we present the selection of innovative content-based, collaborative and hybrid recommendation systems developed in the past. Next, the learning style personalization works are presented in Section 2.6.2.

2.6.1 Recommendation System

MEMOIR (Roure et al., 2001) is an agent-based system, designed to support researchers working with vast quantities of distributed information in finding both relevant documents and other researchers with related interests. Although not developed as such, it can be viewed as a collaborative recommendation system. MEMOIR finds related documents and people through a comparison of user trails, which the system regards as first class objects. There are two types of trails; user trails formed from the documents a user visits, and shared trails created by users who grouped related interesting document together. If the current document appears in one of these shared trails, then the system makes recommendations to other documents in the trail. A proactive “Similar User” agent informs the user of other users with similar interests by analyzing the overlap of the current user trail with those of other users in the system.

The Queries In Context (QuIC) system (El-Beltagy et al., 2001) provides a collaborative recommendation service using an agent-based distributed information management environment. The agent infrastructures mainly Java-based and uses KQML as the communications language. Agents work together within the system to support collaborative user queries and recommend links to users based on the current context of the document.

The central agent in QuIC is a directory service agent called the facilitator for registering services and routing messages. The facilitator supports the dynamic addition to this, there is an organizational memory agent that records the URL’s and bookmarks of the users. This agent is capable of responding to queries such as “Who has seen the following URL” and “Recommend URLs related to this document”. The organizational memory agent is written in Prolog. QuIC defines context as a feature vector of related terms; a collection of keywords that form a collective representation of the destination of each link. The link service agent receives a request for links containing a keyword or group of keywords and uses its own internal linkbase in addition to the services of other agents to compose a set of links that match the initial query.

Each user is assigned a personal user interface agent to interact with. This agent records information entered by the user such as their preferences, personal information, etc. Browsing history and bookmarked page information are presented to other agents upon request. The interface agent also provides the user with a query facility for interacting with other agents in the system. Responses are collated by the agent and presented to the user.

Personal WebWatcher (PWW) (Mladenic, 1996), is inspired by earlier work on WebWatcher (Joachims et al., 1997), which located information on the web and presented it to users when they provided a search engine-like request in the form of a set of keywords. PWW is a content-based recommender system that accompanies a user from page to page as they browse the web and presented it to users when they provided a search engine-like request in the form of a set of keywords. The designers wanted to remove the need for users to enter explicit information about themselves as was required by WebWatcher, so instead during periods of reduced user activity, e.g. at night, PWW analyses the user's navigational trail and constructs the user model from this information. The content from these trails is processed to obtain a set of keywords. The TF-IDF algorithm is then applied to form a set of associated weights. Finally, these vectors of word-weight pairs are analyzed using a Naïve Bayesian classifier algorithm to form a model of the user interests. The Naïve Bayesian classifier (Langley et al., 1992) is a modified version of Bayes' theorem where a simple probabilistic equation is used to form a probability given a set of incomplete data items and which can then update its probability when new information arrives.

WebMate (Chen and Sycara, 1998), another content-based recommendation system is a stand-alone proxy that monitors a user's web activity and uses an applet controller to act as a user interface to the proxy. Explicit feedback occurs whenever a user is interested in the page. They select an 'I like it' option in the controller applet and then WebMate utilized the TF-IDF algorithm to produce a weight vector for that document. Documents are categorized by applying a similarity function and a nearest neighbor algorithm to group similar documents together.

FAB is a recommender system which combines the advantage elements of both collaborative-based and content-based recommendation techniques. This recommendation system is developed since 1994 as part of the Standford University digital library project (Balabanovic and Shoham, 1997).

The FAB system is based around a two-stage document collection and then user selection process, and this is reflected in underlying agent architecture. The agents in FAB are simple processes that keep a persistent record of their state and demonstrate many of the primary and secondary agent characteristics. In the collection stage, documents are farmed from the web via a set of collection agents, each of which maintains a profile for a particular topic, using the information retrieval TF-IDF algorithm, key words are harvested from the web pages forming a representation vector for that page. Each agent then employs a cosine function and periodically sends the pages that best match its topic to a central repository.

2.6.2 Learning Style Personalization

AES-CS (Triantafillou et al., 2003) uses both adaptive presentation technique and adaptive navigation support to individualize the information and the learning path to the field dependence (FD)/field independence (FI) characteristic of the student. Specifically, AES-CS uses conditional text and page variants to present the information in a different style: from specific to general in case of FI learners (who have an analytic preference) and from general to specific in case of FD learners(who have a global preference). AES-CS offers also two control options: program control for FD learner, by means of which the system guides the learner through the learning material, and learner control for FI learners, by means of which the learners can choose their own learning paths, through a menu.

INSPIRE (Papanikolaou et al., 2006) is the web-based system that uses adaptive presentation techniques to adapt the learning content to the four learning styles in Honey and Mumford model (2002). The learning styles consist of activist, pragmatist, reflector and theorist. All learners are presented with the same knowledge modules, but their order and appearance differs for each learning style.

Tangow (Carro et al, 2001) is based on a similar adaptation approach, but uses two of the Felder and Silverman learning style dimension: sensing/intuitive and sequential/global and only two types of modules: “example” and “exposition”. For example, in case of sensing learners, the students are first present with an example and only after that with exposition regarding that concept.

Heritage Alive Learning System (Cha et al, 2006) is the learning system that provides adaptively customized learning interface. It contains 3 pair of widget placeholders (text/image, audio/video, Q&A board/Bulletin Board). Each pair consists of a primary and secondary information area. The space allocated on the screen for each widget varies according to the student’s Felder and Silverman learning styles. For example, for a visual learner the image data widget is located in the primary information area, which is larger than the text data widget.

Bajraktarevic (Bajraktarevic, 2003) presents the course content in a specific layout, corresponding to the Felder and Silverman learning styles (only sequential/ global preference). Pages for global students contain diagrams, table of contents, overview of information, summary, while pages for sequential learner only include small pieces of information, and Forward and Back buttons.

Graf (Graf, 2007) uses adaptation features such as: order of examples, exercises, self assessment tests and content objects and number of presented examples and exercises to adapt the course to the four Felder and Silverman learning styles.

Having covered the relevant research fields, topics, issue and history behind the work documented in this thesis, the next chapter will describe the developing the learning object model and learner model for proposed recommendation framework.

CHAPTER III

THE DESIGN OF LEARNER MODEL

In this chapter, the data preprocessing for recommendation method is presented. There are consist of the two groups of input space; i) the learner profile in term of learner model and ii) the data about learning object which is represented in learning object model. Section 3.1 shows the overall architecture of the proposed recommendation model. Section 3.2 presents definition of learner model. Next, the detail of learner model creation is proposed in Section 3.3. Finally, the Chapter ends with summary in Section 3.4.

3.1 Overall Architecture

An overall architecture that is presented as an abstract model in Figure 3.1 is used to design and develop our mechanisms for solving thesis problems.

The abstract model presents all processes for learning object recommendation in learning of university environment. Three processes: learner model generating, concept map and course creation, and learning object recommendation are defined as follows:

- **Learner model generating**: The learner model generation that is presented in dot-line box provides learner model by using the semantic mapping between learning style and learning object features. The learner model generating starts at learning styles assessment to find the learning styles of learner. Then, learning style scores are analyzed to define the degree of preferences in preference degree weighting process. The results from the previous process are used to construct the learning style set (LSS). To create the learner preference set (LPS) that describes about the mapping between learning styles and learning object features, the mapping rules are created, and validated by the semantic analysis. 44 Index of learning style questionnaires (ILS) are analyzed into the semantic groups and compared with learning object value space. The selected features of learning object in this thesis are analyzed by learner rating when we provide the learning object feature questionnaire. The details of each process are described in this Chapter.

- **Concept map combination and course creation**: These processes provide integrated course concept map from various instructor's concept map designs. Firstly, the instructors consider the main course concept map (MCC) to create their own concept maps which called instructor intention map (IIM). Secondly, all IIMs are combined into the integrated concept map and collect in concept map database. Finally, the course is created when the instructor contains learning objects into the concept map. The details of these processes are explained in Chapter IV.

- **Learning object recommendation:** The learning object recommendation is provides the computation of the suitability score of learning objects and rank them. When learner requests to learner the course, he/she has to select the concept which wants to learn. Then, the learning object recommendation is used to compute the preference scores to recommend the most compatible learning object based on learning styles. Next, ranked learning objects will be presented to learner. The details of processes are shown in Chapter IV.

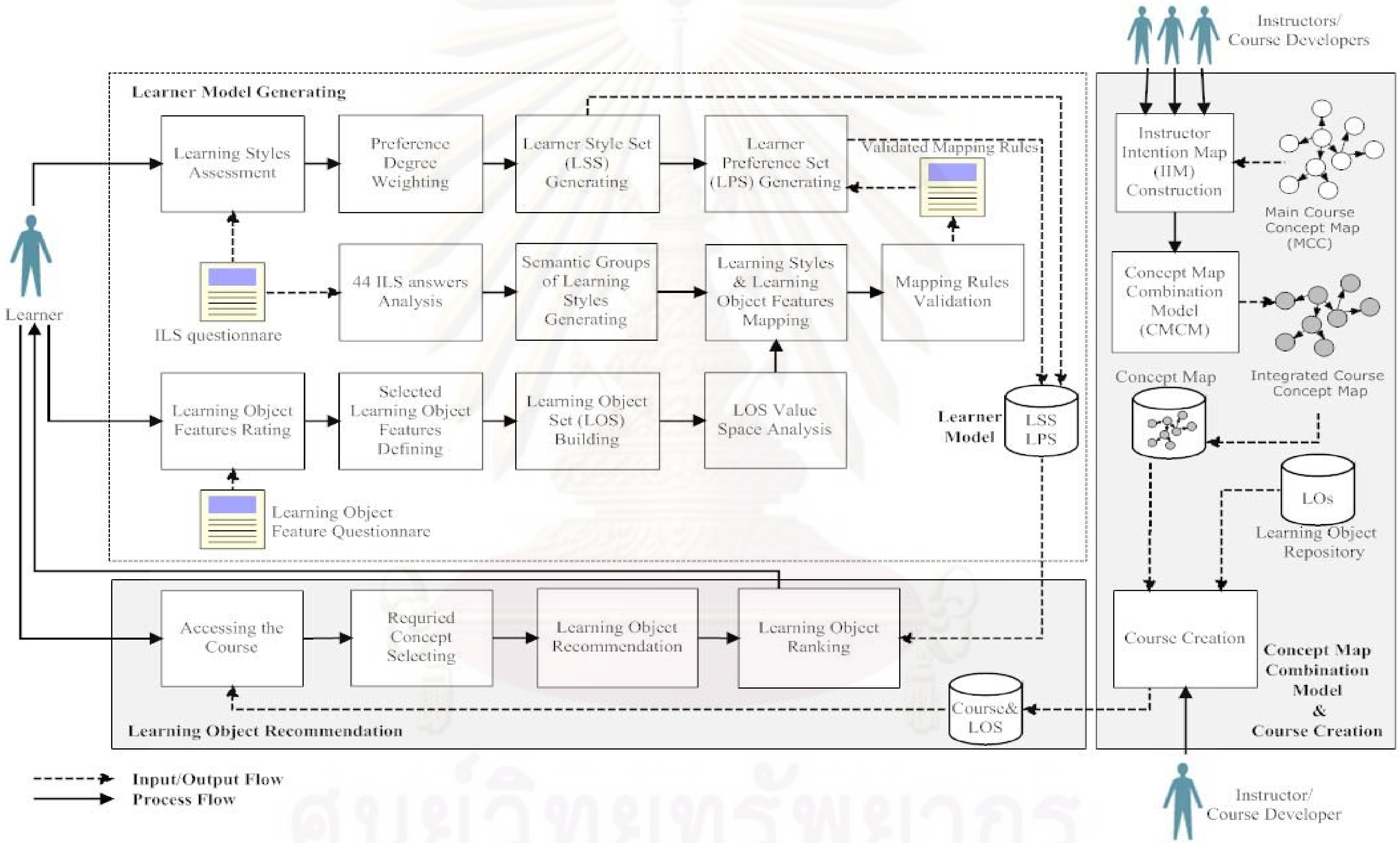


Figure 3.1: Abstract model for learning object recommendation based on learning styles

3.2 Learner Model

A learner model is the model constructed from observation of interaction between a learner and learning system of instructional environment. Brusilovsky (Brusilovsky, 2001) defined a definition of learner model as "The learner model is a model of the knowledge, difficulties and misconceptions of the individual. As a student learns the target material, the data in the learner model about their understanding is update to reflect their current beliefs".

Before constructing the learner model, we have to know the information about learners. In chapter II, the measuring instrument associated to the learning style models for diagnosing purpose and the example of existing system were described. The example systems were classified into two groups: those that use questionnaires for identifying the learning style and those that use learners' observable behavior. In this thesis the questionnaires approach is selected to use with the reasons as follows:

- It is simplicity: the instructor/researcher only has to apply a dedicated psychological questionnaire, proposed by the learning style model creators.
- The proposed is the part of LMS, so we do not implement all components of learning environment. So, it suits for our experimental setting.

However, the disadvantages of this questionnaire-based approach is it is static, so the leaner model is created at the beginning of the course and stored once and for all, without the possibility to be update. A method of improving this approach is to give the student the possibility to modify his/her own profile, if he/she considers that the one inferred from the questionnaire results is not appropriate. This is called an "Open model" approach and it is used either in conjunction with the questionnaires or in place of them. This direct access of learners to their own learner model has several advantages: it provides an increased learner control, it helps the learners develop their metacognitive skills and it also offers and evaluation of the quality of the model created by the system (Kay, 2001).

3.3. Learner Model in Proposed Approach

Thai learners are taught from elementary school through many graduate level programs in a traditional style: lecture and textbook generated learning. In forcing those to learn in a new environment without preparing those with the necessary skills for successful teaming, learners can become frustrated. In educational research, the study showed that students who possess these skills have a better opportunity to learn than those who do not.

The creation of learner model of this thesis is based on an assumption of the relation between learning style and learning object. Learning objects allow the learner to use the content learned in a particular part of a course by the following ways:

- demonstrate master of the content
- apply that knowledge to solving a problem
- use the content in a critical thinking exercise that both demonstrates mastery and allows the learner to place the content within the context of the larger topic of the course.

Based on the topics mentioned above, we note that the learner is the main factor for learning objects development. So the learning object can define and describe in terms of styles of learning and teaching allow instructors and course developers to develop a deeper understanding of the learning object for supporting their learners. If the learning objects are designed based on the learning styles, the learning object recommendation process can use the learners' learning styles to suggest the compatible learning object to the learners. This approach seems easy than leaner directly access to feature of learning object, because there are difficult for learner to understand the nature of learning object in terms of LOM metadata.

The processes to build learner model in proposed approach start at learner's learning style analysis by using Index of Learning Styles (ILS) questionnaire. The ILS is an instrument designed to assess preference on the four dimensions of the Felder-Silverman Learning style model. The answers of learner are evaluated by the learning style indicator that described in Subsection 3.3.1. Then, we classify learner in form of learner style set (LSS) by assigning the degree of learning style to each learning style. So, learning style set is the set of collection of each learning style and its weight. Finally, the learning object selection rules will be used to identify the preferred learning object features of each learner to create the learner preference set (LPS). Both of leaner style set and learner preference set will be stored in the learner model database. The overall processes are presented in Figure 3.2 and the detail of each process will be explained in subsection below.

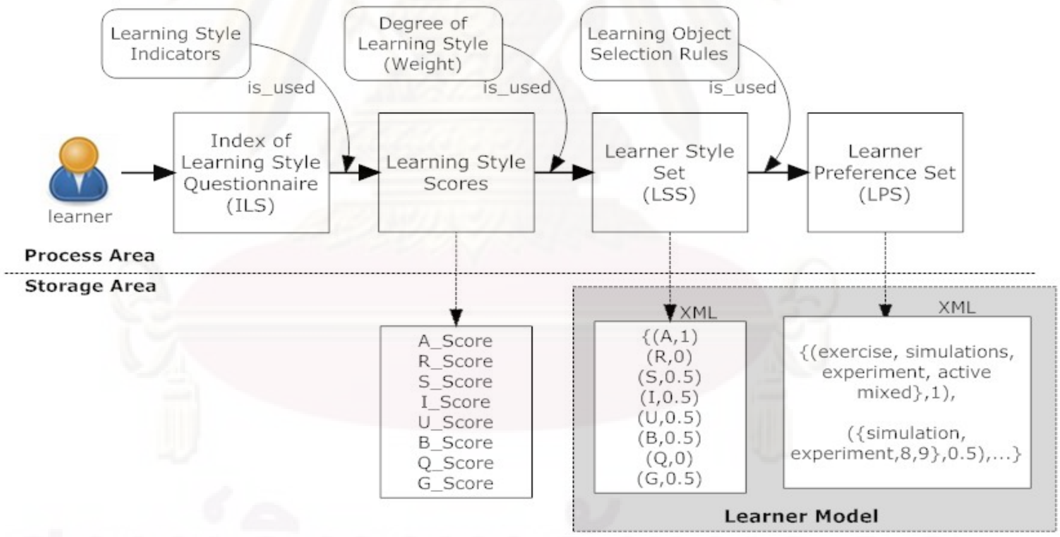


Figure 3.2: Overall processes of learner model creation.

3.3.1 The Index of Learning Styles(ILS)

Web-based version of ILS is taken hundreds of thousands of times per year and has been used in number of published studies, some of which include data reflecting on the reliability and validity of the instrument (Felder and Soloman, 2007). Table 3.1 shows the detail of Index of Learning Style (ILS).

Table 3.1: The indications of Index of Learning Style.

Dimension #	Questionnaire Indicator #	Preference and Symbol	Detail of Question
D1	1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41	**A**-**A**ctive/ **R**-**R**eflective	How does the student prefer to process information: actively—though engagement in physical activity or discussion, or reflectively—though introspection?
D2	2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42	**S**-**S**ensing/ **I**-**I**ntuitive	What type of information does the student preferentially perceive: sensory—sights, sounds, physical sensations, or intuitive—memories, ideas, insights?
D3	3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43	**U**-Vis**U**al/ **B**-Ver**B**al	Through which modality is sensory information most effectively perceived: visual—pictures, diagrams, graphs, demonstrations, or verbal—sounds, written and spoken words and formulas?
D4	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	**Q**-Se**Q**uential/ **G**-**G**lobal	How do the students progress toward understanding: sequentially—in a logical progression of small incremental steps, or globally—in large jumps, holistically?

The index of Learning Styles (ILS) is a 44-question instrument designed to assess preferences on the four dimensions of the Felder-Silverman model, see the questionnaire in Appendix A.1. Each learning style dimension has associated with it 11 forced-choice items, with each option (a or b) corresponding to one or the other category of the dimension (e.g., sensing or Intuitive). For statistical analyses, it is convenient to use a scoring method that counts ‘a’ responses, so that a score on a dimension would be an integer ranging from 0 to 11.

Felder points out that learner with a strong preference for a specific learning object may have difficulties in learning if the teaching style does not match with their learning styles (Felder and Silverman, 1988). In this thesis, preference scores is scaled into three groups:

Strong Preference: Learner strongly prefers to learn with this learning style. The interval of score is 8-11.

Medium Preference: Learner quiet prefers to learn with this learning style. The interval of score is 4-7.

Weak Preference: Learner is not prefer or do not like this learning style. The interval of score is 0-3.

3.3.2 Learner Analysis Experiment

Learner analysis is the first process to develop the learner model because we have to know the style of our learners for developing the suitable learner model in our system.

3.3.2.1 Participants and Methods

In this study, we examined the learning style of third and fourth-year students in major of Computer Science (CS) and Information Technology (IT) of faculty of Science at Thaksin University during the 2008 academic year.

The Index of Learning Styles (ILS)-Thai version was administered to all participants. Students were asked to complete the self-administered questionnaire at the end of one lecture period in the first semester. This instrument consisted of 44-item sentences in Thai language, translated with permission from the English version (Felder and Soloman, 1998). ILS was developed by Babara A. Soloman and Richard M. Felder of North Carolina State University, USA and was validated in (Felder and Spurlin, 2005) and (Zywno, 2003). It is an instrument use to assess preferences on four dimension of learning style mode l(see in Table 3.1). Each dimension of the ILS as a two-pan scale, with each pan representing one of the two categories of the dimension (for example, sensing and intuiting), and weights in a pan representing skills associated with that category. If you have a preference for sensing, it means you have more weights in the sensing pan than the intuitive pan, and conversely if you have a preference for intuition.

3.3.2.2 Experimental Results

Of the learners in the 2008 cohort, 142 learners participated in the study by completing the ILS.

Table 3.2: The example of learner learning style classifications in each dimension.

LearnerID	AS	RS	SS	IS	US	BS	QS	GS
00001	9	2	4	7	11	0	3	8
00002	7	4	6	5	6	5	6	5
00003	7	4	8	3	11	0	7	4
00004	9	2	8	3	7	4	5	6
00005	4	7	7	4	7	4	5	6
00006	9	2	10	1	9	2	4	7
00007	8	3	7	5	8	3	4	7
00008	4	7	6	5	8	3	3	8
00009	9	2	9	2	8	3	6	5
00010	8	3	7	4	9	2	3	8
00011	9	2	8	3	9	2	6	5
00012	7	4	7	4	4	7	2	9
00013	9	3	5	6	8	3	6	5
00014	5	6	9	2	8	3	7	4
00015	6	5	10	1	11	0	7	4
00016	8	3	5	6	10	1	6	5
00017	7	4	7	4	7	4	6	5
00018	8	3	9	2	4	7	5	6
00019	9	2	8	3	8	3	7	4
00020	9	2	7	4	9	2	7	4
00021	9	2	7	4	6	5	4	7
00022	8	3	6	5	7	4	6	5
00023	6	5	5	6	8	3	2	9
00024	6	5	7	4	8	3	6	5
00025	7	4	6	5	6	5	4	7
00026	6	5	7	4	9	2	5	6
00027	10	1	8	3	8	3	9	2
00028	8	3	7	4	10	1	4	7
00029	9	2	8	3	9	3	5	6
00030	6	5	9	2	8	3	6	5
00031	8	3	9	2	4	7	4	7
00032	8	3	7	4	5	6	3	8
00033	8	3	8	3	10	1	4	7
00034	8	3	8	3	5	6	6	5

For defining the degree of preference, we define the three level of weight value of each learning preference as: strong weight = 1, medium weight = 0.5, weak weight = 0. Table 3.4 shows the result of 12 weighted patterns defined of fourth-year IT learners from the scores in Table 3.3. Each learner has eight preferences (A, R, S, I, U, B, Q, G) with different weights to describe their learning preference degrees. For example, if *A_weight = 1*, it means the learner has a strong “Active” preference.

The learner style preference is converted into the form of weight value for providing the computational process. The example of converting by using the information from Table 3.2 is shown in Table 3.3.

Table 3.3: The leaner classifications that categorized in 12 weighted patterns.

Learners Classification with Preference Weight Patterns[a]

																#Cases	LearnerID
A_weight	0.5	R_weight	0.5	S_weight	0.5	I_weight	0.5	U_weight	0.5	B_weight	0.5	Q_weight	0.5	G_weight	0.5	1	2
																2	5
																3	17
																4	25
													0	G_weight	1	1	12
									1	B_weight	0	Q_weight	0	G_weight	1	1	8
																2	23
													0.5	G_weight	0.5	1	24
																2	26
					1	I_weight	0	U_weight	1	B_weight	0	Q_weight	0.5	G_weight	0.5	1	3
																2	14
																3	15
																4	30
	1	R_weight	0	S_weight	0.5	I_weight	0.5	U_weight	0.5	B_weight	0.5	Q_weight	0.5	G_weight	0.5	1	21
																2	22
													0	G_weight	1	1	32
									1	B_weight	0	Q_weight	0	G_weight	1	1	1
																2	10
													0.5	G_weight	0.5	1	7
																2	13
																3	16
																4	20
																5	28
					1	I_weight	0	U_weight	0.5	B_weight	0.5	Q_weight	0.5	G_weight	0.5	1	4
																2	18
																3	31
																4	34
									1	B_weight	0	Q_weight	0.5	G_weight	0.5	1	6
																2	9
																3	11
																4	19
																5	29
																6	33
													1	G_weight	0	1	27

All of 142 learners are evaluated with ILS questionnaire. The results of each learner group in form of the preference type and preference level are shown in Table 3.4-3.7. Then, we summarize the nature of the population of the learner learning style survey in Figure 3.3.

Table 3.4: Reported learning style preferences of third-year IT's learners.

Preference Type	Preference Level	N [N =31]	% of Total N	Preference Type	Preference Level	N [N =31]	% of Total N
Active	Weak	1	3.2%	Visual	Weak	0	0%
	Medium	16	51.6%		Medium	13	41.9%
	Strong	14	45.2%		Strong	18	58.1%
Reflective	Weak	14	45.2%	Verbal	Weak	0	0%
	Medium	16	51.6%		Medium	18	58.1%
	Strong	1	3.2%		Strong	13	41.9%
Sensing	Weak	1	3.2%	Sequential	Weak	8	28.5%
	Medium	19	61.3%		Medium	22	71.0%
	Strong	11	35.5%		Strong	1	3.2%
Intuitive	Weak	11	35.5%	Global	Weak	1	3.2%
	Medium	19	61.3%		Medium	22	71.0%
	Strong	1	3.2%		Strong	8	28.5%

Table 3.5: Reported learning style preferences of third-year IT's learners.

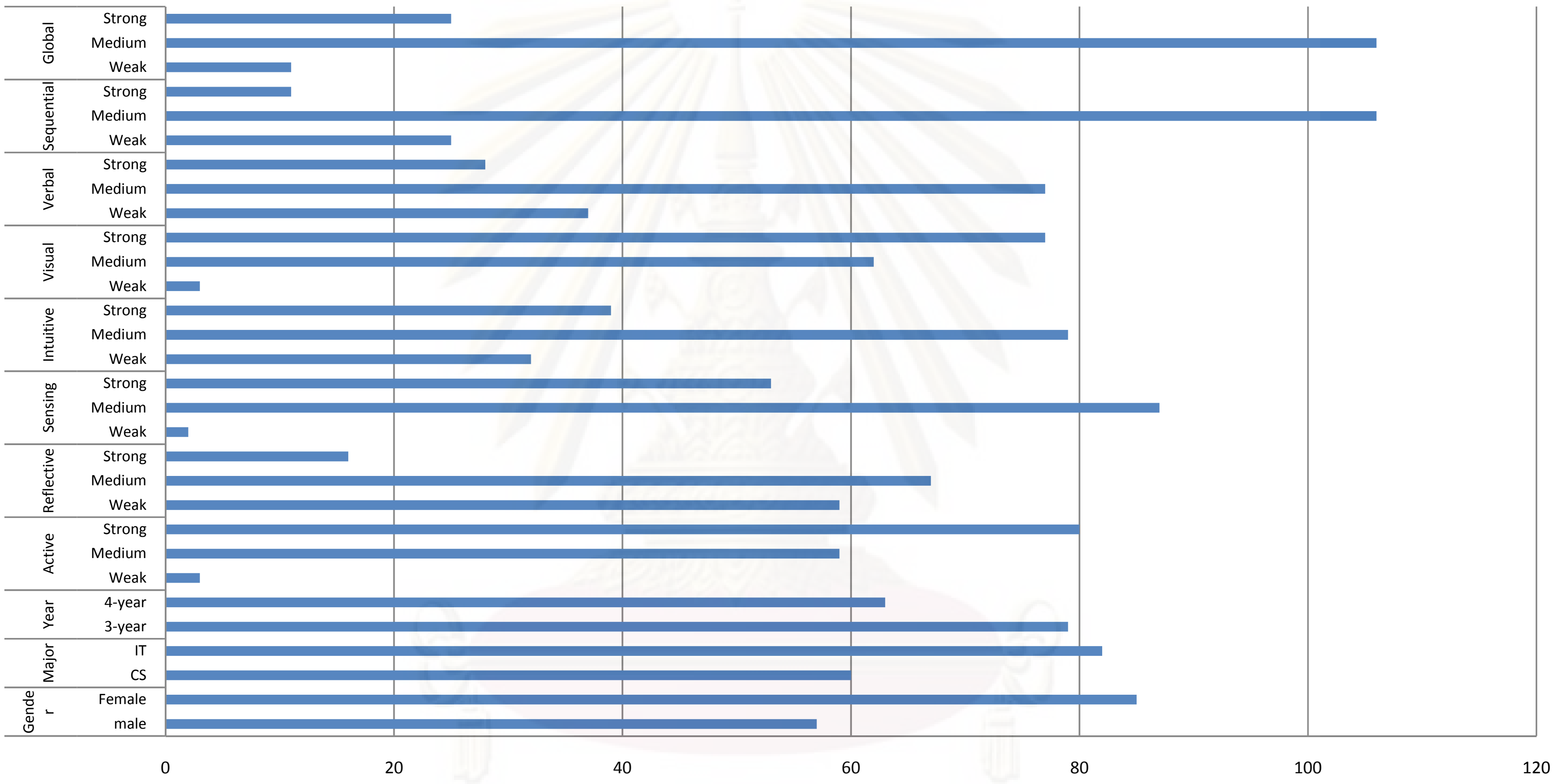
Preference Type	Preference Level	N [N =48]	% of Total N	Preference Type	Preference Level	N [N =48]	% of Total N
Active	Weak	1	2.1%	Visual	Weak	1	2.1%
	Medium	17	51.6%		Medium	25	52.1%
	Strong	30	62.5%		Strong	22	45.8%
Reflective	Weak	30	62.5%	Verbal	Weak	22	45.8%
	Medium	17	51.6%		Medium	25	52.1%
	Strong	1	2.1%		Strong	1	2.1%
Sensing	Weak	1	2.1%	Sequential	Weak	8	16.7%
	Medium	32	66.7%		Medium	35	72.9%
	Strong	15	31.3%		Strong	5	10.4%
Intuitive	Weak	15	31.3%	Global	Weak	5	10.4%
	Medium	32	66.7%		Medium	35	72.9%
	Strong	1	2.1%		Strong	8	16.7%

Table 3.6: Reported learning style preferences of fourth-year CS's learners.

Preference Type	Preference Level	N [N=29]	% of Total N	Preference Type	Preference Level	N [N=29]	% of Total N
	Weak	1	3.4%		Weak	2	6.9%
Active	Medium	13	44.8%	Visual	Medium	12	41.4%
	Strong	15	51.7%		Strong	15	51.7%
	Weak	15	51.7%		Weak	15	51.7%
Reflective	Medium	13	44.8%	Verbal	Medium	12	41.4%
	Strong	1	3.4%		Strong	2	6.9%
	Weak	0	0%		Weak	3	10.3%
Sensing	Medium	17	58.6%	Sequential	Medium	22	75.9%
	Strong	12	41.4%		Strong	4	13.8%
	Weak	12	41.4%		Weak	4	13.8%
Intuitive	Medium	17	58.6%	Global	Medium	22	75.9%
	Strong	0	0%		Strong	3	10.3%

Table 3.7: Reported learning style preferences of fourth-year IT's learners.

Preference Type	Preference Level	N [N=34]	% of Total N	Preference Type	Preference Level	N [N=34]	% of Total N
	Weak	0	0%		Weak	0	0%
Active	Medium	13	38.2%	Visual	Medium	12	35.3%
	Strong	21	61.8%		Strong	22	64.7%
	Weak	0	0%		Weak	0	0%
Reflective	Medium	21	61.8%	Verbal	Medium	22	64.7%
	Strong	13	38.2%		Strong	12	35.3%
	Weak	0	0%		Weak	6	17.6%
Sensing	Medium	19	55.9%	Sequential	Medium	27	79.4%
	Strong	15	44.1%		Strong	1	2.9%
	Weak	0	0%		Weak	1	2.9%
Intuitive	Medium	15	44.1%	Global	Medium	27	79.4%
	Strong	19	55.9%		Strong	6	17.6%



	Gender		Major		Year		Active			Reflective			Sensing			Intuitive			Visual			Verbal			Sequential			Global		
	male	Female	CS	IT	3-year	4-year	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong	Weak	Medium	Strong
Number of Student	57	85	60	82	79	63	3	59	80	59	67	16	2	87	53	32	79	39	3	62	77	37	77	28	25	106	11	11	106	25

Figure 3.3: Nature of the population of the learner learning style survey.

49

We notice that in active/reflective preference dimension, the most learners prefer the strong active preference (80 learners) and a few learners prefer strong reflective preference (16 learners). In visual/verbal, many learners prefer the strong visual preference (77 learners) and a few learners prefer strong verbal preference (28 learners). The results of this two cases show that the probability of learning object preferring of active and visual learning object is obviously higher than others. So, we can define that the features of learning object that related with both of active/reflective and visual/verbal will be closely considered. This is the implicit information to be used in the process of matching between learner and learning object based on their learning style.

3.3.3 Set of Learner's Learning Style

The result of the learner's learning style analysis from above subsection is used for creating the learner's learning style set. We defined the definition of learning style set of each learner as definition 3.1.

Definition 3.1: *Learner Style Set* $LSS(L) = \{(P_i, Pw_i)\} \mid P_i \in \{A, R, S, I, U, B, Q, G\}$, Pw_i *is the weight which has interval [0-1] of each* P_i *and i is number of learning styles.*

For example, for a particular learner L_1 we might have $LSS(L_1)$= {(A,1), (R,0), (S,0.5), (I,0.5), (U,1,), (B, 0), (Q,0), (G,1)}

3.3.4 Associating Learning Style Set to Learner Model

For generating the learner preference set (LPS) that describes the preferred learning object features of learner, we develop the learning object selection rules for matching the learner preference to suitable features of learning object (LO-learner preference matching). The learning object selection rules developments in proposed approach are presented as follows:

3.3.4.1 Learning Object Feature Analysis and Selection

Based on IEEE LOM standard, there are many kinds of metadata but we do not need all of them. The learning object features for recommendation process are analyzed from proposed learners. So, the theory of Felder and Silverman learning style model is considered and reviewed an existing system which uses this model for the learners. In many researches defined the required features of learning object for their recommendation system. In our learning object's feature selection, we collected the popular features of IEEE LOM that was proposed by Manouselis and Samson in the Nemo project (Manouselis and Samson, 2005) and adjust them into form of questionnaire for evaluating the importance of selected features. The questionnaire consist of 20 features asking for the learner's opinion on the importance of features of learning object, such as presentation format, size, learning resource type, etc. Thirty-one learners rate the score of feature that they think it is suitable for identifying the recommendation with the scale 1 to 5 (1= very disagree, 2 = disagree, 3 = common, 4 = agree, 5= very agree). We defined the threshold value for selecting the strong rating feature as α= 0.7, where $\alpha = \frac{Feature\ Score}{Total\ Score}$, then the results of feature selection are shown in Table 3.8. Please refer to Appendix A.1 for detail of the questionnaire and in Appendix B.1 for detail of learner opinion results.

Table 3.8: Learning objects feature score rated by 31 learners.

Feature		Feature Score (Number of Learners = 31, Total Score=5*31)	Normalized Score (α = 0.7)
LOM category	element		
General	Title	67	0.4323
	Language	140	0.9032
	Description	119	0.7677
	Structure	83	0.5355
	Aggregation	67	0.4323

Table 3.8: Learning objects feature score rated by 31 learners. (cont.)

Feature		Feature Score (Number of Learners = 31, Total Score=5*31)	Normalized Score (α = 0.7)
LOM category	element		
Technical	Format	129	0.8323
	Size	60	0.3871
	Location	61	0.3935
Educational	Interactivity Type	111	0.7161
	Learning Resource Type	143	0.9226
	Interactivity Level	109	0.7032
	Semantic Density	112	0.7226
	Context	69	0.4452
	Difficulty	87	0.5613
	Auditory Loudness	89	0.5742
	Color Brightness	90	0.5806
	Color Complexity	86	0.5548
	Detail of Sound	84	0.5419
	Detail of Text	85	0.5484
	Detail of Sentence	96	0.6194

Table 3.8 summarizes the result of feature analysis and selecting with α=0.7, a set of selected features is shaded, these are { Language, Description, Format, Interactivity Type, Learning Resource Type, Interactive Level, Semantic Density}.

In general LMS, we can filter learning objects with the feature "Language" by no need to know the learning style of learners, so we can discard this feature in proposed learning style-based recommendation. In the same reason, the feature "Description" is not required for a basis of learning style-based approach. So, the rest features and their value space based on IEEE LOM metadata in proposed recommendation algorithm are identified in Table 3.9.

Table 3.9: The selected features for learning object recommendation algorithm.

Feature ID	Name	Element Path	Value Space
F1	Format	LOM/Technical/Format	Video, Image, Text, Audio, Animation
F2	Interactivity Type	LOM/Educational/Interactivity_Type	Active, Expositive, Mixed
F3	Interactivity Level	LOM/Educational/Interactivity_Level	Very low (0), Low (1), Medium (2), High (3), Very high (4)
F4	Semantic Density	LOM/Educational/Semantic_Density	Very low (5), Low (6), Medium (7), High (8), Very high (9)
F5	Learning Resource Type	LOM/Educational/Learning_Resource_Type	Exercise, Simulation, Experiment, Definition, Algorithm, Example, Slide, Index

This feature set will be used in proposed recommendation algorithm. We define this feature set when used to describe learning object by definition 3.2.

Definition 3.2: *Learning Object Set* LOS_{LO} *is the discrete set of all selected learning object feature for describing the characteristic of the specific learning object.* LOS_{LO} *is denoted by* $LOS_{LO} = \{F_1, F_2, F_3, F_4, F_5 \mid \forall F_i \in LOM, F_i \neq F_j\}$.

For example, three learning objects were explained by definition 3.2 that was defined in Table 3.10 as follows:

LOS_{LO001} = { *F1, F2, F3, F4, F5*} = { *animation , active, 4, 8, simulation* }

LOS_{LO002} = { *F1, F2, F3, F4, F5*} = { *text , expositive, 2, 7, algorithm* }

LOS_{LO003} = { *F1, F2, F3, F4, F5*} = { *video , active, 4, 7, definition* }.

3.3.4.2 Mapping Selected Learning Object Features to Learner's Learning Style

Felder and Silverman defined learning style in eight learning styles: Active, Reflective, Sensing, Intuitive, Visual, Verbal, Sequential and Global. The semantic groups associated with the ILS answers are explained in Section 3.2.4.2.1.

The values of these properties constitute the input for the planner to generate a recommendation adjusted to the learner preferences and their learning styles. However, this process is only possible if there is an implicit relationship between the learners' characteristics and the different kinds of learning object and activities associated to the learning design. If learning objects are characterized with metadata, rules can be applied to assign learning object to the learner's learning style in LMS. In this work, IEEE LOM is used to characterize the learning objects. In Section 3.2.4.2.2, the relationship between the different Felder's dimensions for each learning style and LOM feature of the learning objects is presented. An appropriate learning object is one which addresses at least one characteristics of learner.

3.3.4.2.1 Grouping of Learning Style Preference

In an empirical study (Graf et al., 2007), the groups of preferences within each dimension of Felder and Silverman learning style model were analyzed and their relevance for each dimension was investigated. Table 3.10 shows the proposed groups as well as the related answers of ILS questions (Felder and Soloman, 2007) for each group. A question may appear twice in the table, if the two possible answers to the question point to two different groups.

The semantic groups (SG) within the dimensions provide relevant information in order to be able to identify learning styles. For example, if a learner has a preference for trying things out and tends to be more impersonal oriented, learner would have a balanced learning style on the active/reflective dimension. However, a learner has also a balanced learning style if they prefer to think about the material and tends to be more social oriented. Although both learners have different preferences and therefore different behavior in an online course, both are considered according to the result of ILS. Considering the proposed semantic groups leads therefore to more accurate information about learners' preferences and to a more accurate model for identifying learning styles based on the behavior of learners in an online course.

Table 3.10: Semantic groups associated with the ILS answers.

Learning Style	Semantic Group	ILS questionnaire indicator #		Extracted words for validating mapping rule
		answer 'a'	answer 'b'	
A-Active	Trying something out (SG1)	1	-	Try out
		17		Start solution immediately
		25		Try out
		29		Practice

Table 3.10: Semantic groups associated with the ILS answers. (cont.)

Learning Style	Semantic Group	ILS questionnaire indicator # answer 'a'	ILS questionnaire indicator # answer 'b'	Extracted words for validating mapping rule
	Social oriented (SG2)	5	-	Talk
		9		Contribute idea
		13		Group
		21		Group
		33		Group
		37		Group, outgoing
		41		Group
R-Reflective	Think about material (SG3)	-	1	Think it though
			5	Think about it
			17	Try to understand
			25	Think about it
			29	Think about it
	Independent (SG4)	-	9	Listen
			13	Independent
			21	Independent
			33	Independent
			37	Independent, reserved
			41	Independent
S-Sensing	Existing way (SG5)	2	-	Realistic
		30		Reality
		34		Sensible

Table 3.10: Semantic groups associated with the ILS answers. (cont.)

Learning Style	Semantic Group	ILS questionnaire indicator #		Extracted words for validating mapping rule
		answer ‘a’	answer ’b’	
	Concrete material (SG6)	6	-	Real situation
		10		Fact
		14		Fact
		18		Certainly
		26		Fact
		38		Concrete, fact, data
	Careful with details (SG7)	22	-	Careful detail
		42		Careful detail
I-Intuitive	New ways (SG8)		2	Innovative
			14	New idea
			22	Creative
			26	Creative
			30	New ways
			34	Imagination
	Abstract material (SG9)		6	Idea, theory
			10	Concept
			18	Theory
			38	Concept, theory
	Not careful with details (SG10)		42	Not careful with details

Table 3.10: Semantic groups associated with the ILS answers.(cont)

Learning Style	Semantic Group	ILS questionnaire indicator #		Extracted words for validating mapping rule
		answer 'a'	answer 'b'	
U-visUal	Pictures (SG11)	3	-	Picture
		7		Picture, diagram, graph,map
		11		Picture, chart
		15		Diagram
		19		Want to see
		23		Map
		27		Picture
		31		Chart, graph
		35		Remember by looking
		39		Watch,
		43		Picture
B-verBal	Spoken word (SG12)	-	3	Word
			7	Written direction, verbal information
			15	Explaining
			19	Hear
			27	Said
			35	Said

Table 3.10: Semantic groups associated with the ILS answers. (cont.)

Learning Style	Semantic Group	ILS questionnaire indicator # answer 'a'	ILS questionnaire indicator # answer 'b'	Extracted words for validating mapping rule
	Written word (SG13)	-	3	Word
			7	Written direction, verbal information
			11	Written text
			23	Written instruction
			31	Summarizing text
			39	Read
			43	Without much detail
	Difficulty with visual style (SG14)	-	43	Difficultly with picture
Q-seQuential	Detail oriented (SG15)	4	-	Understand detail
		28		Focus on detail
		40		Outline are somewhat helpful
	Sequential progress (SG16)	20	-	Sequential step
		24		Sequential
		32		Work on beginning, progress forward
		36		Focus on subject
		44		Step solution

Table 3.10: Semantic groups associated with the ILS answers.(cont.)

Learning Style	Semantic Group	ILS questionnaire indicator # answer 'a'	ILS questionnaire indicator # answer 'b'	Extracted words for validating mapping rule
	From part to the whole (SG17)	8 12 16	-	From part to the whole One step at time Think of incident, put them together
G-Global	Overall picture (SG18)	-	4 8 12 16 28 40	Understand overall From the whole to part Overall picture Know theme Big picture Outline
	Non-sequential progress (SG19)	-	24 32	Global Work on different part
	Relations/connection (SG20)	-	20 36 44	Overall picture Make connection among Wide range solution

In this analysis process we define the learner characteristics required to generate recommendations according to learning styles and collaborative competences. Furthermore, we describe the mechanism to link together those features with learning objects and resources to be generated for creating the learning object selection rules.

Table 3.11 presents the domain knowledge of Learning Object Set (LOS). We may infer from LOS definition 3.2 that $LOS_{LO} = \{F_1, F_2, F_3, F_4, F_5 \mid \forall F_i \in LOM, F_i \neq F_j\}$. Since LOS $\subseteq$ LOM where LOM can always describes every learning object LO_i, the result implies directly that LOS can always describes every learning object LO_i. If we can define mapping rules that cover all LOS features, every LO_i can be accessed. Table 3.12 presents the LOM value spaces analysis in learning object set (LOS) domain. Vi is defined as the LOM value space, where i is value space (V) number i. The knowledge will be used in mapping rule construction and validation.

Table 3.11: LOM value spaces analysis in Learning Object Set (LOS) Domain. (Wiktionary, 2009) (Cancore, 2004) (LOM, 2002)

Feature of LOS	Feature Description	LOM value space	LOS Domain
Format (F1)		Video (V1)	"I see" , "moving eye picture", "a recording of both the visual and audible components"
		Image (V2)	"Two-dimensional figure", "map", "graph", "pie chart", "abstract painting", "computer graphic", "drawing", "painting", "photograph", "visual media", "picture", "idea"
		Text (V3)	"set of writing", "message"

Table 3.11: LOM value spaces analysis in Learning Object Set (LOS) Domain. (cont.)

Feature of LOS	Feature Description	LOM value space	LOS Domain
		Audio (V4)	"hear", "listen", "sound"
		Animation (V5)	"motion picture", "the act of animating", "spirit", "liveliness", "airiness", "sequence of image"
Interactivity Type (F2)	Predominant mode of learning supported by this learning object Indicate whether the object requires action on the part of the user approaches	Active(V6)	"simulation", "questionnaire", "exercise", "problem", "practice"
		Expositive(V7)	"hypertext", "graphics", "audio", "essay"
		Mixed(V8)	"video", "simulation"
Interactivity Level (F3)	The degree of interactivity characterizing this learning object. Interactivity in this context refers to the degree to which the learner can influence the aspect or behavior of the learning object.	Very low (V9)	"text", "message"
		Low (V10)	"audio", "sound"
		Medium (V11)	"image", "hypertext", "online multiple choice"
		High (V12)	"video", "simulation"
		Very high (V13)	"animation", "motion picture", "3-D simulation"

Table 3.11: LOM value spaces analysis in Learning Object Set (LOS) Domain. (cont.)

Feature of LOS	Feature Description	LOM value space	LOS Domain
Semantic Density (F4)	The degree of conciseness of a learning object. The semantic density of a learning object may be estimated in terms of its size, span or – in the case of self-timed resources such as audio or video – duration.	Very low (V14)	“message”, “text”
		Low (V15)	“definition”, “image”
		Medium (V16)	“audio”
		High (V17)	“video”, “exercise”
		Very high (V18)	“simulation”, “experiment”
Learning Resource Type (F5)	Specific kind of learning object.	Exercise (V19)	“planned sequence of actions ”, “assignment”, “worksheet”, “tutorial”
		Simulation (V20)	“behavior of some situation”, “visual training”
		Experiment (V21)	“discover unknown”, “test hypothesis” , “establish some know truth ”
		Definition (V22)	“explanation”, “give meaning”, “objective”
		Algorithm (V23)	“step for action”

Table 3.11: LOM value spaces analysis in Learning Object Set (LOS) Domain. (cont.)

Feature of LOS	Feature Description	LOM value space	LOS Domain
		Example (V24)	"case study", "show how to act"
		Index (V25)	"glossary", "reference", "reference list", "list of content"
		Slide (V26)	"photographic transparency", "sequential step"

Next, the information from both of Table 3.10 and 3.11 are considered the semantic of their mapping. The valid mapping rule is the rule that is the member of the intersection set of word meaning or semantic between semantic group (SG) and LOS features. Figure 3.4 presents the mapping process between learning style and LOS.

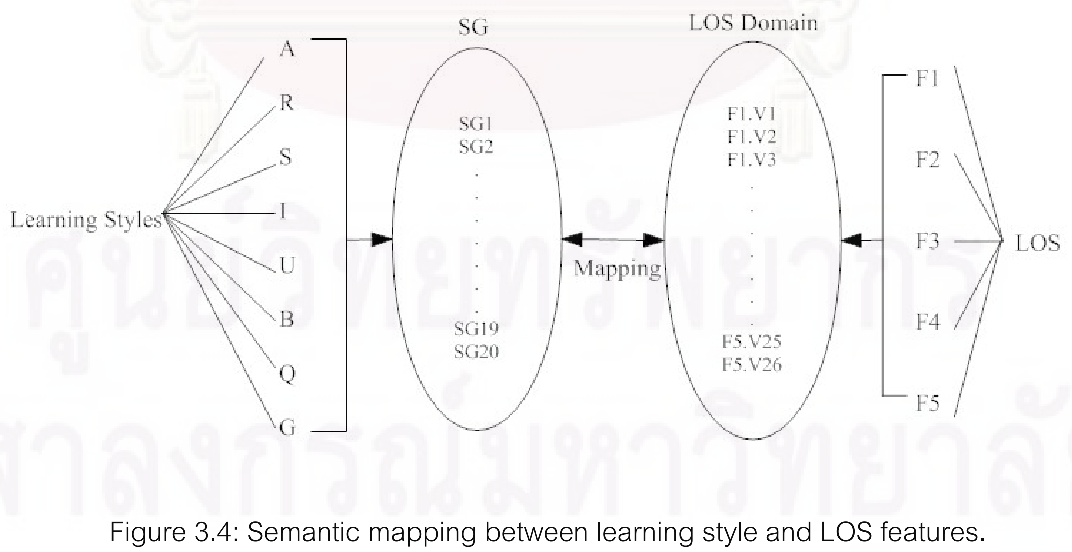


Figure 3.4: Semantic mapping between learning style and LOS features.

All mapping rules are used to define what learning objects are presented to each learner are described in subsection 3.3.4.3.

3.3.4.2.2 Learning Object Mapping Rules Construction and Validation

A common way to perform the analysis of mapping is to let the domain knowledge of learning style and learning object features perform this task, and word analysis support this process. Figure shows the mapping rules building process.

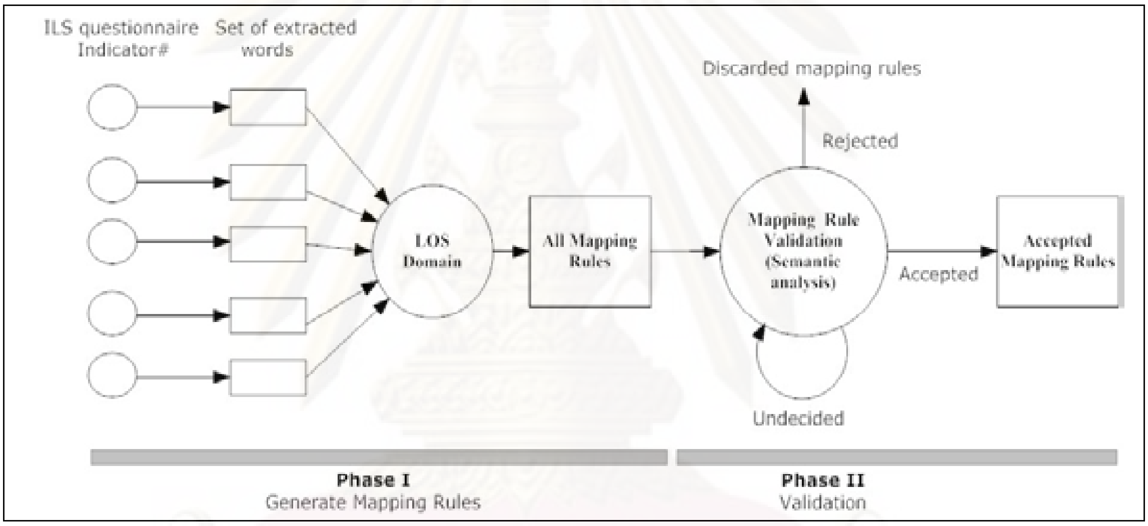


Figure 3.5: Learning object mapping rules building process.

In proposed approach, learning style and learning object feature mapping rules discovered with Learning Object Set (LOS) domain are validated by the expert, and, depending on how well they represent the actual behaviors of the learner, some rules are "accepted" and some "rejected" by the expert.

In Phase I, the mapping rule generation, mapping rules describing the learning object preferences of individual learners are generated from the learners' ILS answer as was described in Section 2.2. Phase II constitutes the mapping rule validation process. Mapping rule validation, unlike rule discovery (Phase I), is not performed separately for each learner, but rather for all learners at once. The reason we propose performing mapping rule validation collectively (rather than individually) for all learners is that there are usually many similar or even identical rules across different learners.

All mapping rules are collected into one set. The mapping rule validation process is performed as a second part of Phase II. This process is described in Figure 3.5. All mapping rules are considered invalidated. We analyze the meaning of extracted words from 44 ILS answers and compare with learning object features in LOS. Then, the validation mapping as O is defined and applies them successively to the set of invalidated mapping rules. The application of each validation results in validation of some of the rules. In particular, some mapping rules get accepted and some rejected (sets O_{accept} and O_{reject} in Algorithm 1). Then, the next validation mapping would be applied to the set of the remaining invalidated rules (set $MR_{invalid}$). This validation process stops when the Terminate Validation Process condition is met. Our condition is that the set of validated mapping rules are covered by LOS domain (all learning objects features are referred). After the validation process is stopped, the set of all the discovered rules (MR_{all}) is split into three disjoint sets: accepted rules (MR_{accept}), rejected rules (MR_{reject}), and possibly some remaining invalidated rules ($MR_{invalid}$). At the end of Phase II all the accepted mapping rules are used to transform the learning style set (LSS) to learner preference set (LPS).

Algorithm 1: Mapping Rules Validation Process
Input: Set of all discovered mapping rules MR_{all}. Output: Sets of mapping rules MR_{accept}, MR_{reject}, $MR_{invalid}$ such that $MR_{all} = MR_{accept} \cup MR_{reject} \cup MR_{invalid}$ Methods: $MR_{invalid}$: = MR_{all}, MR_{accept}:=ϕ, MR_{reject} :=ϕ. While (not TerminateValidationProcess()) begin Expert selects a validation operator (called, O) from the set of available

validation mapping.

O is applied to $MR_{invalid}$, Result: disjoint sets O_{accept} and O_{reject}.

$MR_{invalid} := MR_{invalid} - O_{accept} - O_{reject}$, $MR_{accept} := MR_{accept} \cup O_{accept}$,

$MR_{reject} := MR_{reject} \cup O_{reject}$

End

Figure 3.6: Mapping rules validation process.

Based on Felder and Silverman learning style model, the association between each learning style and the learning object features is analyzed. Figure 3.7-3.15 demonstrates the example of validated mapping rule selection from all possible mapping rules.

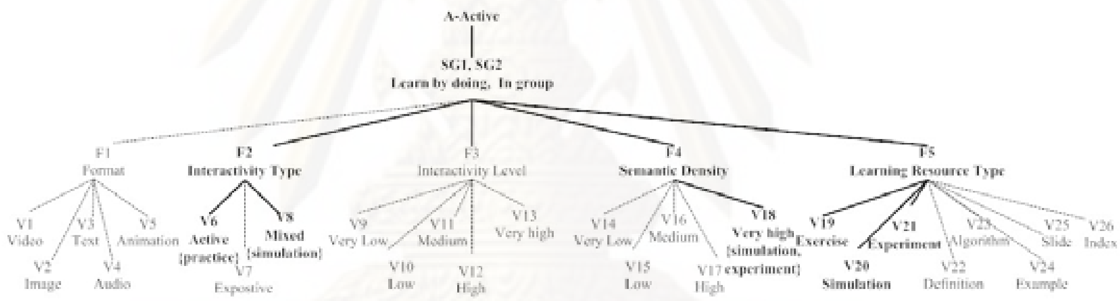


Figure 3.7: Mapping active style to LOS features.

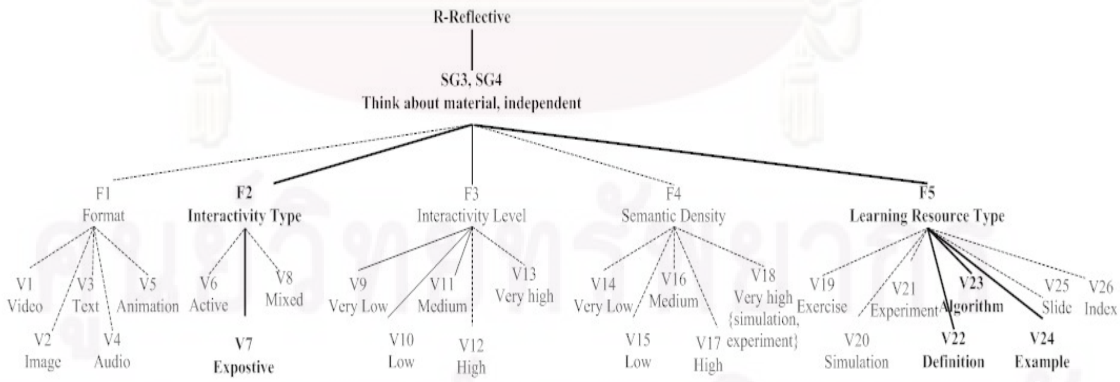


Figure 3.8: Mapping reflective style to LOS features.

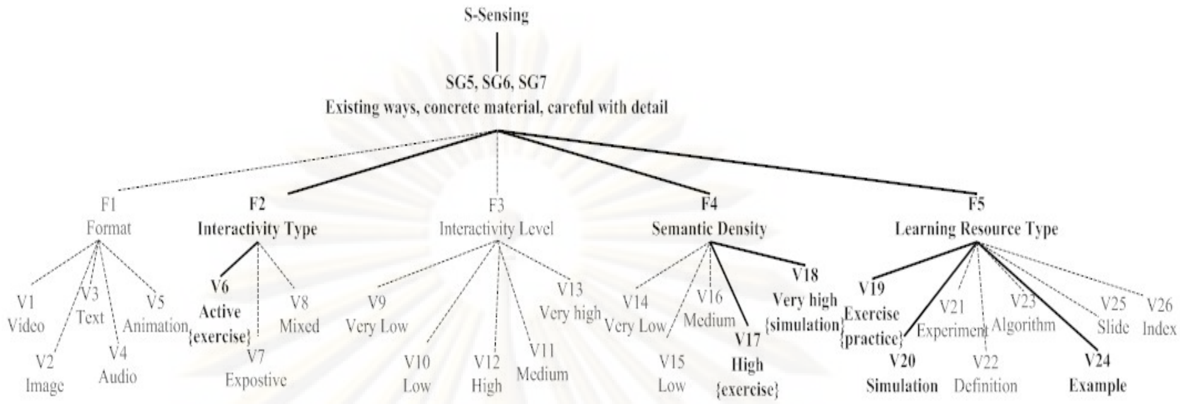


Figure 3.9: Mapping Sensing style to LOS features.

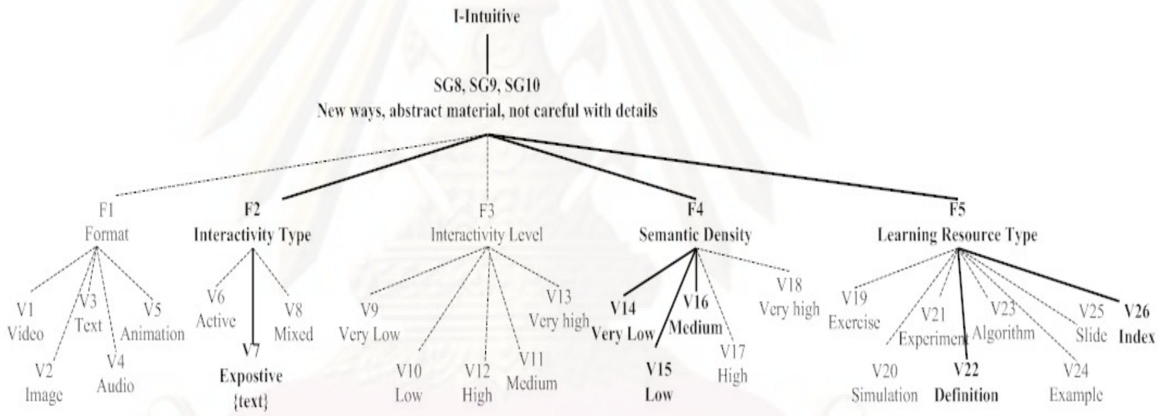


Figure 3.10: Mapping intuitive style to LOS features.

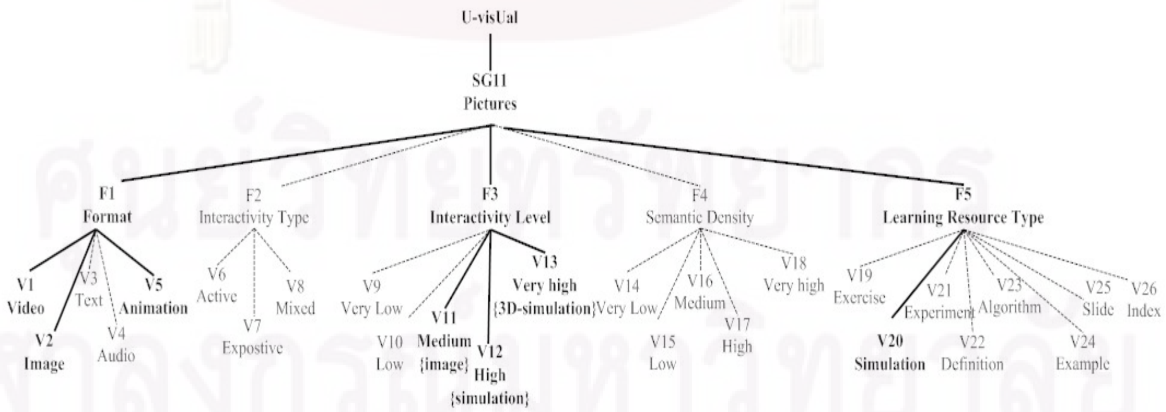


Figure 3.11: Mapping visual style to LOS features.

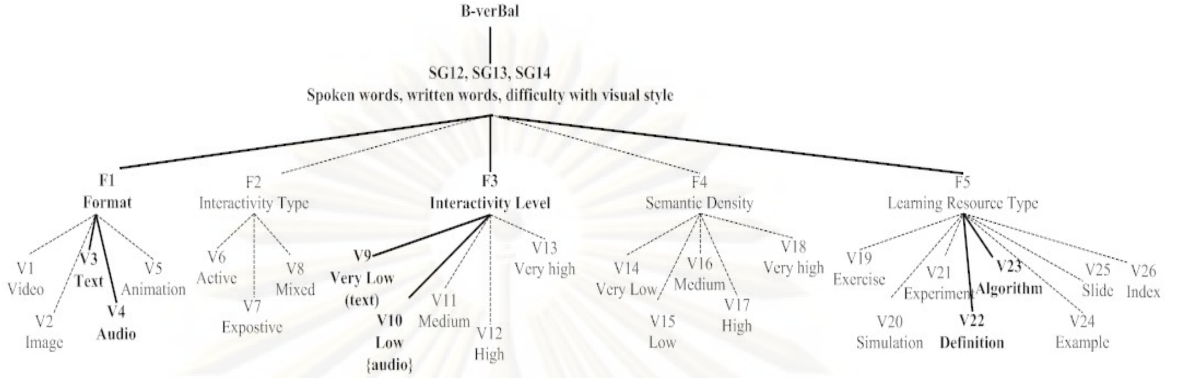


Figure 3.12: Mapping visual style to LOS features.

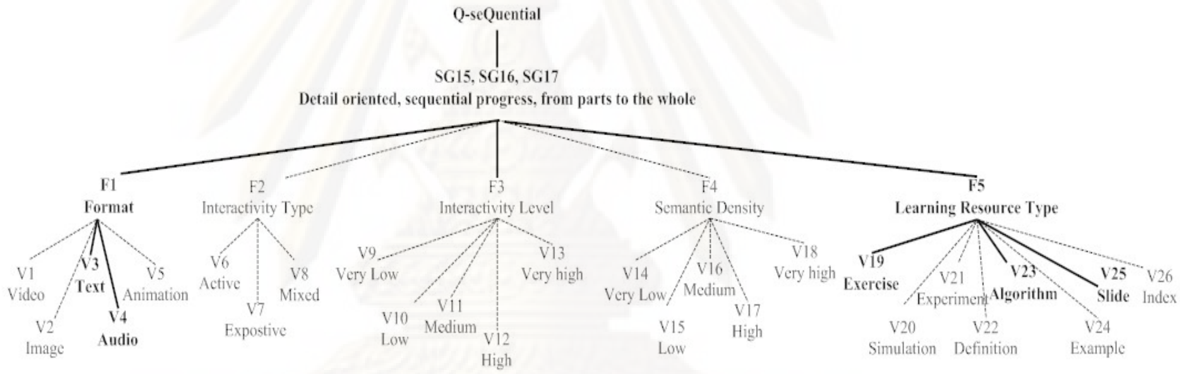


Figure 3.13: Mapping sequential style to LOS features.

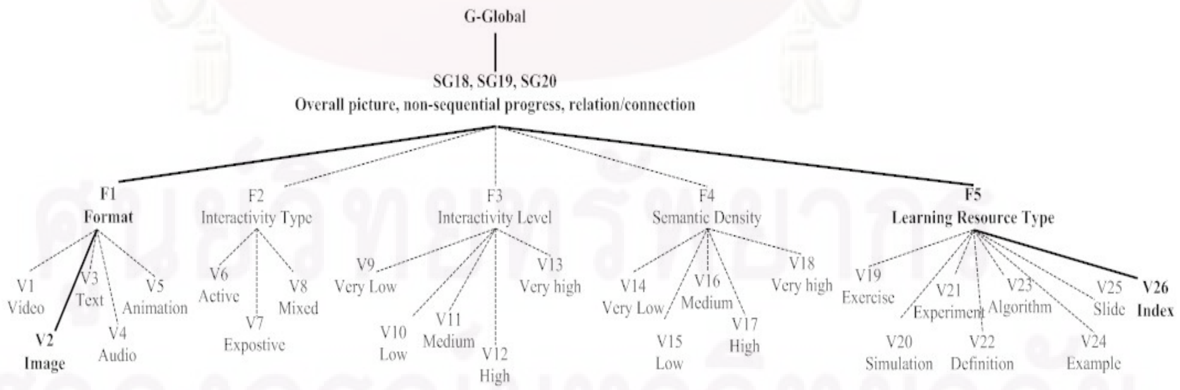


Figure 3.14: Mapping global style to LOS features.

From Figure 3.7-3.14, we construct the relation between learning style preferences and learning object features in association of rule-based that are presented in Figure 3.15. The learning object features are based on IEEE LOM standard and matched with the selected learning object features in LOS.

Learning Object Mapping Rules

Mapping 1. Recommend learning object for "**A**-**A**ctive" learner

If "A" ∈ LSS(L)

Then LOM.educational.interactivity_type = "active" or "mixed"

And LOM.educational.LearningResourceType = "exercise" or "simulations" or "experiment"

Mapping 2. Recommend learning object for "**R**-**R**eflective" learner

If "R" ∈ LSS(L)

Then LOM.educational.interactivity_type = "expositive"

And LOM.educational.ResourceType = "definition" or "algorithm" or "example"

Mapping 3. Recommend learning object for "**S**-**S**ensing" learner

If "S" ∈ LSS(L)

Then LOM.educational.semanticDensity = "high" or very "high"

And LOM.educational.learningResourceType = simulation or experiment

Mapping 4. Recommend t learning object for "**I**-**I**ntuitive" learner

If "I" ∈ LSS(L)

Then LOM.educational.semanticDensity =" very low" or "low or medium"

And LOM.educational.learningResourceType = "definition" or "exercise"

Mapping 5. Recommend learning object for "**U**-vis**U**al" learner

If "U" ∈ LSS(L)

Then LOM.technical.format = “video” or “image” or “animation”

And LOM.educational.interactivity_level= “high” or “very high”

And LOM.educational.learningResourceType = “simulation”

Mapping 6. Recommend learning object for “**B**-ver**B**al” learner

If “B” ∈ LSS(L)

Then LOM.technical.format = “text” or “audio”

And LOM.educational.interactivity_level= “medium” or “low” or” very low”

And LOM.educational.learningResourceType = “definition” or “exercise”

Mapping 7. Recommend learning object for “**S**-se**Q**uential” learner

If “Q” ∈ LSS(L)

Then LOM.technical.format = “text” or “audio”

And LOM.educational.learningResourceType = “exercise” or “algorithm” or “ slide”

Mapping 8. Recommend learning object for “**G**-**G**lobal” learner

If “G” ∈ LSS(L)

Then LOM.technical.format = “image”

And LOM.educational.learningResourceType = “index”

Figure 3.15: Mapping rules between learning object features and learning styles.

Eight mapping rules that are described above are covered by the LOS domain and can map in every value space of LOS feature. So, all learning objects in this system will be accessed for learning.

3.3.4.2.3 Mapping Rule Accepting

Based on the process of word analysis process in Section 3.3.4.2.1, the proposed mapping rules accepting (validation mappings O in algorithm 1)are demonstrated as follows:

Mapping 1.

Active = {try out, start solution, immediately, practice, talk, contribute idea, group}

Map to:

Interactivity type

Interactivity type = "active" = {simulation, questionnaire, exercise, problem, practice}

Interactivity type = "mixed"= {video, simulation}

Interactivity type = "expositive" = {hypertext, graphics, audio, essay}

Learning Resource Type

Learning resource type = "exercise"= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = "simulation"= {behavior of some situation, behavior of process}

Learning resource type = "experiment"= {discover unknown, test hypothesis, establish some know truth}

Learning resource type="definition"= {explanation, give meaning, objective}

Learning resource type="example"= {case study, show how to act}

Learning resource type="index"= {glossary, reference, list of content}

Mapping 2.

Reflective = {think about it, try to understand, listen, independent, reserved}

Map to:

Interactivity type

Interactivity type = "active" = {simulation, questionnaire, exercise, problem, practice}

Interactivity type = "mixed"= {video, simulation}

Interactivity type = “<u>expositive</u>” = {hypertext, graphics, <u>audio</u>, essay}

Learning Resource Type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “simulation”= {behavior of some situation, behavior of process}

Learning resource type = “experiment”= {discover unknown, test hypothesis, establish some know truth}

Learning resource type=”<u>definition</u>”= {<u>explanation</u>, <u>give meaning</u>, objective}

Learning resource type=”<u>algorithm</u>”={<u>step for action</u>}

Learning resource type=”<u>example</u>”= {case study, <u>show how to act</u>}

Learning resource type=”index”= {glossary, reference, list of content}

<u>Mapping 3.</u>

Sensing = {realistic, reality, <u>sensible</u>, <u>fact</u>, certainly, concrete, <u>careful detail</u>, <u>existing way</u>}

Map to:

Semantic Density

Semantic density = “very low”= {message, text}

Semantic density =”low”= {definition, image}

Semantic density = “<u>medium</u>”={<u>audio</u>}

Semantic density = “<u>high</u>”= {<u>video</u>, exercise}

Semantic density = “<u>very high</u>”= {<u>simulation</u>, experiment}

Learning Resource Type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “<u>simulation</u>”= {behavior of some situation, <u>visual training</u>}

Learning resource type = “<u>experiment</u>”= {<u>discover unknown</u>, test hypothesis, establish some know truth}

Learning resource type=”definition”= {explanation, give meaning, objective}

Learning resource type=”algorithm”={step for action}

Learning resource type=”example”= {case study, show how to act}

Learning resource type=”index”= {glossary, reference, list of content}

Mapping 4.

Intuitive= {innovation, new idea, image, theory, concept, not careful about detail, insight, abstract material}

Map to:

Semantic Density

Semantic density = “very low”= {message, text}

Semantic density =”low”= {definition, image}

Semantic density = “medium”={audio}

Semantic density = “high”= {video, exercise}

Semantic density = “very high”= {simulation, experiment}

Learning Resource Type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “simulation”= {behavior of some situation, behavior of process}

Learning resource type = “experiment”= {discover unknown, test hypothesis, establish some know truth}

Learning resource type=”definition”= {explanation, give meaning, objective}

Learning resource type=”algorithm”={step for action}

Learning resource type=”example”= {case study, show how to act}

Learning resource type=”index”= {glossary, reference, list of content}

Mapping 5

Visual= {picture, diagram, graph, map, remember by looking, watch}

Map to:

Format

Format = “video” = {I see, moving eye picture, a recording of both the visual and audible components}

Format = “Image” = {two-dimension figure, map, graph, pie chart, abstract painting, computer graphic, drawing, painting, photograph, visual media, picture, idea }

Format = “text” = {set of writing, message}

Format =”audio” = {hear, listen, sound}

Format = “animation” = {motion picture, the act of animating, spirit, liveliness, airiness}

Learning resource type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “simulation”= {behavior of some situation, visual training}

Learning resource type = “experiment”= {discover unknown, test hypothesis, establish some know truth}

Learning resource type=”definition”= {explanation, give meaning, objective}

Learning resource type=”algorithm”={step for action}

Learning resource type=”example”= {case study, show how to act}

Learning resource type=”index”= {glossary, reference, list of content}

Interactivity level

Interactivity level = “very low”= {text, message}

Interactivity level =”low”= {audio, sound}

Interactivity level = “medium”={image, hypertext, online multiple choice}

Interactivity level = “high”= {video, simulation}

Interactivity level = “very high”= {animation, motion picture, 3D simulation}

Mapping 6

Verbal= {Spoken word, Explaining, verbal information, hear, written direction, written text, information}

Map to:

Format

Format = "video" = {I see, moving eye picture, a recording of both the visual and audible components}

Format = "Image" = {two-dimension figure, map, graph, pie chart, abstract painting, computer graphic, drawing, painting, photograph, visual media, picture, idea }

Format = "text" = {set of writing, message}

Format ="audio" = {hear, listen, sound}

Format = "animation" = {motion picture, the act of animating, spirit, liveliness, airiness}

Learner resource type

Learning resource type = "exercise"= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = "simulation"= {behavior of some situation, visual training}

Learning resource type = "experiment"= {discover unknown, test hypothesis, establish some know truth}

Learning resource type="definition"= {explanation, give meaning, objective}

Learning resource type="algorithm"={step for action}

Learning resource type="example"= {case study, show how to act}

Learning resource type="index"= {glossary, reference, list of content}

Learning resource type="slide"= {sequential step}

Interactivity level = "very low"= {text, message}

Interactivity level ="low"= {audio, sound}

Interactivity level = "medium"={image, hypertext, online multiple choice}

Interactivity level = “high”= {video, simulation}

Interactivity level = “very high”= {animation, motion picture, 3D simulation}

Mapping 7

Sequential= (understand detail, focus on detail, one time at time, think of incident, step solution}

Map to:

Format

Format = “video” = {I see, moving eye picture, a recording of both the visual and audible components}

Format = “Image” = {two-dimension figure, map, graph, pie chart, abstract painting, computer graphic, drawing, painting, photograph, visual media, picture, idea }

Format = “text” = {set of writing, message}

Format =”audio” = {hear, listen, sound}

Format = “animation” = {motion picture, the act of animating, spirit, liveliness, airiness}

Learning resource type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “simulation”= {behavior of some situation, visual training}

Learning resource type = “experiment”= {discover unknown, test hypothesis, establish some know truth}

Learning resource type=”definition”= {explanation, give meaning, objective}

Learning resource type=”algorithm”= {step for action}

Learning resource type=”example”= {case study, show how to act}

Learning resource type=”index”= {glossary, reference, list of content}

Learning resource type=”slide”= {sequential step}

Mapping 8

Global= (overall picture, global, relation, connection, work on different part}

Map to:

Format

Format = “video” = {I see, moving eye picture, a recording of both the visual and audible components}

Format = “Image” = {two-dimension figure, map, graph, pie chart, abstract painting, computer graphic, drawing, painting, photograph, visual media, picture, idea }

Format = “text” = {set of writing, message}

Format =”audio” = {hear, listen, sound}

Format = “animation” = {motion picture, the act of animating, spirit, liveliness, airiness}

Learning resource type

Learning resource type = “exercise”= {planned sequence of actions, assignment, worksheet, tutorial}

Learning resource type = “simulation”= {behavior of some situation, visual training}

Learning resource type = “experiment”= {discover unknown, test hypothesis, establish some know truth}

Learning resource type=”definition”= {explanation, give meaning, objective}

Learning resource type=”algorithm”={step for action}

Learning resource type=”example”= {case study, show how to act}

Learning resource type=”index”= {glossary, reference, list of content}

Learning resource type=”slide”= {sequential step}

Next, the learner style set (LSS) will be considered with mapping rules for creating the learner preference set (LPS). The definition of LPS is shown in definition 3.3.

Definition 3.3: *Learner Preference Set LPS is the set of learning object features which learner prefer to learn and its preferred weight.*

$LPS = \{(\{PF_i\}, Pw_i) | PF_i \in F_i, Fw_i \in \{0, 0.5, 1\}\}$

PF is preference feature and denoted by PF= {A, R, S, I, U, B, Q, G},

Fw is feature weight and i is number of feature.

From the rules are presented as above, we can convert the Learner Style Set of learner L1 (LSS_{L1}) to Learner Preference Set of learner L1 (LPS_{L1}) as follows.

LSS_{L1}= {(A,1), (R,0), (S,0.5), (I,0.5), (U,1,), (B, 0), (Q,0), (G,1)}

LPS_{L1} = {({*exercise, simulations, experiment, active, mixed*},*1), (*{*simulation, experiment, 8, 9*}*, 0.5),* {*definition, exercise, 5, 6, 7*}*, 0.5), (*{*video, image, animation, simulation, 3, 4, 5*}*,1), (*{*image, index*}*,1)*}

Both of LSS and LPS will be used as input value in the recommendation algorithm in the next chapter.

3.4 Summary

In this chapter, the model of learner and learning object is designed. Both of learning object model and learner model are the important parts of learning object recommendation. This analysis and design of them will help us illustrate the well-known fact about learning object characteristic that learner prefers during the learning process. Learning Object Set (LOS) will be collected in learning object database (Learning object repository). Both of learner style set (LSS) and learner preference set (LPS) will be stored in learner model database and used as input value of recommendation algorithms that are proposed in Chapter IV.

CHAPTER IV

LEARNING OBJECT RECOMMENDATION MODEL

In this chapter, general requirements of learning object recommendation model and the detail inside of system designing for supporting the learning object recommendation process were addressed. The first section presents system architecture that designed based on multi agent model. In the thesis, the multi agent-based model for describing the overall architecture and how the recommendation algorithm work in the system that designed with agents were designed. Then, Section 4.2 presents concept map analysis and selection for filtering the suitable concepts to learners. Next, the proposed recommendation algorithms are presented in Section 4.3 and the chapter ends with a summary in Section 4.4.

4.1 The Learning Object Recommendation Model

General requirements of an agent-based system for learning object recommendation are listed as follows:

- This is a system where learner will be connected in order to get access to learning object. At this point, learning object is considered to be created in the form of full concept in courses, but it can also be viewed as learning objects that can be synthesized into full courses according to learner needs.
- This is a system where content providers will be connected in order to publish learning object. That is, to describe the e-learning course they have created and publish offers to the rest of the users of the system.
- This is a system that can provide learner modeling and content modeling services. To be more specific, in this architecture the learner will be provided with a

recommendation service, based on the parameters of the learner model and the description of the learning object characteristics.

- This is a system that is able to analyze learner and provide the matching services to the learner in process of learning object recommendation.

In this thesis we design all components that required in multi agent- based learning object recommendation system and implement some of important functions of each agent to show recommendation algorithm results. For implementing the real system, this designed model can be used to guild the developers.

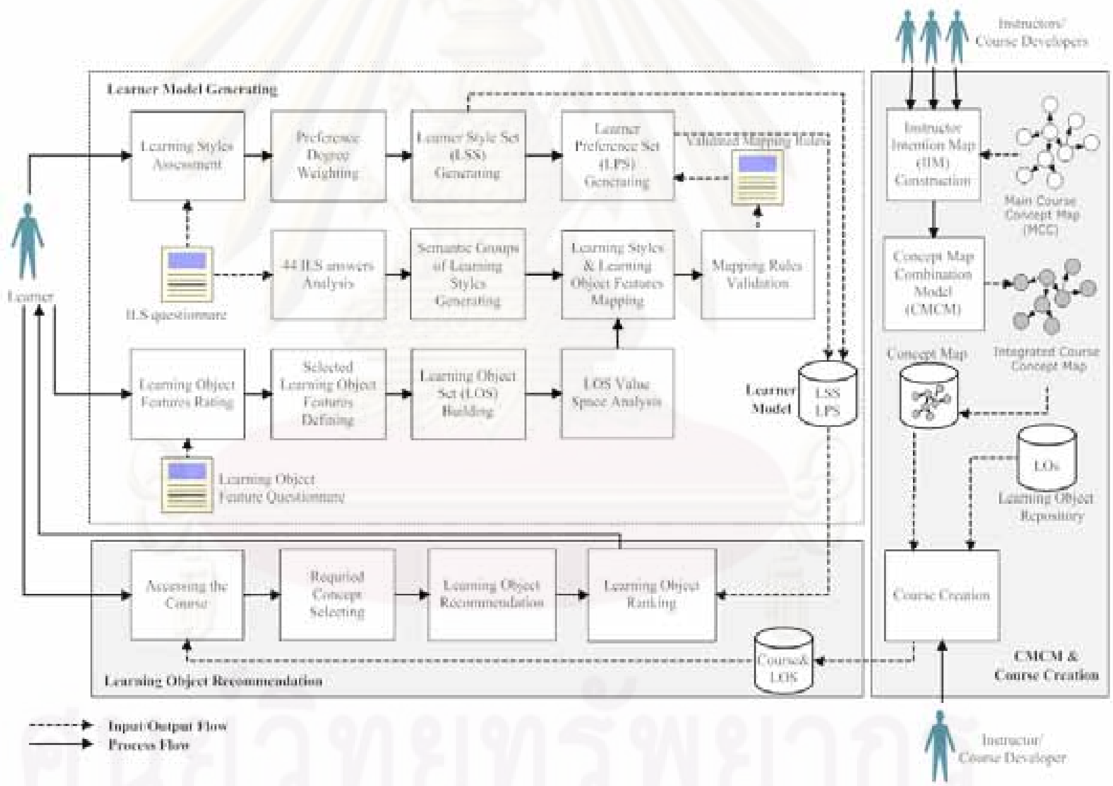


Figure 4.1: Overall process in learning object recommendation model.

In this chapter two sub-models (two gray boxes in Figure 4.1) for supporting learning object recommendation are explained. Firstly, the course creation that provides service of concept map combination. The activity diagram of this process is shown in Figure

4.2 and the step descriptions are shown in Table 4.1. The detail of concept map combination process is presented in Section 4.2.

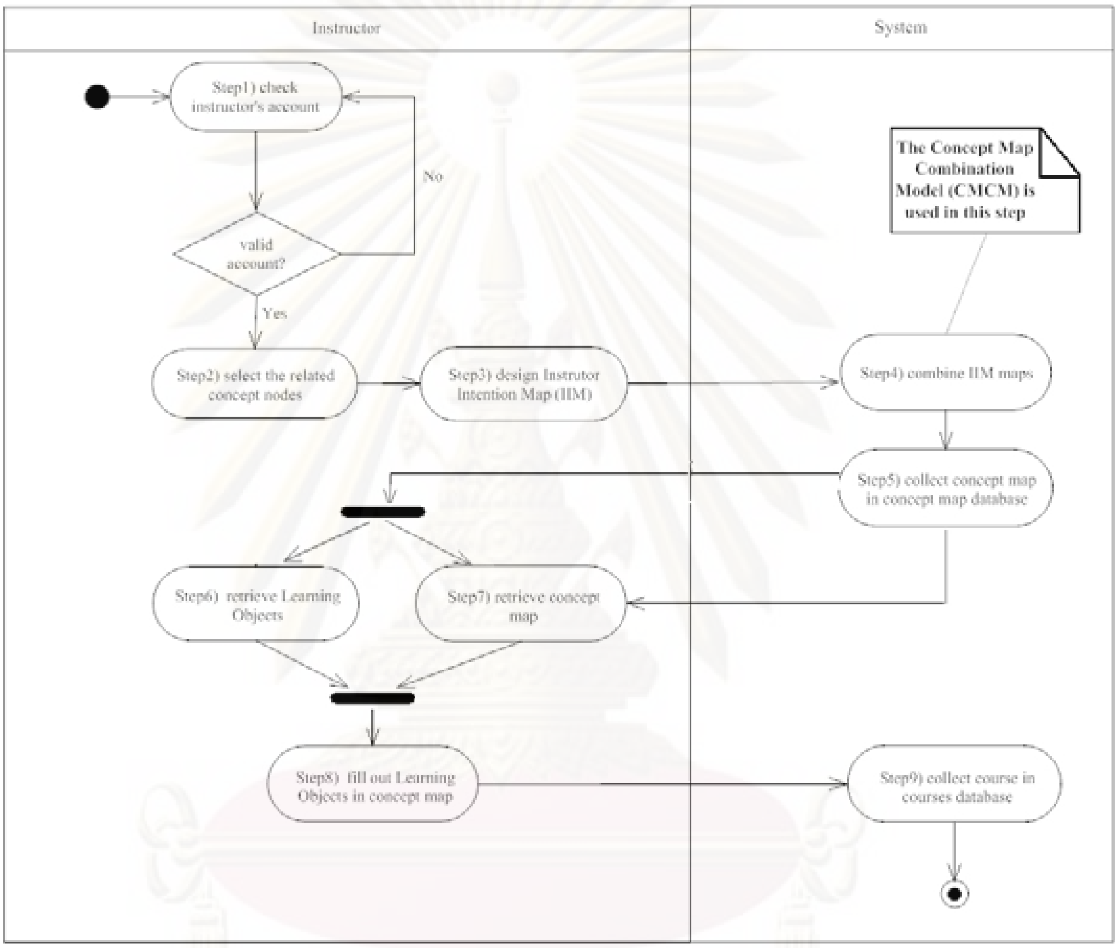


Figure 4.2. The activity diagram of CMCM and course creation

Table 4.1: Step description of CMCM and course creation.

Step	Description
Step 1:	Instructor's account is checked by the legal instructor's account database.
Step 2:	Instructors select the related concept nodes those corresponding to the learning goal.
Step 3:	Instructors design their own concept map by using their background knowledge and experience.
Step 4:	The concept map combination model (CMCM) is used to combine various designs of concept map and describe it into the ontology format.
Step 5:	The concept map ontology from step 4 is collected in concept map ontology database.

Table 4.1: Step description of CMCM and course creation. (cont.)

Step	Description
Step 6-8:	Instructor creates their course by using concept map and learning object from learning object repository.
Step 9:	The course is collected in courses database.

Secondly, a learning object recommendation model that provides service of personalized learning object selection for learners. There are four intelligent agents in this model. Four intelligent agents are learner interface agent, feedback agent, learner model agent and learning object recommendation agent. Figure 4.3 presents the activity diagram for creating personalized learning object recommendations.

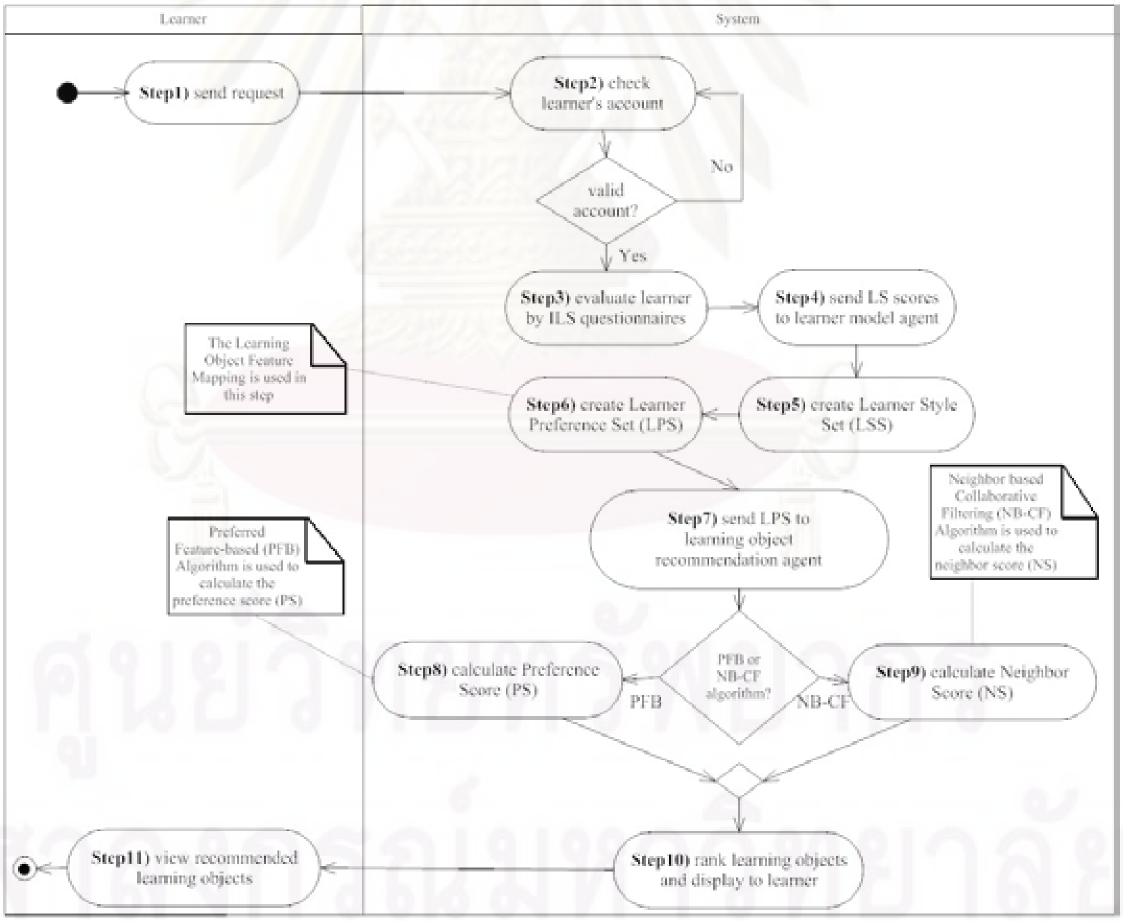


Figure 4.3: The activity diagram of learning object recommendation.

Table 4.2: Step description of learning object recommendation.

Step	Description
Step 1:	Learner logs in the system through the learner interface agent.
Step 2:	The learner interface agent checks learner's account in the account database. If the learner is the new learner, the system provides the questionnaire (ILS) for learner assessment process.
Step 3:	The learner is evaluated by ILS questionnaire.
Step 4:	The ILS score of learner will be sent to learner model agent.
Step 5:	The Learner Style Set (LSS) is created from ILS score.
Step 6:	Learner model agent processes the LSS for transforming into Learner Preference Set (LPS) and collects in learner model database.
Step 7	Learner model agent provides the LPS to learning object recommendation agent.
Step 8:	The learning object recommendation agent evaluates the learner according to the LPS and the information about the feature of learning object for each learner by using the recommendation algorithm. Then, compute the preference score (PS) of each candidate learning object.
Step 9:	Neighbor –based recommendation is used to calculate the neighbor score (NS).
Step 10:	The learning object recommendation agent ranks the candidate learning object and sends the list of ranking to learner interface agent. The ranking of learning object will be shown to learner.
Step 11:	The learner views recommended learning objects.

A generic architecture for agent-based course brokering is defined to represent the main roles participating in the recommendation process. The main agents participating in sequential diagram (Figure 4.4) are the following:

- **Learner Interface Agent:** The learner interface agent detects any user interaction with the learner interface and records the results, if any, of these interactions. When the learner first logs in, the learner interface agent reads the given username and password and passes this information to the learner model agent, which then accesses the learner's profile. When the learner completes a questionnaire, or other fill out feedback, the interface agent passes the responses and results to the leaner model agent. This information is used to build a model of learner's abilities and how they view their own experience to date.

- **Learner Model Agent:** The learner model agent is responsible for maintaining, updating and analyzing the learner profile. The learner model agent uses a learning object selection rules (we described in chapter III) to create the learner preference set (LPS). Then, the negotiation between learner model agent and learning object recommendation agent will happen when learner sends the recommendation request via learner interface agent.

- **Learning Object Recommendation Agent:** The learning object recommendation agent needs learner's information from learner model agent to compute the preference score of each learning object (the detail of each algorithm will be presented in Section 4.3). Moreover, it provides the ranking process and recommends the most compatible learning object to the learner via learner interface agent.

- **Feedback Agent:**
  For adaptation of the system, the feedback agent will be designed for learner feedback to the recommended learning objects. If learner does not satisfy to them, the learning object selection rule or learner model will be updated and the all process of recommendation will be restarted.

Each agent has different functions, learner interface agent aims at providing learner's learning style assessment process and flexible learning interface for learners to interact with feedback agent and learning object recommendation agent. The main function of feedback agent aims to collect learner explicit feedback information for learner interface agent and store it in user model database for personalized learning object operations. The learning objects recommendation agent provides recommendation algorithm to find the most compatible learning object to learner according to learner model database, learning object characteristics and the selected concept map.

From functions of each agent, we show cooperation among agents as the sequence diagram in Figure 4.4.

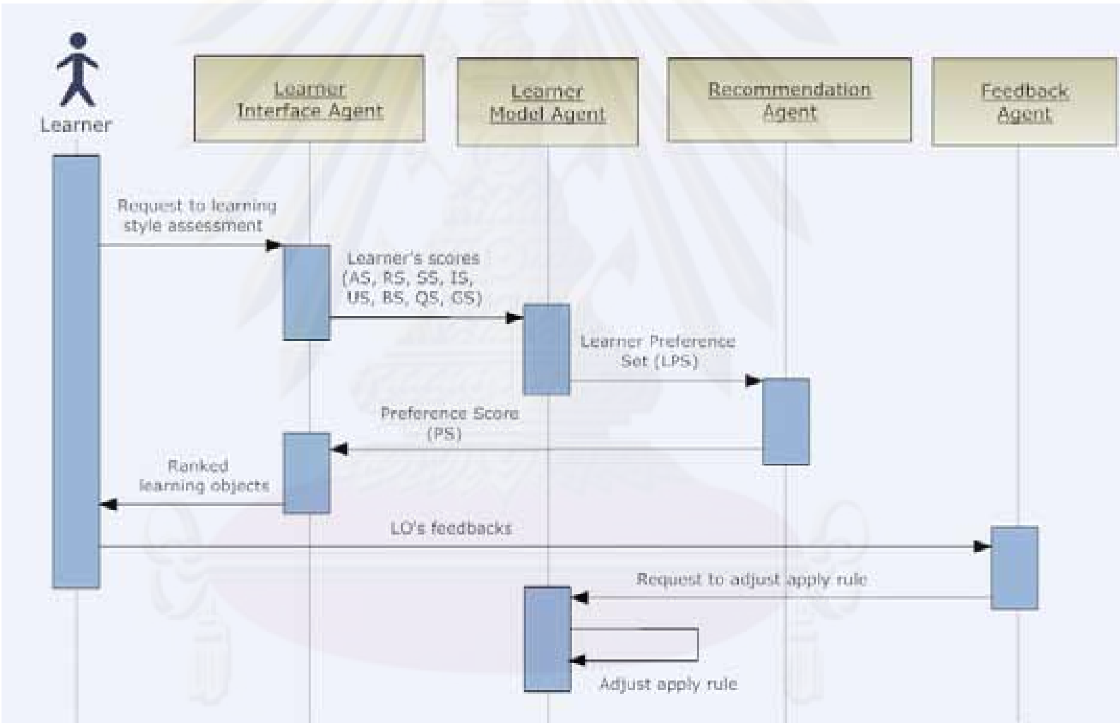


Figure 4.4: Sequence of personalized learning object selection.

For demonstrating the result of learning object recommendation algorithms in this thesis, we implement some functions of recommendation and learner model agents. The class diagram in Figure 4.5 shows the detail of each class. The detail of function implementation will be explained in section 4.3.

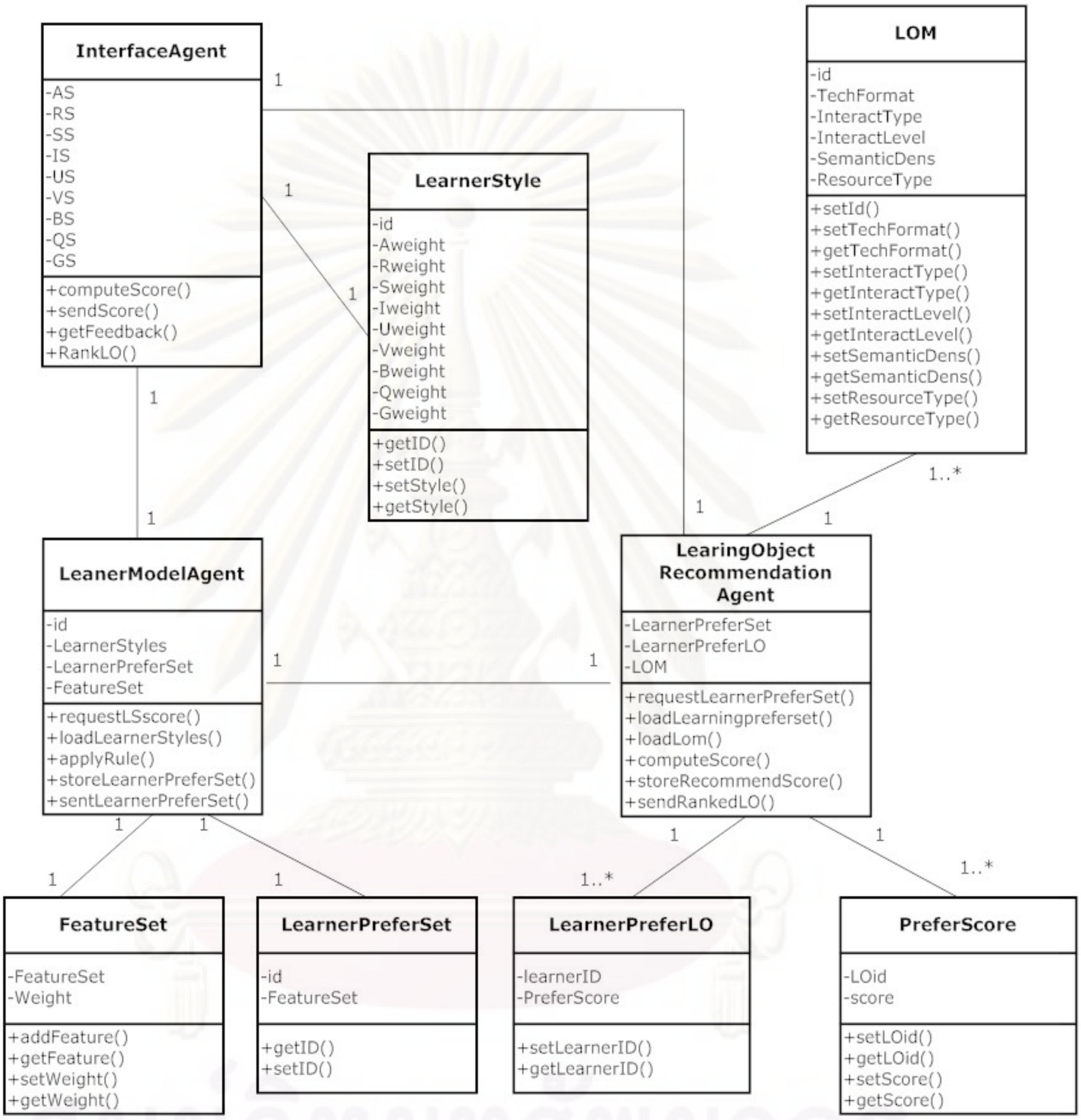


Figure 4.5: Class diagram for describing agent's functions.

In conclusion, the multi agent-based learning object recommendation is strongly based on a continuous interaction among involved agents: such an activity is facilitated by the choice of XML for both representing agent ontologies and handling data exchange.

4.2 The Concept Map Analysis and Selection

The term "instructional strategy" is used to describe a process of choosing a delivery system, sequencing and grouping clusters of content, describing learning materials that will be combined in instruction, establishing the lesson structure, and selecting learning objects for delivering instruction.

Identifying a concept map and manageable groupings of contents is the first step to develop an instructional strategy. We are looking for an efficient mechanism that presents suitable concepts for containing learning objects to learners. The instructional goal is an important consideration when designing the strategy. So, we use the learning concept selection recommended by instructors who have experience and know their learner's background to design the most suitable course concept map to reach the instructional goal. In this work, we make the assumption that all of the instructors who know the profile or characteristics of their learners define an individual sequencing style based on experience, specialization and personal characteristics. Therefore, we would like to integrate the different styles of each instructor into the most suitable concept map of learning concept using the relationships between them.

4.2.1 Challenges and Benefits of Collaboration

Designing an online course is not as simple as putting the syllabus on the internet. In traditional classroom, the responsibility of instructors is to define the concepts and design paths of related concept for that course follow as the course syllabus by their own design.

In e-learning system, course concept design for Web-based education is one that entails combining a variety of instructional strategies into a unique environment (Ritter, 2004). From our knowledge, no previous studies have explored any roles of such factors in human judgments of concept importance. Most the problem of decision making for selecting the suitable concepts and paths are happened to new instructors who have no teaching's experience or instructors who want to develop their teaching strategies by sharing their knowledge with other experts. Another problem is also the question, how they do if some instructors want to add some new concepts to the course and try to share with others. Moreover, in the situation of multiple experts, who is the most reliable expert and we can trust? All problems mentioned earlier are the reason for this work. Since our concrete approach is an interdisciplinary professional community of experts, the assumption is that the instructors will be willing to share their designs.

In this work, the method called “Collaborative course concept designing”, is implemented based on knowledge sharing approach in terms of structure design. Each instructor (as expert) will design his/her structure, then the process start at the similarity computation among the instructors. The output from similarity computation is an input to the method to find the confidence of each instructor by the closeness index computing. The confident value will be used for judging process of concept path selection in terms of “weight”. The high confidence shows that this instructor has a high similarity design characteristic with other instructors. These weights will be combined with the concept important weight and affected to the consideration for making two decisions, keep or skip concept node, compared by the threshold value which is usually 0.5.

The result of the combination model is an integrated concept map. It provides the choice for instructors; totally trust, partial trust or not trust. In case of distrustful result, the instructor can ignore that concept map and will use the map which only designed by their own. The advantages of this work are explained as follows:

For instructors, collaboration provides access to alternative focused skill sets. They can ignore unnecessary course concepts and can expand the range of knowledge and expertise available for their course, their students and themselves. They view this expanded range of expertise and perspectives as invaluable to the development of the course. Additionally, sharing designs of course with these instructors leads to an enhanced sense of overall responsibility. This collaboration is so helpful because it allows the instructors to focus on areas of their expertise (content). The instructors have choices to share the designs if they completely trust that their partners will assume responsibility for their areas of expertise. The sense of trust is the keystone that can generate innovation and continual improvement. At the same time, collaboration does have challenges:

- Store the design in form of ontology and will be reused in another course.
- Solve the problem of multiple instructors' opinion in the same course.
- Share and exchange the experience among instructors.
- Make a guideline for a new instructor.

For learners, they need a process of concept filtering before using the learning object recommendation system. The concept map scopes only the concept which they need to learn. The representation in form of ontology make efficiency semantic search of learning object in the repository.

For instructional designers, the collaborative design from multiple instructors will return the feedback to instructional designers. For example, the concept which is never selected from every instructor will be considered to discard from the course map. These feedbacks improve the instructional strategy to make the most benefit to instructors and learners.

For learning material developers, the concept map represents overview of a specific course. The developers can know what the missing learning materials or learning object are? In case of developing new learning objects, they can use the concept map as learning object's categories for supporting the semantic searching process.

4.2.2 Defining Terminology

The terms concept map combination are explained as follows:

Concept: In this work, we define the concept as the knowledge for describing the subject domain.

Concept Node and link: An object in the concept map or concept sequence extension is called "concept node" and a line drawn between nodes is called "link". Link represents relationships between concept nodes.

Concept Map: A concept map is a directed acyclic graph (DAG) (McKay et al., 2004) that represents a set of sequencing rules determining the order of the concepts. They should be followed by a list of the behaviors that the instructor intends to design for their learner. Learning goals are considered the pattern of concept map based on the domain model and user model. In a DAG with collection concepts, deletion of a collection concept does not automatically result in destruction of links to that collection concept from leaf concepts or other collection concepts because a child can have multiple parents (but if the last parent is deleted, the last remaining link is destroyed). In this case we say that the concept map is a *Loose Hierarchy* (Andre et al, 2008). A concept map consists in grouping concepts into classes that materialize concepts of the domain knowledge under study. Individual concepts are discriminated according to their common properties.

Concept Bit Stream: The representation of existing nodes or concepts in bit stream; "1" is defined to existing nodes in the map and "0" is defined to non-existing node. E.g. [1, 0, 1, 1, 1, 1, 0]

Concept Ontology: Concept ontology is categorized by knowledge areas as specific course in curriculum. All the concept ontologies in a repository are collected from the process of instructional design. They are examined and formalized, and then classified according to the specific course.

For demonstrating the collaborative course concept designing in the thesis, we use an example of "Operating System" concept map designed by four instructors to demonstrate the mechanism of each process.

4.2.3 Concept Map Combination Model (CMCM)

Process 1: Share Learning Goal

Instructors learn the learning goal and review all concepts (nodes) in the main concept map of specific course. The first step is to generate the main course concept map of the specific course.

Generating of the Main Course Concept Map (MCC Map)

The course concept map is the domain model that represents all possible sequences of learning concept for a specific course (Shyu et al., 2003). The domain model stores the knowledge about the course preferences, instructor's characteristics and experiences.

The main concept map was implemented by using the Cmaptool (Novak et al., 2006). CmapTools is a suite of tools for generating and sharing concept maps in electronic form. CmapTools supports generating and modifying concept maps, as well as adding navigational links from concepts to other concept maps and multi-media material such as images, diagrams, and video clips, enabling the construction of rich knowledge models. The tools facilitate storage and access of concept maps on multiple servers, providing the network services required to support knowledge sharing across geographically-distant sites.

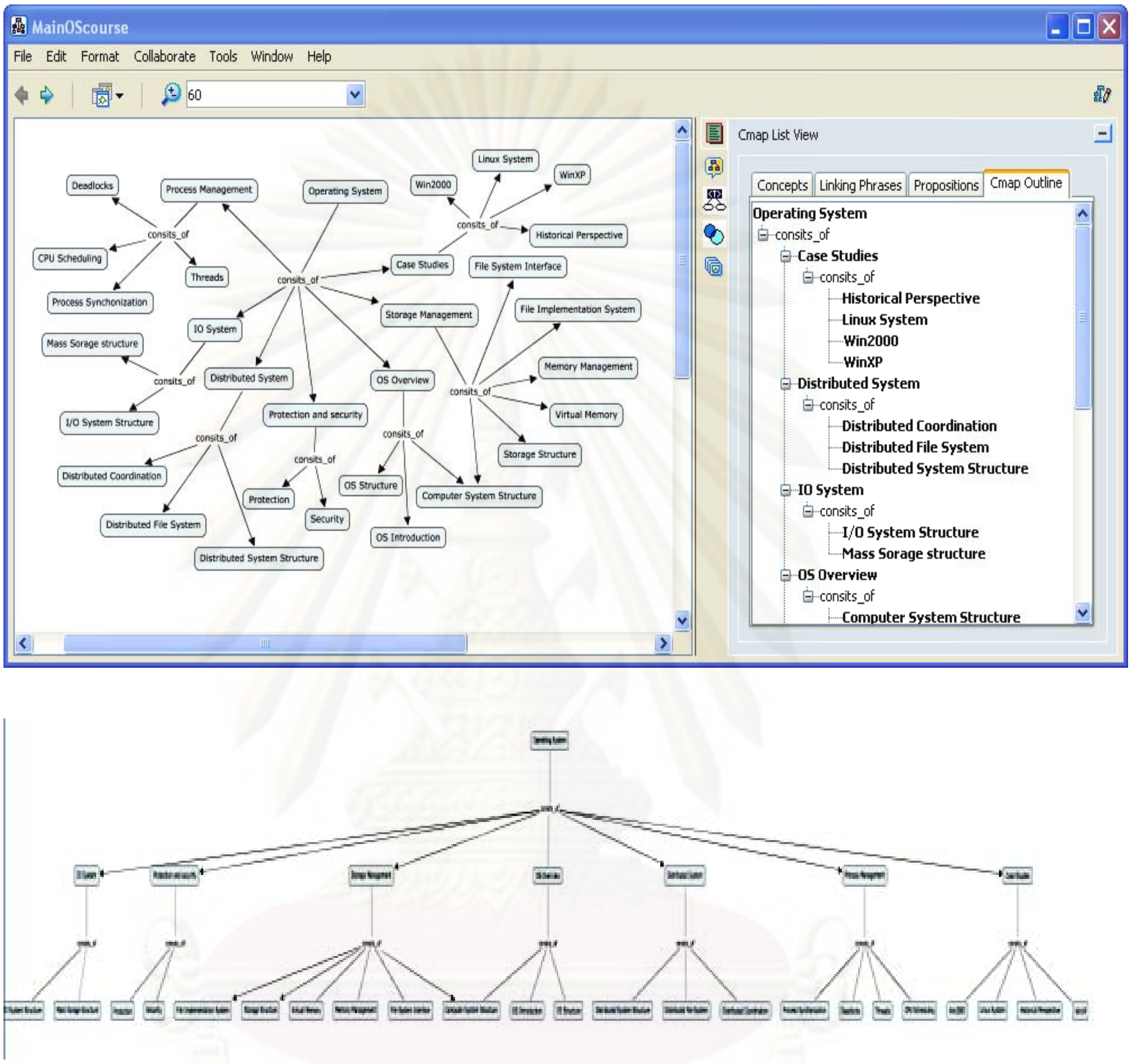


Figure 4.6: Main course concept map generated by instructional designers.

The structure of main concept map for each course is developed by instructional designer including the domain knowledge of learning objectives. The concept map can be viewed as graph which have only single root and each node can have a collection of concepts called “concepts set”. Every link in MCC map represents the consist-of relation, it means the parent concepts can have a set of child concepts. Figure 4.6 shows an example of MCC map of Operating System (OS) course and views of MCC map in structure of hierarchical tree.

Concept Extraction and Label Defining

Next, we present the computation example by using the fragmentation of Operating System MCC map that is illustrated in Figure 4.7. The top right window shows the concepts in hierarchical view and the bottom right window represents the number of *Links In* and *Link Out* of each concept.

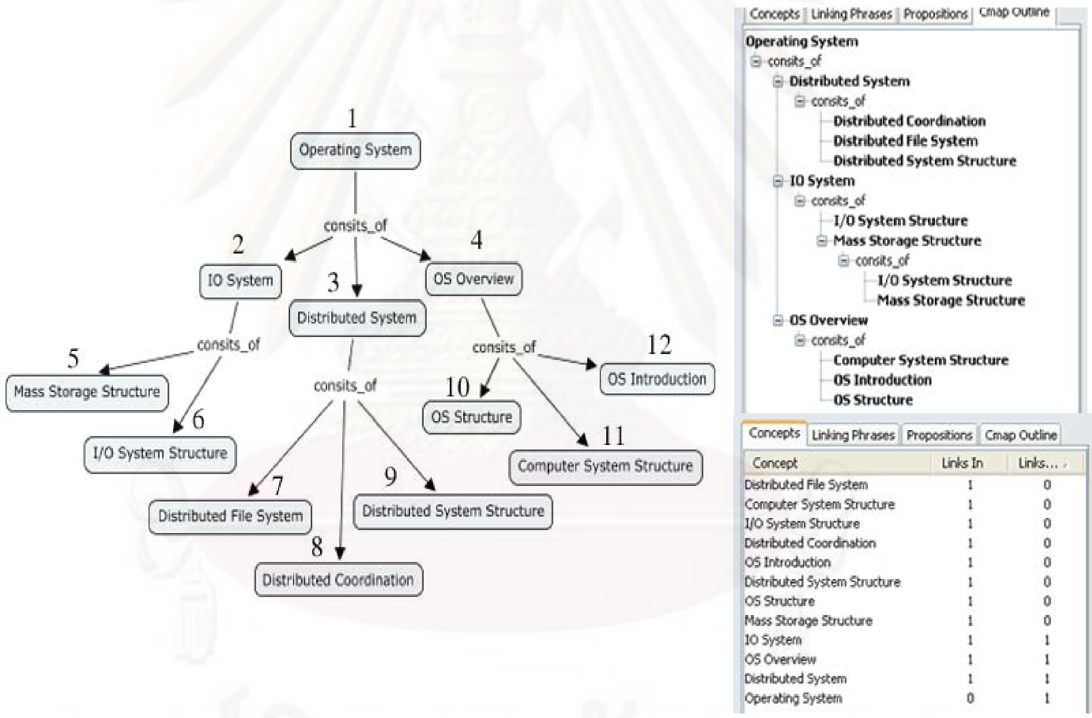


Figure 4.7: The fragmentation of operating system MCC Map and its properties.

The concept nodes in MCC map is labeled with ordering number, in Figure 4.7 MCC map consist of 12 concept nodes with their labels.

Table4.3: The information about MCC map.

Parent Concept	Child Concepts Collection (label)	Represented Bit	Concept Level	MCC bit stream
{}	{1}	1	-	1,1,1,1,1,1,1,1,1,1,1,1
1	{2, 3, 4}	1, 1, 1	1	
2	{5, 6}	1, 1	2	
3	{7, 8, 9}	1, 1, 1	2	
4	{10,11,12}	1, 1, 1	2	
5	∅	-	3	
6	∅	-	3	
7	∅	-	3	
8	∅	-	3	
9	∅	-	3	
10	∅	-	3	
11	∅	-	3	
12	∅	-	3	

The first process of combination algorithm is to extract the child concepts collection from their parents from MCC Map and represent the existing value by concept bit “0” or “1”. The concept level describes the depth of parent concepts in MCC map and they have an implicit weight for each concept up to the depth of them. Table 4.3 shows the information about MCC map.

Process2: Multiple Perspective& Design

Instructors design their concept map called “Instructor Intention Map” (IIM) based on main concept map that describes relationship between concepts by using their background knowledge and experience. Note that, in this process, they have to consider the suitability for their learner preferences and characteristics.

Generating of the Instructor Intention Map (IIM)

For collaborative design, instructors in the same course design concept map of a specific course by pruning the MCC map. This process provides the instructor with considering the MCC Map and gives an agreement for each concept node. The concept map is a loose hierarchy, so if the parent does not exist, all child node in their collection will be deleted too. The concept map preferred by them called "Instructor Intention Map" shows the preference concepts in each instructor's opinion. The set of concepts will be presented in format of bit vector. Figure 4.8 presents the IIM map defined by four instructors. Based upon the example in Figure 4.8, an algorithm to combine a different concept map into a single pattern was presented. In this example, we demonstrate the process of concept map combination from designs of four instructors. The algorithm steps are illustrated by the example described in detail below.

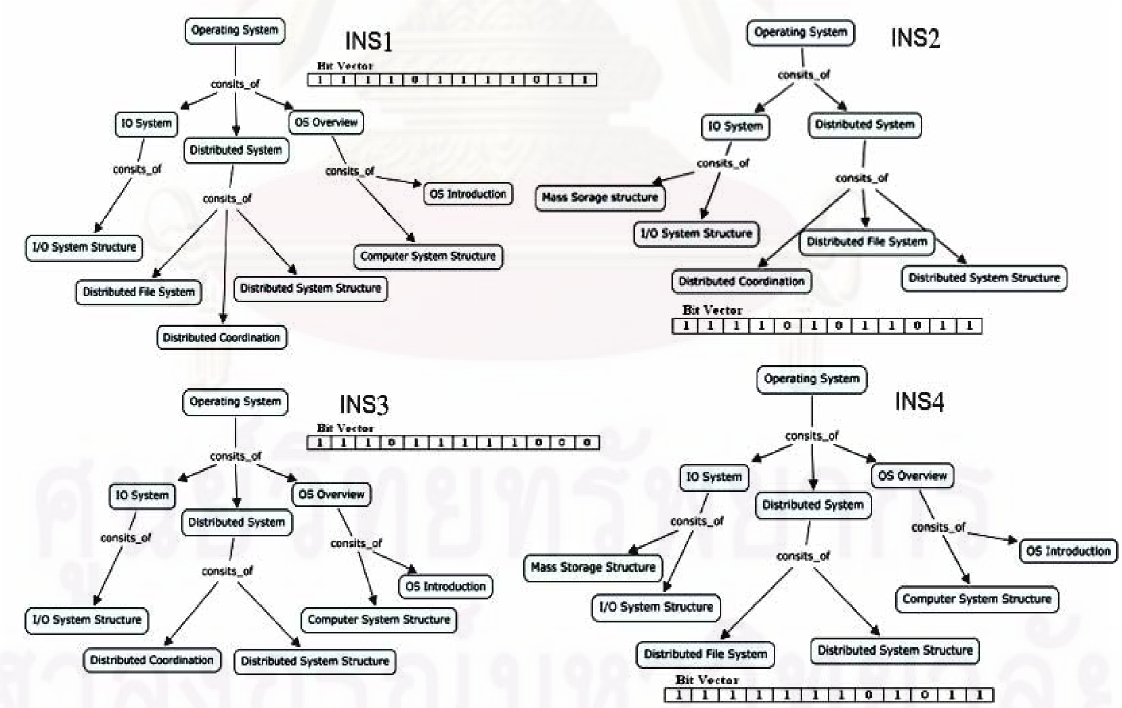


Figure 4.8: The instructor intention maps designed by four instructors.

Process 3: Instructors Negotiation

Combination agent views all designs of instructors and creates the message for combination process by instructor requirement. The conceptual follow this process is shown below.

For every Instructor Intention Maps,

Step1: Assign the existing value of nodes by using the set of concept nodes. This is the matching process between MCC nodes set and IIM nodes set with two choices: *agree* or *disagree*. Two types of choice affect to concept node existing in format of bit stream. The groups of selected and unselected concept nodes will be kept in the collection of concepts called Instructor Node Set (INS).

Step2: Represent INS with bit stream which have selecting of only one from two values: 0 or 1, after this process the system return the INS in bit stream format which describe about the agreement of instructors to the MCC map.

The result from step 2 will be used as an input in the next process, course concept combination which consists of the computation of closeness index and confident value.

Process 4: Course Concept Combination

Closeness index calculation

Goldsmith's method (Goldsmith et al., 1990) is used to calculate the closeness index and ordering processes. This method is repeated for every pair of instructors. The closeness index calculation algorithm is explained in Figure 4.9.

Algorithm 2 : Closeness Index Calculation

Input: Set of node $INS_i = \{C_1, C_2, ..., C_n\}$ $, C_n$ represent the concept nodes in the IIM map.

Output: The closeness index of instructors in the specific course.

Methods:

1. For every pair of set of node INS_i of instructors

    1.1 For every node belonging to INS_i

    Consider the concept nodes in INS from instructor number j (INS_j) and instructor number k (INS_k).

    1.2 For every concept nodes of each instructor insert the related node into the related set.

    1.3 Calculate the Intersection Set (IS_{jk}), and Union Set (US_{jk}).

    1.4 Calculate the closeness coefficient (CC_{jk}) with

$$CC_{jk} = \frac{|IS_{jk}|}{|US_{jk}|} \tag{4}$$

    where $|US_{jk}|$ is the number of members in the Union Set, $|IS_{jk}|$ is the number of members in the Intersection Set

    1.5 Calculate the closeness index between instructor ID_j and ID_k with

$$C\left(ID_j, ID_k\right) = \frac{1}{|L|} \sum CC_{jk} \tag{5}$$

    where $|L|$ is the number of items in INS .

2. Return closeness index $C\left(ID_j, ID_k\right)$

Figure 4.9: Closeness index calculation.

For example, the comparison among four instructors is derived from example in Figure 4.8. The closeness coefficient (CC_{jk}) results are shown in Table 4.4.

Table 4.4: Comparison of results of closeness index calculation among four instructors.

ID1 vs ID2

Node	ID1	ID2	$US_{1,2}$	$IS_{1,2}$	$CC_{1,2}$
1	{2,3,4}	{2,3}	{2,3,4}	{2,3,4}	1.00
2	{1,6}	{1,5,6}	{1,5,6}	{1,6}	0.67
3	{1,7,8,9}	{1,7,8,9}	{1,7,8,9}	{1,7,8,9}	1.00
4	{1,11,12}	∅	{1,11,12}	∅	0.00
5	∅	{2}	{2}	∅	0.00
6	{2}	{2}	{2}	{2}	1.00
7	{3}	{3}	{3}	{3}	1.00
8	{3}	{3}	{3}	{3}	1.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	{4}	∅	{4}	∅	0.00
12	{4}	∅	{4}	∅	0.00
$\sum CC_{12}$					**7.67**
C(ID1,ID2)					**0.6389**

ID1 vs ID3

Node	ID1	ID3	$US_{1,3}$	$IS_{1,3}$	$CC_{1,3}$
1	{2,3,4}	{2,3,4}	{2,3,4}	{2,3,4}	1.00
2	{1,6}	{1,6}	{1,5,6}	{1,6}	0.67
3	{1,7,8,9}	{1,8,9}	{1,7,8,9}	{1,8,9}	0.75
4	{1,11,12}	{1,11,12}	{1,11,12}	{1,11,12}	1.00
5	∅	∅	∅	∅	1.00
6	{2}	{2}	{2}	{2}	1.00
7	{3}	∅	{3}	∅	0.00
8	{3}	{3}	{3}	{3}	1.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	{4}	{4}	{4}	{4}	1.00
12	{4}	{4}	{4}	{4}	1.00
$\sum CC_{13}$					**10.42**
C(ID1,ID3)					**0.8681**

ID1 vs ID4

Node	ID1	ID4	$US_{1,4}$	$IS_{1,4}$	$CC_{1,4}$
1	{2,3,4}	{2,3,4}	{2,3,4}	{2,3,4}	1.00
2	{1,6}	{1,5,6}	{1,5,6}	{1,5,6}	1.00
3	{1,7,8,9}	{1,7,9}	{1,7,8,9}	{1,7,9}	0.75
4	{1,11,12}	{1,11,12}	{1,11,12}	{1,11,12}	1.00
5	∅	{2}	{2}	∅	0.00
6	{2}	{2}	{2}	{2}	1.00
7	{3}	{3}	{3}	{3}	1.00
8	{3}	∅	{3}	∅	0.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	{4}	{4}	{4}	{4}	1.00
12	{4}	{4}	{4}	{4}	1.00
$\sum CC_{14}$					**9.75**
C(ID1,ID4)					**0.8125**

ID2 vs ID3

Node	ID2	ID3	$US_{2,3}$	$IS_{2,3}$	$CC_{2,3}$
1	{2,3}	{2,3,4}	{2,3,4}	{2,3}	0.67
2	{1,5,6}	{1,6}	{1,5,6}	{1,6}	0.67
3	{1,7,8,9}	{1,8,9}	{1,7,8,9}	{1,8,9}	0.75
4	∅	{1,11,12}	{1,11,12}	∅	0.00
5	{2}	∅	{2}	∅	0.00
6	{2}	{2}	{2}	{2}	1.00
7	{3}	∅	{3}	∅	0.00
8	{3}	{3}	{3}	{3}	1.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	∅	{4}	{4}	∅	0.00
12	∅	{4}	{4}	∅	0.00
$\sum CC_{23}$					**6.08**
C(ID2,ID3)					**0.5069**

ID2 vs ID4

Node	ID2	ID4	$US_{2,4}$	$IS_{2,4}$	$CC_{2,4}$
1	{2,3}	{2,3,4}	{2,3,4}	{2,3}	0.67
2	{1,5,6}	{1,5,6}	{1,5,6}	{1,5,6}	1.00
3	{1,7,8,9}	{1,7,9}	{1,7,8,9}	{1,7,9}	0.75
4	∅	{1,11,12}	{1,11,12}	∅	0.00
5	{2}	{2}	{2}	{2}	1.00
6	{2}	{2}	{2}	{2}	1.00
7	{3}	{3}	{3}	{3}	1.00
8	{3}	∅	{3}	∅	0.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	∅	{4}	{4}	∅	0.00
12	∅	{4}	{4}	∅	0.00
$\sum CC_{24}$					**7.42**
C(ID2,ID4)					**0.6181**

ID3 vs ID4

Node	ID3	ID4	$US_{3,4}$	$IS_{3,4}$	$CC_{3,4}$
1	{2,3,4}	{2,3,4}	{2,3,4}	{2,3,4}	1.00
2	{1,6}	{1,5,6}	{1,5,6}	{1,6}	0.67
3	{1,8,9}	{1,7,9}	{1,7,8,9}	{1,9}	0.50
4	{1,11,12}	{1,11,12}	{1,11,12}	{1,11,12}	1.00
5	∅	{2}	{2}	∅	0.00
6	{2}	{2}	{2}	{2}	1.00
7	∅	{3}	{3}	∅	0.00
8	{3}	∅	{3}	∅	0.00
9	{3}	{3}	{3}	{3}	1.00
10	∅	∅	∅	∅	1.00
11	{4}	{4}	{4}	{4}	1.00
12	{4}	{4}	{4}	{4}	1.00
$\sum CC_{34}$					**8.17**
C(ID3,I4)					**0.6806**

Confidence Value Calculation

We compute the confidence values for each instructor by using the closeness index from the steps above with instructors' confidence calculation algorithm in Figure 4.10. The results are shown in Table 4.5.

Algorithm 2: Instructor's Confidence Calculation

Input: Instructors' Closeness Index Values.

Output: Confidence values of each instructor.

Method:

For every intructor ID_j

1. Compute the total closeness index values of each instructors by using the closeness index from algorithm 3 with

$$Total(ID_j) = \sum_{k=1}^{m} C\left(ID_j, ID_k\right) \tag{6}$$

2. Compute the confidence values with

$$Conf\,(ID_j) = \frac{Total(ID_j)}{\sum_{i=1}^{m} Total(ID_i)} \tag{7}$$

,where m is the number of instructors

Figure 4.10: Instructors Confidence Calculation.

Table 4.5: Results of confident value calculation.

ID#	ID1	ID 2	ID 3	ID 4
ID 1		0.6389	0.8681	0.8125
ID 2	0.6389		0.5069	0.6181
ID 3	0.8681	0.5069		0.6806
ID 4	0.8125	0.6181	0.6806	
Total	**2.3194**	**1.7639**	**2.0556**	**2.1111**
Confidence	**0.2811**	**0.2138**	**0.2492**	**0.2559**

Integrating Concept Node Weight to IIM maps

Defining the weight of concept nodes in IIM map, two factors were considered: Instructor's confidence weight and concept important weight. The steps in the integrated concept nodes weights mechanism are explained as follows.

Step 1: Instructors Confidence Weighting

In weighting process of concept nodes, the bit stream represents the existing concept nodes with bit "1". On the other hand, bit "0" represents the nonexistent concept nodes. We define the weighted value as 1 for existing nodes and 0 for nonexistent nodes. We compute the instructor confidence weight by using equation (8). In equation, the existing nodes are represented as "Existing Values" (EV) and the total node weight is represented as "Instructors Confident Weight" (ICW). The important factor for computation is the instructor's confidence (Conf(ID)). The results are shown in Table 4.6.

$$ICW(C) = \sum_{1}^{m} (EV * Conf(ID_j)) \qquad (8)$$

Table 4.6: The weighted values of nodes.

Bit Vector

		1	2	3	4	5	6	7	8	9	10	11	12
INS1		1	1	1	1	0	1	1	1	1	0	1	1
INS2		1	1	1	0	1	1	1	1	1	0	0	0
INS3		1	1	1	1	0	1	0	1	1	0	1	1
INS4		1	1	1	1	1	1	1	0	1	0	1	1
Weighted Node													
ID1	0.2811	0.2811	0.2811	0.2811	0.2811	0.0000	0.2811	0.2811	0.2811	0.2811	0.0000	0.2811	0.2811
ID2	0.2138	0.2138	0.2138	0.2138	0.0000	0.2138	0.2138	0.2138	0.2138	0.2138	0.0000	0.0000	0.0000
ID3	0.2492	0.2492	0.2492	0.2492	0.2492	0.0000	0.2492	0.0000	0.2492	0.2492	0.0000	0.2492	0.2492
ID4	0.2559	0.2559	0.2559	0.2559	0.2559	0.2559	0.2559	0.2559	0.0000	0.2559	0.0000	0.2559	0.2559
ICW		**1.0000**	**1.0000**	**1.0000**	**0.7862**	**0.4697**	**1.0000**	**0.7508**	**0.7441**	**1.0000**	**0.0000**	**0.7862**	**0.7862**

Step 2: Integrating Important Value to Concept Nodes

Our model was considered several structural influences on concept keeping including: Firstly, voting of instructors, measured in terms existing value which calculated from instructor's confidence that described above. Secondly, the Connectivity Root Distance (CRD) weight (Leake et al., 2004) is used to assign the helpfulness value for concept nodes in MCC Map. The model parameters α, β, and δ adjust the effect of the number of incoming connections (i), the number of outgoing connections (o) and the distance to the root concept (d). This weighting process has two assumptions in structural effects on concept importance: firstly, both authority nodes and nodes with incoming connections are considered more important than hub nodes or nodes with outgoing connections, and secondly, nodes close to the root node are considered more important than nodes more distant from the root node (Leake et al., 2004). The connectivity root distance weight of each node is called as W(c) and computed with equation (9). The weights are shown in table 4.7.

$$W(C) = (\alpha \cdot o + \beta \cdot i) \cdot (1/(d+1))^{1/\delta}, \alpha, \beta \geq 0, \delta \geq 1 \quad (9)$$

Table 4.7: The connectivity root distance weight of concept nodes in MCC map with α = 0.05, β = 0.5, δ =3.

Concept Label	Concept Title	Link #		connectivity root distance weight W(C)
		In (i)	Out (o)	
1	Operating System	0	11	0.7235
2	I/O System	1	2	0.6344
3	Distributed System	1	3	0.6543
4	OS Overview	1	3	0.6543
5	Mass Storage Structure	1	0	0.6204
6	I/O System Structure	1	0	0.6204
7	Distributed File System	1	0	0.6204
8	Distributed Coordination	1	0	0.6204
9	Distributed System Structure	1	0	0.6204
10	OS Structure	1	0	0.6204
11	Computer System Structure	1	0	0.6204
12	OS Introduction	1	0	0.6204

Step 3: Combine instructor confident weight with connectivity root distance weight

Each concept node in MCC map is the aggregation of both the instructor confident weight and the connectivity root distance weight with equation (10) and represented in Table 4.8.

$$AW(C) = \mu \cdot ICW + (1-\mu) \cdot W \qquad (10)$$

The parameters, μ and (1-μ), are used to weight complementarily the instructor confident weight and the connectivity root distance weight. The μ=0.5 is the optimal ratios for the two weights dynamically but in this work, μ=0.8 is used to achieve the collaborative design goal.

Table 4.8: The aggregated weight of concept nodes in MCC.

Concept Label	Concept Title	Weight		Aggregated Weight AW(C)
		ICW(C)	W(C)	
1	Operating System	1.0000	0.7235	0.9447
2	I/O System	1.0000	0.6344	0.9269
3	Distributed System	1.0000	0.6543	0.9309
4	OS Overview	0.7862	0.6543	0.7598
5	Mass Storage Structure	0.4697	0.6204	0.4998
6	I/O System Structure	1.0000	0.6204	0.9241
7	Distributed File System	0.7508	0.6204	0.7247
8	Distributed Coordination	0.7441	0.6204	0.7194
9	Distributed System Structure	1.0000	0.6204	0.9241
10	OS Structure	0.0000	0.6204	0.1241
11	Computer System Structure	0.7862	0.6204	0.7530
12	OS Introduction	0.7862	0.6204	0.7530

Generating Integrated Instructor Intention Maps

This process is generating the integrated instructor intention maps that designed by various instructors into single course concept map. This concept map will be described with ontology and will be used in the recommendation system. In this work, the threshold $\lambda = 0.5$ is used for concept node keeping judgments. The Table 4.8 shows the results that concept nodes, “Mass Storage Structure” and “OS Structure”, will be deleted from the main course concept map, so the final map result is shown in Figure 4.11.

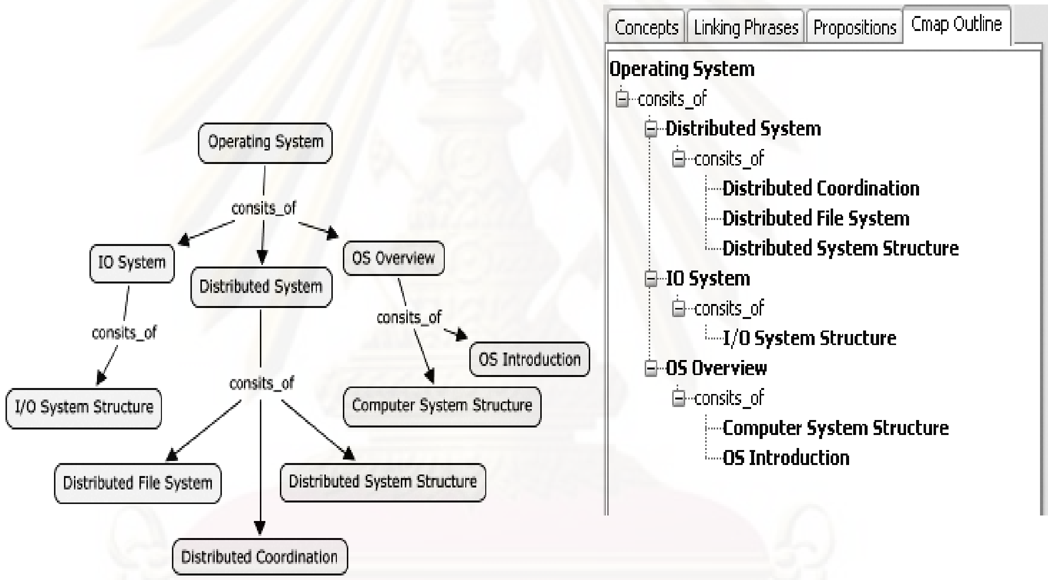


Figure 4.11: The integrated course concept map of four instructors.

Process 5: Course Concept Map Acceptation

Recheck the validation of integrated course concept map. Two choices of this process are accepted or not accept. If an instructor accepts the design, the integrated concept map will be used in filtering process of learning object recommendation. On the other hand, if not accept they will return to the process of concept map designing.

Process 6: Collective Feedback Processing

Instructors send feedback to the system. Then, the system processes all of feedback and collects them to the database. Finally, the result of this process is used to consider the most suitable course concept map in adaptive learning system.

Process 7: Iteration of Design Process

In case of new instructors or new designs requirements, the main course concept map or the instructor's intention map is changed. The system is restarted all ordering process and return the new integrated concept map to the learning object recommendation.

To use the integrated concept map in multi agent-based system, we design the concept map in form of ontolgogy that is proposed in (Pukkhem and Wiwat, 2009).

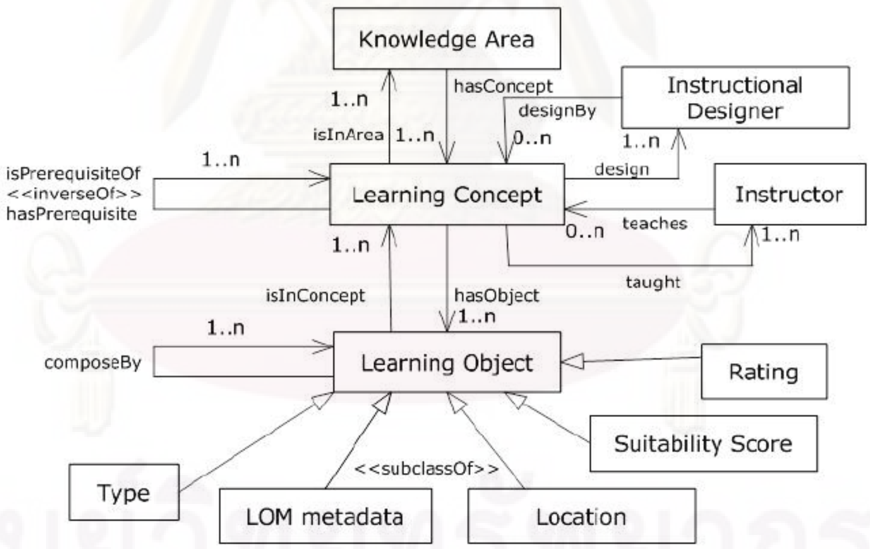


Figure 4.12: The course concept ontology and its class description.

From Figure 4.12, the concept map ontology presents the concepts of knowledge domain. Each instance of concept class has a name and relates to other concepts in two ways: i) as a pre-requisite or ii) as a sub-concept. Moreover, it has a meaning to refer to their knowledge area, author and its candidate learning objects, forming the domain's conceptual map. Instances of learning object class represents the learning objects used in the concept. In addition, some attributes of concept class also define on LOM standard, described their technical and pedagogical characteristics. Figure 4.13 shows the extended LOM metadata consisting of main categories in LOM, such as General, Educational and Technical.

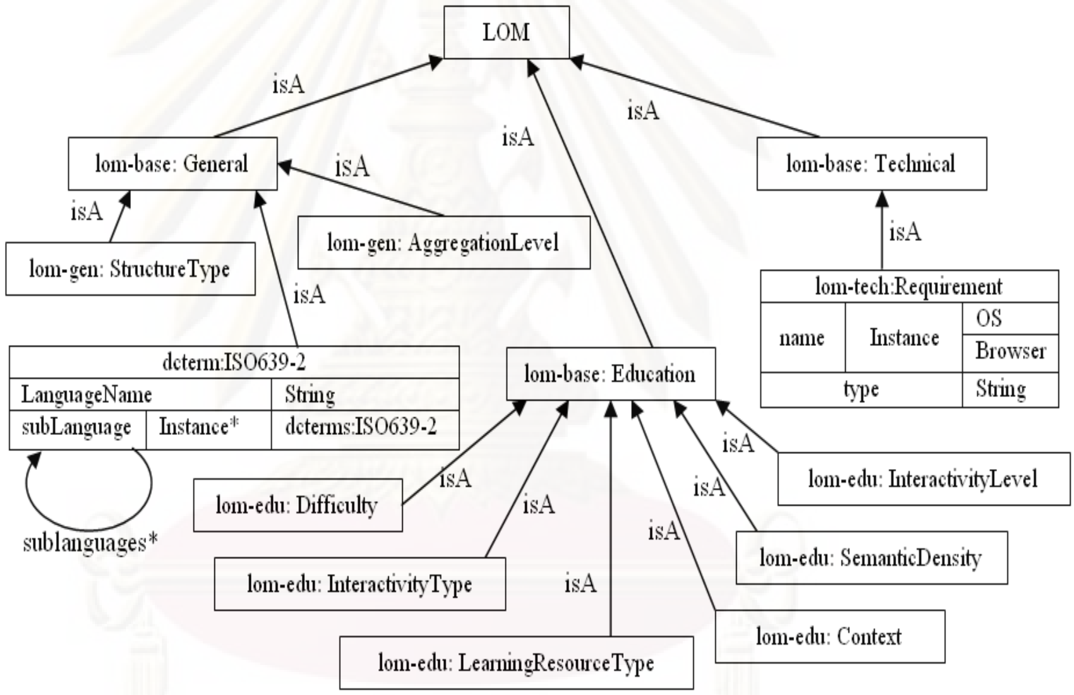


Figure 4.13: LOM standard ontology.

In each category class, there is an instance to represent the characteristics of candidate learning objects that shown in Figure 4.14. For this reason, it is possible to compare data when selecting learning objects to the concept or course, aiming to improve the assistance to the learner's need.

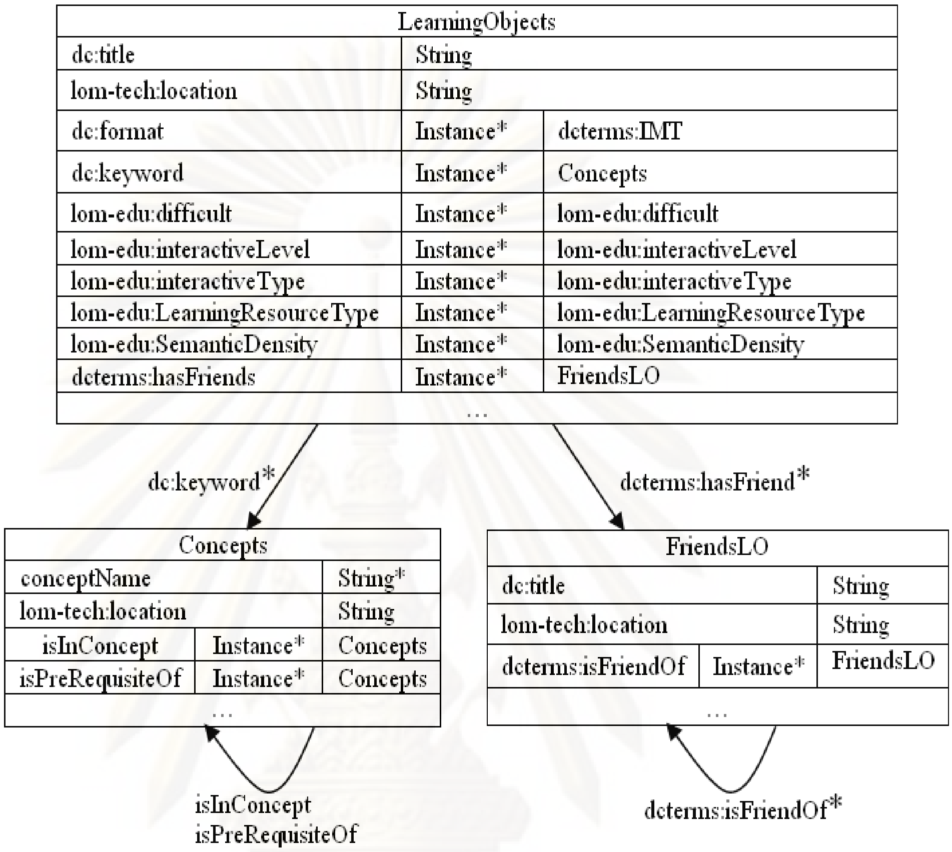


Figure 4.14: Learning object classes ontology.

4.2.4 Concept Map Combination Model Reliability

To verify the proposed method, the reliability analysis was carried out. An instructor questionnaire with seven questions was established and the detail is listed in Appendix A.3. 13 instructors in major of Computer Science and Information Technology of Thaksin University were invited to fill out the questionnaire. The test score and the variance of each question are listed in Table 4.9.

Table 4.9: The test scores from questionnaires.

Instructor ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Score
1	4	4	5	4	3	5	4	29
2	5	5	4	4	3	5	5	31
3	4	4	4	3	3	4	3	25
4	5	5	5	5	2	4	4	30
5	5	5	5	5	4	5	5	34
6	5	4	4	5	3	4	4	29
7	5	5	5	5	3	5	5	33
8	3	4	4	4	3	4	4	26
9	5	5	5	5	4	4	4	32
10	4	5	5	4	4	5	5	32
11	5	4	5	5	4	5	5	33
12	4	3	3	4	2	4	5	25
13	5	5	4	5	3	4	4	30
variance	0.4359	0.4359	0.4359	0.4359	0.4744	0.2692	0.4231	9.2436

Through the test scores from the questionnaires filled out, the reliability can be measured by Cronbach's α coefficient (Cronbach, 1981). The coefficient α can be calculated as follows:

$$\alpha = \left(\frac{n}{n-1}\right)\left(1-\frac{\sum_{i=1}^{n} Si^2}{St^2}\right) \tag{11}$$

where n is the number of components in the questionnaire, Si^2 is the variance of component i, and St^2 is the variance of the observed total test scores. The variance of the observed total test scores (St^2) is 9.2436 and individual variance is listed in Table 4.8. The coefficient α can be calculated by equation (11).

In the analysis, the reliability level of the proposed model can be defined by the reliability reference model. The reliability is classified in six level and related ranges of Cronbach's α are summarized in Table 4.10. It is demonstrated that the proposed combination model is strongly reliable due to α being 0.7459.

Table 4.10: Reliability levels and relevant ranges of Cronbach's α coefficient.

Assigned Range	Reliability Level
$\alpha \leq 0.3$	Unreliable
$0.3 < \alpha \leq 0.4$	Few Reliable
$0.4 < \alpha \leq 0.5$	Slightly Reliable
$0.5 < \alpha \leq 0.7$	Reliable
$0.7 < \alpha \leq 0.9$	Strongly Reliable
$0.9 < \alpha$	Very Strongly Reliable

4.3 Learning Object Recommendation Algorithms

The learning object recommendation method is divided into several steps. A learner selects the course which he/she wants to learn. The concept map from proposed combination model will be used for a specific course. The lessons of specific course will be shown and the learner can select an interesting topic (concept) contained each lesson to find the related learning objects.

In this work, the recommendation techniques based on learning style model were used to solve the problem of personalized selecting learning object. A system can recommend learning object according to a learner's preferences and attract the learner to come back for more.

In order to recommend the compatible learning object to learner, the concept map is defined to the concept structure that learners should learn. The concept map is analyzed by various instructors who have teaching experience and knowledge about specific course. Then, the different learning object recommendation algorithms are developed to solve individualized selection problem. The detail of each algorithm is described in the following subsection.

4.3.1 Information Requirement for Recommendation Algorithm

The existing learning object metadata specifications (such as IEEE LOM) defined a set of attributes that describe learning objects. The suitability of a learning object, however, is a contextual feature. It can be decided only when the learning object is situated in a certain context. To recommend the most compatible learning object to learner, information about learner and learning situation is necessary in addition to information about the learning object itself. Besides feature and requirement matching, the suitability of a learning object depends on some features that are more difficult to describe and measure. The historical usage from previous learners can provide valuable information for recommending learning object to target learner.

According to (IMS MD, 2008), learning objects represent any digital resources that complex learning object is the course, while the finest granularity learning object is the elementary educational resource. We have conceptualized the learning material using the hierarchical organization illustrated in Figure 4.15. Each course consists of several lessons or chapters, and each chapter contains several topics. The lowest level topic contains the actual course resources. Each such elementary learning object corresponds to a physical file and has a metadata file associated to it. This fine grained representation of the learning content is needed to insure the adaptation and modeling requirement.

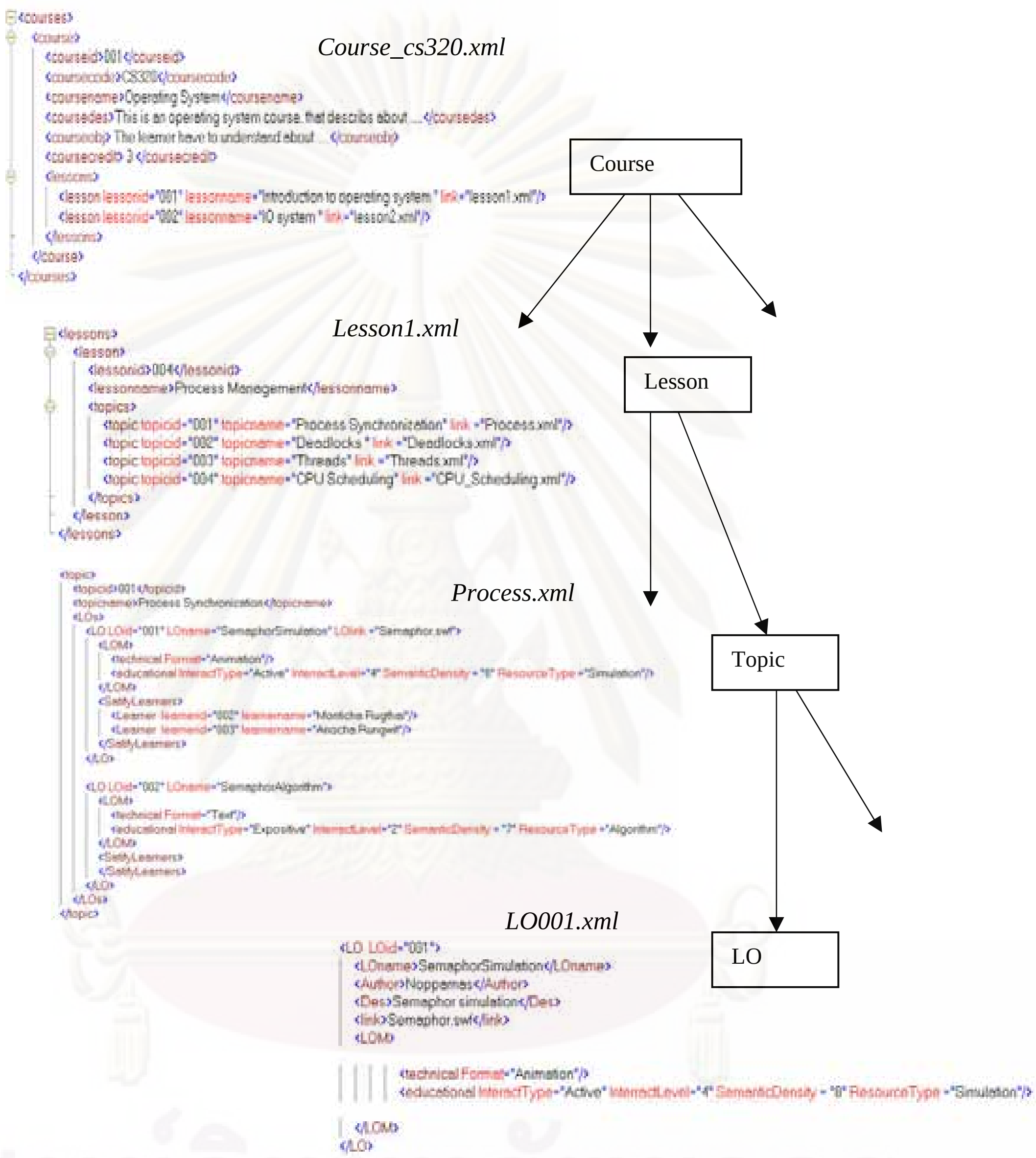


Figure 4.15: Organization of learning object content in recommendation model.

The following subsections discuss attributes related to the three areas required for learning object recommendation. It is not necessary to get explicit input for ever attribute in order to perform recommendation process. Some of them can be inferred from other attributes, and also sometimes the selection has to be done while some information is lacking. We need the three kinds of information for using as input value of proposed recommendation system.

4.3.1.1 Information of Course Concepts

The information of Concept Map (see the detail in Section 4.2) is represented in XML format when the course concept is used in the system. The example of XML file is shown in Figure 4.16.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <coursed xmlns:p="http://purl.org/dc/element/1.1">
  - <courses>
    - <course>
        <courseid>001</courseid>
        <coursecode>CS320</coursecode>
        <coursename>Operating System</coursename>
        <coursecreater>Noppamas Pukkhem</coursecreater>
        <coursedes>This is an operating system course, that describs about ...</coursedes>
        <courseobj>The learner understand about .....</courseobj>
        <coursecredit>3</coursecredit>
        <courselang>En</courselang>
      - <lessons>
          <lesson lessonid="001" lessonname="OS Overview" />
          <lesson lessonid="002" lessonname="IO System" />
          <lesson lessonid="003" lessonname="Distributed System" />
          .....
        </lessons>
      </course>
    </courses>
  </coursed>
```

Figure 4.16: The example of XML course file for the Operating System course.

4.3.1.2 Information of Learners

Learner information consists of Learner Style Set (LSS) and Learner Preference Set (LPS). Figure 4.17shows the learning style of learners in XML format and will be converted to LPS with applyRule() function in Figure 4.18 .

```
<?xml version="1.0" encoding="UTF-8" ?>
- <LearnerStyleSet>
  - <Learner>
      <learnerid>001</learnerid>
      <learnername>Piyanut Wichensang</learnername>
      <learnerMajor>IT</learnerMajor>
      <learnerYear>4</learnerYear>
      <learnerStyle A="1" R="0" S="0.5" I="0.5" U="1" B="0" Q="0" G="1" />
    </Learner>
  - <Learner>
      <learnerid>002</learnerid>
      <learnername>Wipawan Bilgorthem</learnername>
      <learnerMajor>IT</learnerMajor>
      <learnerYear>4</learnerYear>
      <learnerStyle A="0.5" R="0.5" S="0.5" I="0.5" U="0.5" B="0.5" Q="0.5" G="0.5" />
    </Learner>
  - <Learner>
      <learnerid>003</learnerid>
      <learnername>Nantawut Jareantiwakorn</learnername>
      <learnerMajor>IT</learnerMajor>
      <learnerYear>4</learnerYear>
      <learnerStyle A="0.5" R="0.5" S="1" I="0" U="1" B="0" Q="0.5" G="0.5" />
    </Learner>
    ....
  </LearnerStyleSet>
```


Figure 4.17: The example of XML file for describing learning style of learners.

```
public void applyRule(){
   // module for convert the learner style set (LSS) to learner preference set (LPS)
    //read Learner Style Set from LSS.xml
      for (int i=0;i<learnerStyleVector.size();i++){
           LearnerStyle style = new LearnerStyle();
           LearnerPreferSet preferSet= new LearnerPreferSet();

           style = (LearnerStyle)learnerStyleVector.get(i);
           preferSet.setId(style.getId());
            if(style.getWeight(0)!=0){
                    preferSet.setWeight(0, style.getWeight(0));
                    preferSet.addFeature(0, "Active");
                    preferSet.addFeature(0, "Mixed");
                    preferSet.addFeature(0, "Execise");
                    preferSet.addFeature(0, "Simulation");
                    preferSet.addFeature(0, "Experiment");
                    }
            if(style.getWeight(1)!=0){
                   preferSet.setWeight(1, style.getWeight(1));
                   preferSet.addFeature(1, "Expositive");
                   preferSet.addFeature(1, "Definition");
                   preferSet.addFeature(1, "Algorithm");
                   preferSet.addFeature(1, "Example");
                   }
```

```
          ..........
          if(style.getWeight(6)!=0){
                    preferSet.setWeight(6, style.getWeight(6));
                    preferSet.addFeature(6, "Text");
                    preferSet.addFeature(6, "Audio");
                    preferSet.addFeature(6, "5");
                    preferSet.addFeature(6, "6");
                    preferSet.addFeature(6, "Exercise");
                    preferSet.addFeature(6, "Algorithm");
                    preferSet.addFeature(6, "Slide");
                    }
          if(style.getWeight(7)!=0){
                    preferSet.setWeight(7, style.getWeight(7));
                    preferSet.addFeature(7, "Image");
                    preferSet.addFeature(7, "Index");
                    }
        // apply all rule
                    learnerPreferVector.add(preferSet);

                    }
      printLSV();  // apply rule from style vector to prefer vector
  }
```

Figure 4.18: The fragmentation of applyRule() for converting LSS to LPS.

4.3.1.3 Information of Learning Objects


```
 <?xml version="1.0" encoding="UTF-8" ?>
- <LOs>
 - <LO LOid="001">
     <LOname>SemaphorSimulation</LOname>
     <Author>Tim S. Roberts</Author>
     <Des>This is the simulation of process ....</Des>
   - <LOM>
       <technical Format="Animation" />
       <educational InteractType="Active" InterractLevel="4" SemanticDensity="8" ResourceType="Simulation" />
     </LOM>
   </LO>
 - <LO LOid="002">
     <LOname>SemaphorAlgorithm</LOname>
     <Author>Niclas Winquist</Author>
     <Des>A semaphore restricts the number of simultaneous users ...</Des>
   - <LOM>
       <technical Format="Text" />
       <educational InteractType="Expositive" InterractLevel="2" SemanticDensity="7" ResourceType="Algorithm" />
     </LOM>
   </LO>
   ....
 </LOs>
```


Figure 4.19: The example of XML file for describing the features of learning objects.

Learning Object Set (LOS) in this thesis is explained a characteristic of learning object such as format (animation, text, etc.), interactivity type and learning resource type. All of feature describe in LOS are compared with learner preference set (LSP) in section 4.3.1.2, the XML file of LOS is presented in Figure 4.19.

4.3.1.4 Information of Learning Objects Preference History

The information about learning objects preference history is very important for collaborative filtering algorithm, because the similarity between the learner and other learners is considered and used to define the most compatible learning object to learner.


```
<?xml version="1.0" encoding="UTF-8" ?>
- <LOsatisfy>
  - <LO LOid="001" LOname="SemaphorSimulation">
    - <Learners>
      - <Learner learnerid="002">
          <ILSscore A="7" R="4" S="6" I="5" U="6" B="5" Q="6" G="5" />
        </Learner>
      - <Learner learnerid="003">
          <ILSscore A="7" R="4" S="8" I="3" U="6" B="5" Q="6" G="5" />
        </Learner>
        ...
      </Learners>
    </LO>
  - <LO LOid="002" LOname="SemaphorAlgorithm">
    - <Learners>
      - <Learner learnerid="001">
          <ILSscore A="9" R="2" S="4" I="7" U="11" B="0" Q="3" G="8" />
        </Learner>
      - <Learner learnerid="008">
          <ILSscore A="4" R="7" S="6" I="5" U="8" B="3" Q="3" G="8" />
        </Learner>
        ...
      </Learners>
    </LO>
  - <LO LOid="003" LOname="Semaphor">
    - <Learners>
      - <Learner learnerid="006">
          <ILSscore A="9" R="2" S="10" I="1" U="9" B="2" Q="4" G="7" />
        </Learner>
      - <Learner learnerid="009">
          <ILSscore A="9" R="2" S="9" I="2" U="8" B="3" Q="6" G="5" />
        </Learner>
        ...
      </Learners>
    </LO>
    ...
  </LOsatisfy>
```


Figure 4.20: The example of XML file for describing the history of learning object preferences.

Figure 4.20 shows the records of preferred learning object of learners. This information is used in the process of neighbor-based filtering of learning object recommendation algorithm.

4.3.2 Non-personalized Recommendation Algorithm

The following non-personalized algorithms are examined to provide the comparison of recommended accuracy in evaluation experiment.

4.3.2.1 Random Algorithm (Rand)

This algorithm use Random function to randomly select the learning objects in the same topic independently from what evaluations on the learner or other learners. The detail of this algorithm is shown in Figure 4.21.

ALGORITHM 4: Random Algorithm

<u>INPUT</u>: Learner ID *//learner*
Learning Object Sets *(LOSs)* of topic j *// all candidate learning objects in topic j*
<u>OUTPUT</u> : Predicted learning object (LO_{pd})
FUNCTION: RandomLO()
//randomly selects LO for learner i
FOR EACH L_i //consider all of learning objects in the same concept
    FOR EACH *LOS of learning object i*
        IF $(LOi \in LO_{Topic\ j})$
        THEN $LO_{pd} = Rand(LOi, \ldots, LOn)$
        BREAK
  RETURN $RandLO()=LO_{pd}$
END FUNCTION

Figure 4.21: Random Algorithm.

4.3.2.2 Arithmetic Mean (AriMean)

AriMean algorithm calculates a recommendation as the arithmetic mean of each learning object that other learners prefer, independently of how similar they are to the learner. The candidate learning object that most popular in the same concept will be chosen to learner. The AriMean algorithm is described in Figure 4.22.

ALGORITHM 5: AriMean Algorithm

<u>INPUT</u>: Learner ID *//learner*
Learning Object Sets *(LOSs)* of topic j *// all candidate learning object in topic j*
Learning Objects Preference History
<u>OUTPUT</u> : Predicted learning object (LO_{pd})
FUNCTION: AriMeanLO()
//find the most popular learning object in the same concept by using the LO preference
//History
//consider all of learning objects in the same concept
    FOR EACH LO_i *in LO preference History*
        FOR EACH L_i *// consider all learners that used* LO_i
            IF ($L_i \in$ { *Satisfy Learners*})
            THEN Count = Count +1
        END
            $Mean_{LOi}$ = *Count / Number of* L_i
            RANK LO_i *by maximum mean*
            SET LO_{pd} = *LO ID of LO that has maximum mean*
    BREAK
    RETURN *AriMeanLO()=*LO_{pd}
END FUNCTION

Figure 4.22: AriMean Algorithm.

For example, we have 34 learners in the same concept with three candidate learning objects (LO1, LO2, LO3). The learning object preference history shows: LO1 was selected by 15 learners, LO2 was selected by 8 learners and LO3 was selected by 11 learners. Then, by using the AriMean recommendation algorithm we can compute mean of using of each learning object as follows.

Mean score of LO1 = 15/34 = 0.4412

Mean score of LO2 = 8/34 = 0.2353

Mean score of LO3 = 11/34 = 0.32358

So, the order of learning object for learner when used the AriMean algorithm are presented as LO1, LO3 and LO2.

4.3.3 Preferred feature-based Recommendation Algorithm (PFB)

The learner analysis process and learning object modeling in chapter III bring us to know about learners' learning styles and significant learning object features. The values of feature of a learning object can help determine if a learner may prefer the learning object. For example, under the feature format of learning object, learning object may compose of various media, such as text document, audio/video, picture, and etc. Different learners may prefer different formats of learning object for the same concept depended on their learning styles. The preferred feature-based algorithm is to bias the learning objects with a learner's preferences. Learning object tending to suit a learner's preference more will get higher priorities when it is ranked to the learner. We propose two variations of preferred feature-based recommendation algorithm – non-weighting feature preferred feature-based (NWF-PFB) and weighted feature preferred feature-based (WF-PFB). We present the detail of each variation in the following subsections.

4.3.3.1 Non-weighting feature preferred feature-based (NWF-PFB) recommendation

In NWF-PFB, the preference score (PS) was calculated by NWF-PFB algorithm. The result shows the suitability of each learning object to learner independently from feature weighting. So, in this algorithm we define a feature frequency weight of learning object features as 1 (ω = 1) in every learning object feature.

4.3.3.2 Weighted feature preferred feature-based (WF-PFB) recommendation

In WF-PFB, the learning object feature is weighed by using the frequency that target feature is referred in learning object selection rule. To define the value of ω, the relation between LPS and LOS is shown in Figure 4.23 is considered.

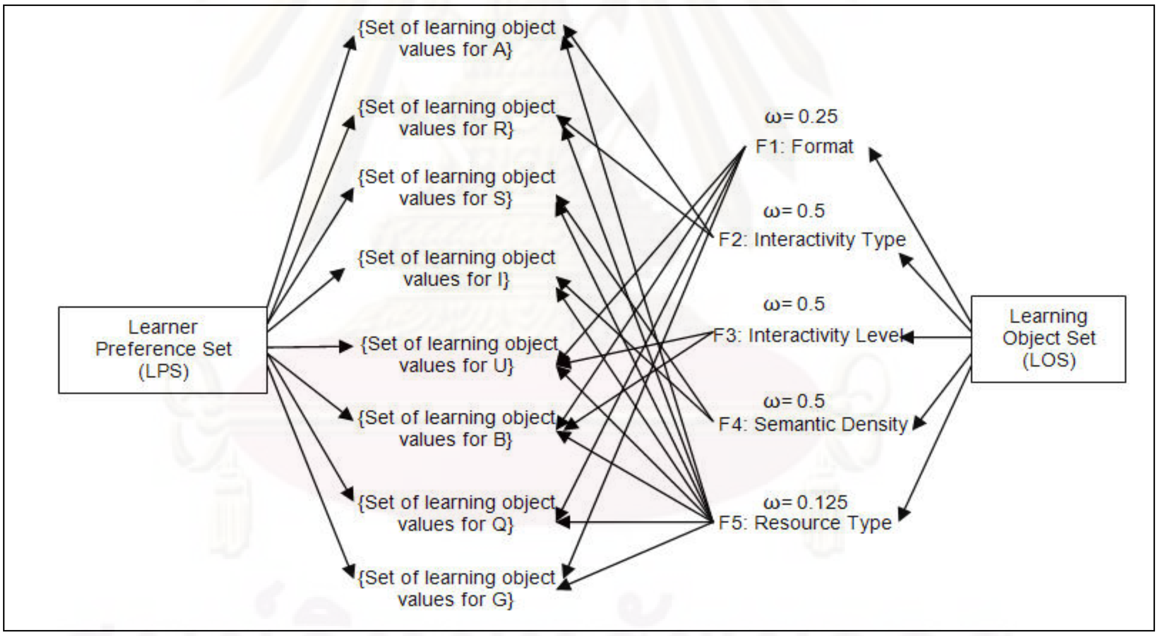


Figure 4.23: The relation between LPS and LOS.

We note that the frequency of learning object features when are referred in a learning object selection rules are different In WF-PFB, ω is computed by $\omega = \dfrac{1}{\sum F_i \text{ appearing in each Rule}}$ and the result of ω of learning object features are shown in the Figure 4.21 .

Both of NFW-PFB and FW-PFB are described by using Preferred feature-based Algorithm, but they have different in variations of ω. We define the notation which use in this algorithm as Figure 4.24 and the fragment of source code written in Java is presented in Figure 4.25.

```
ALGORITHM 6: Preferred Feature-based Algorithm
-------------------------------------------------------------
INPUT:    Learner preference set (LPS)
          Specific learning object set (LOS)
          Two choices of variation of feature frequency weight (ω)
              NWF-PFB , ω = 1  for each learning object feature i
          Or   WF-PFB, ω = 1 / (# of RF_i) , RF is the frequency of referred feature.
OUTPUT : Preference Score (PS) of specific LO
FUNCTION:  Preference_Score_Calculation ()
//compute PS of all learners
    FOR EACH LPS
   // compute PS of learner with all learning objects
      FOR EACH   LOS of  learning object i
      INT PS = 0
          //compute all of learner styles {A, R, S, I, B, U, Q, G} in LPS
          FOR EACH  PF_i  ∈ LPS (L)
              IF  ( PF_i  = F_i) and FWi<>0
              THEN  PS =  PS+ ω FW_i
      BREAK
     RETURN  Preference_Score_Calculation()=PS
END FUNCTION
```

Figure 4.24: Preferred feature-based Algorithm.

```
public void computeScore(){

        // compute score from prefer vector and lom vector to score vector
        //for each preference set LPS

            for (int i=0;i<learnerPreferVector.size();i++){
              LearnerPreferSet preferSet = new LearnerPreferSet();
              preferSet = (LearnerPreferSet)learnerPreferVector.get(i);
              LearnerPreferLo preferLo = new LearnerPreferLo();
              preferLo.setId(preferSet.getId());

        //for each LOS
              for (int j=0;j<loVector.size();j++){
                LoM lom = new LoM();
                lom = (LoM)loVector.get(j);
                float loScore = 0;
                for (int k=0;k<8;k++){
                  float weight = preferSet.featureSet[k].getWeight();
                  if(weight!=0.0){
                    for(int x=0;x<preferSet.featureSet[k].featureVector.size();x++){
                      String feature = preferSet.featureSet[k].getFeature(x);
                      if (lom.getTechFormat().equals(feature)){
                        loScore = loScore + weight;    // use  0.25 * weight in WF-PFB
                      }
                      if (lom.getInteractType().equals(feature)){
                        loScore = loScore + weight;   // use  0.5 * weight in WF-PFB
                      }
                      if (lom.getInteractLevel().equals(feature)){
                        loScore = loScore + weight;  // use  0.5 * weight in WF-PFB
                      }
                      if (lom.getSemanticDens().equals(feature)){
                        loScore = loScore + weight;  // use  0.5 * weight in WF-PFB
                      }
                      if (lom.getResourceType().equals(feature)){
                        loScore = loScore + weight; // use  0.125 * weight in WF-PFB
                      }
                    }
                  }
                }
                LoPreferScore preferScore = new LoPreferScore();
                preferScore.setId(lom.getId());
                preferScore.setScore(loScore);
                preferLo.addPreferScore(preferScore);
              }
              loScoreVector.add(preferLo);
            }
}
```

Figure 4.25: Fragment of Java source code implemented with PFB algorithm.

The information of learner model in our experiment that presented in Chapter III is used to be an input for learning object recommendation. The example of computation is described as follows.

Table 4.11: The example of learner learning style sets.

Learner ID	Weight of Learning Preference							
	A	R	S	I	U	B	Q	G
001	1	0	0.5	0.5	1	0	0	1
011	1	0	1	0	1	0	0.5	0.5
027	1	0	1	0	1	0	1	0

strong (w=1), medium (w=0.5), weak (w=0), w=preference weight

We can generate the information about learners following Table 4.11 by using the definition 3.2 from Chapter III:

LSS_{001} = {*(A,1), (R,0), (S,0.5), (I,0.5), (U,1), (B,0), (Q,0), (G,1)*}

LSS_{011} = {*(A,1), (R,0), (S,1), (I,0.5), (U,1,), (B,0), (Q,0.5), (G,0.5)*}

LSS_{027} = {*(A,1), (R,0), (S,1), (I,0), (U,1,) (B,0), (Q,1), (G,0),* }

Then, by using the learning object selection rules such as,

<u>Mapping 1.</u> Recommend learning object for "**A-A**ctive" learner

If "A" ∈ LS(L)

Then Lom.education.interactivity_type = active or mixed

And Lom.educational.LearningResourceType = exercise or simulations or experiment

<u>Mapping 2.</u> Recommend learning object for "**R-R**efective" learner

If "R" ∈ LS(L)

Then Lom.education.interactivity_type = expositive

And Lom.educational.ResourceType = definition or algorithm or example

. . .

The LPS of leaner ID 001 can be defined with definition 3 as follows:

LPS_{L1} = {*({exercise, simulations, experiment, active, mixed},1), ({simulation, experiment, 8, 9}, 0.5), {definition, exercise, 5, 6, 7}, 0.5), ({video, image, animation, simulation, 3, 4, 5},1), ({image, index},1)*}

For example, the concept of "Process" of Operating System course is used to demonstrate learning object recommendation for learner. The information about related learning object is represented as follows:

LOS_{001} = { *animation , active, 4, 8, simulation* }
LOS_{002} = { *text , expositive, 2, 7, algorithm* }
LOS_{003} = { *video , active, 4, 7, definition* }

When use the Preference-contented based algorithm for computing the PS of each LO of Learner ID 001, the results are $PS(LO_{001})$ = 0.6, $PS(LO_{002})$=0.05 and $PS(LO_{003})$= 0.5. Therefore, the recommendation order is LO1, LO3 and LO2.

4.3.4 Neighbor-based Collaborative Filtering Recommendation Algorithm (NB-CF)

The main problem of content-based recommendation approach is we have to know the information about learning objects before recommend them to learner. In some situations, the uncompleted metadata filling when import learning objects to the system may occur. So, it hides some compatible learning objects from learner accessing.

To solve this problem, the suggestions from other learners can solve this problem. The assumption is the learner who has the similar preference as the learner should has a higher probability for selecting the same learning object. For this reason, the collaborative filtering approach is integrated in proposed recommendation algorithm to strengthen the precise of recommendations. This algorithm is called “Nearest neighbor-based algorithm”. It predicts how helpful a learning object will be for a learner by analyzing other similar learner’s feedbacks. A similar learner group is defined as the group of learners who used the same learning objects in the past and returned similar feedbacks. Two main steps are carried out in the nearest neighbor-based algorithm.

Step 1: Collect the related learning objects in the same concept by using the concept map that described in section *4.2*.
Step 2: Extract preferred learners of each related learning object.
Step 3: Compute the neighbor-based score (NS) of each learning object.

The result of this algorithm is the average ranking of the three most similarity neighbors between the learner (SL) and preferred LO learners (PL). We normalize the weight of this value with discount from value 1. So the neighbor score (NB) is 1 – MDIS, where MDIS is the mean of distance. The NB score will be assigned to each preferred learning object for the ranking method. The detail of neighbor-based algorithm is shown in Figure 4.26 and the fragment of source code written in Java is presented in Figure 4.27.

ALGORITHM 7: Nearest Neighbor-based Algorithm
INPUT: Preferred Learning object ID LSS of learner (SL) LSS of preferred learner (PL) of preferred LO n = number of learner style preference k = number of nearest neighbors (k=1, k=3, k=5, k=7, k=9) OUTPUT : Neighbor Score *(NS)* of preferred *LO* FUNCTION: Neighbor_Score_Calculation() FOR EACH *LSS of SL* FLOAT DIS =0, MDIS=0 *// compute distance between SP and PL by using learner style preference in every //dimension* FOR EACH *LSS of PL of preferred LO* FOR EACH *(P_i in LSS)* $DIS_{(SL,PL)} = DIS_{(SL,PL)} + Sqr((P_{SL} \wedge 2) - (P_{PL} \wedge 2))$ *// return k learners who have the least distance of all PLs* FOR ALL $DIS_{(SL,PL)}$ *between SL and PLs* *Rank*($DIS_{(SL,PL)}$) RETURN *Last k of* $DIS_{(SL,PL)}$ $MDIS = SUM(DIS_{(SL,PL)})/k$ RETURN *Neighbor_Score_Calculation()=1-MDIS* END FUNCTION

Figure 4.26: Nearest Neighbor-based Algorithm.

```
public void computeScore(int n){

        for (int i=0;i<learnerStyleVector.size();i++){
            LearnerStyle spLearner = new LearnerStyle();
            spLearner = learnerStyleVector.get(i);    // sp is learner

            LearnerNeighborLo lnl = new LearnerNeighborLo();
            lnl.setId(spLearner.getId());
             System.out.println("Learner id = " + spLearner.getId());

                  for (int j=0;j<loLSVector.size();j++){
                     System.out.println("LO : " + loLSVector.elementAt(j).getId());
                      Vector<Float> score = new Vector<Float>();
                           for (int k=0;k<loLSVector.elementAt(j).learnerSelectLoVector.size();k++){
                                  LearnerStyle cLearner = new LearnerStyle();
                                  cLearner = loLSVector.elementAt(j).learnerSelectLoVector.get(k);
                                         // compute similarity distance
                                         double sum = 0;
                                         for (int style=0;style<8;style++){
                 double pw_sp = java.lang.Math.pow(spLearner.getWeight(style),2);
                 double pw_c = java.lang.Math.pow(cLearner.getWeight(style),2);
                  sum = sum + java.lang.Math.sqrt(java.lang.Math.abs(pw_sp- pw_c));
                                         }
                  sum = sum/8;
                  score.addElement(Float.parseFloat(String.valueOf(sum)));
                               }

             Collections.sort(score);
             float sum_1=0;
             for (int a=0;a<n;a++){
                 sum_1 = sum_1 + score.elementAt(a);
                 System.out.println("score = " + score.elementAt(a));
             }
             float nScore = 1-(sum_1/n);

             NumberFormat formatter = new DecimalFormat("0.0000");
             String s = formatter.format(nScore);
             System.out.println("k-neighbor score = " + s);
             float nbScore = Float.parseFloat(s);
             LoNeighborScore neighborScore = new LoNeighborScore();
             neighborScore.setId(loLSVector.elementAt(j).getId());
             neighborScore.setScore(nbScore);
             lnl.addPreferScore(neighborScore);
                      }

         loNBScoreVector.add(lnl);
           // add all score of each learner to learnerNBScoreVector
           }

    }
```

Figure 4.27: Fragment of Java source code implemented with NB algorithm.

Computation of Neighbor Score (NS) of each related learning objects are helpful to strengthen the recommendation for the learner. These algorithm starts with collecting group of learners that prefer the same learning object. For example, we have a set of learners (SelLO) who prefer the same learning object of each learning object ID 001, 002 and 003 as below.

SelLO$_1$={L_{002}, L_{003}, L_{004}, L_{005} , L_{007}, L_{014} , L_{015} , L_{018} , L_{021}, L_{022}, L_{024}, L_{025} , L_{030} , L_{031} , }
SelLO$_2$={ L_{008}, L_{010} , L_{013} , L_{016} , L_{020} , L_{026} , L_{028} , L_{032}}
SelLO$_3$={L_{006}, L_{009}, L_{011}, L_{012}, L_{017}, L_{019} , L_{023} , L_{027} , L_{029}, L_{033} , L_{034} }

For this process, Learner ID 001 is defined to be learner (The information is shown in Table 4.8). The results of NS scores are presented in the Table 4.12.

Table 4.12: The NS scores of related LO for learner ID 001.

Learning Object ID	Top 3 of Similar Learners	NS Score
001	L_{005}, L_{018} , L_{030}	0.6585
002	L_{013} , L_{016} , L_{020}	0.8975
003	L_{012}, L_{019} , L_{034}	0.6926

Therefore, the recommendation order from this algorithm is LO2, LO3 and LO1.

4.4 Summary

In this chapter, the frameworks of learning object recommendation based on learning style is proposed. In this framework, the learning object recommendation algorithms are proposed in both of non-personalized recommendation algorithm and personalized recommendation algorithm. The personalized algorithms consist of content-based (NWF-PFB and WF-PFB) and collaborative recommendation (NB-CF). A system can recommend learning object according to a learner's preferences, and will attract the learner to come back for more using e-learning materials. The proposed algorithm works with our concept map combination model which solves the different designs of various instructors for increasing collaboration among instructors and supporting the learning object recommendation process in e-learning environment. The results of learner satisfaction in selecting the most compatible learning object when uses each of proposed algorithm are presented in Chapter V.

CHAPTER V

LEARNING OBJECT RECOMMENDATION ALGORITHMS TESTING AND RESULTS

The goal of the experimental testing is twofold. Firstly, to evaluate which of the proposed algorithms is more appropriate for the learning object recommendation based on learner learning styles. Secondly, to examine the appropriate parameterization of the proposed algorithms, by exploring the various design options. First, an experimental setting is present in Section 5.1. Section 5.2 concludes all proposed recommendations and their techniques. Next, the experimental results of each proposed algorithm are presented in Section 5.3. Then in section 5.4., the preference errors (PE) of five algorithms are compared with the same data set. In additional, the predictive behaviors of each recommendation algorithm are analyzed. Finally, Section 5.5 the discussion of how proposed method enhances the accuracy of learning object recommendation is presented.

5.1 Experimental Setting

All of cases in experiments, learning objects are recommended to learners by using different learning object recommendation algorithms, according to their learning styles. Candidate learning objects are filtered by a concept map which is created by Concept Map Combination Model (CMCM) and represented in term of Learning Object Sets (LOSs). Then, an actual preferred feedbacks (actual preferred learning object) from learners are evaluated according to the preference score (PS) and the neighbor score (NS) which are computed by recommendation algorithms. For content-based approach, the PS score represents the suitability of learning object according to learner preference degree in each learning object feature. Therefore, the learning object that has the highest PS score will be recommended to learner. For collaborative filtering approach, the NS score shows the degree of similarity between learner and other learner. It means the learning object that is preferred by other learner, who are the most similar to learner, will be recommend to learner.

The experiment is tested in the three following case scenarios.

Case Scenario 1: The learners in Computer Science and Information Technology, who are new learner in learning object learning environment, need to know the most compatible learning object based on their learning styles. Moreover, they want to know how different between actual preference and system prediction. The results are shown in Section 5.3.2.

Case Scenario 2: The learners want to know how similarity of their learning style when compared with other learners. What is the learning object that can infer from their friends? The results are shown in Section 5.3.1.

Case scenario 3: If the system does not provide the recommendation, how different between actual preference and system prediction. The results are shown in Section 5.3.1.

5.1.1 Participants

In experimental setting, participants are 142 undergraduate students in major of Computer Science (CS) and Information Technology (IT) from Faculty of Science, Thaksin University, Phattalung Campus. We divide the undergraduate students into four groups by their year and major of study. Group1 is 31 students of third-year in computer science major (CS3, n=31), group2 is 48 students of third-year in information technology major (IT3, n=48), group3 is 29 students of fourth-year in computer science major (CS4, n=29) and group4 is 31 students of fourth-year in information technology major (IT4, n=31).

5.1.2 Candidate Learning Objects

The default number of candidate learning objects for our experiment is defined to 5 learning objects with concept of “Semaphore” in “Operating System” course. The Learning Object Sets (LOSs) of candidate learning objects are described as follows:

LOS_{001} = {*animation, active, very high, high, simulation*}

LOS_{002} = {*text, expositive, low, medium, algorithm*}

LOS_{003} = {*video, active, very high, medium, definition*}

LOS_{004} = {*image, mixed, medium, medium, slide*}

LOS_{005} = {*image, mixed, low, low, index*}.

5.1.3 Evaluation Method

Each learner has to return preferred feedback (learning object ID that they most prefer) after he/she has studied every learning object in the same concept. To understand how the recommendation results affect learners, both of feedback analysis and *Preference error (PE)* between the real learner's preference and the system predictions will be compared. Observing the learner's feedbacks directly is to understand whether the proposed model recommends learning object in accord with learners' preference, while calculating PE shows whether it can infer learner's preference and interest accurately or not. The prediction accuracy is better when the PE value is lower. In the experiment, different variation of algorithms will be demonstrated to show the different results. PE can calculate by using equation (12):

$$PE = 1 - \frac{\sum_{i=1}^{N}(LO_{ac} \cap LO_{pd})}{N} \qquad (12)$$

where LO_{ac} is an actual referred learning object, LO_{pd} is the recommended learning object from the algorithm and *N* is the number of learners.

5.2 Summarization of Proposed Algorithms

In the same domain of environmental setting, five algorithms with three approaches are used to demonstrate the Preference error (PE) of each algorithm. Table 5.1 lists approaches, names and their techniques.

Table 5.1: A summarization of the various learning object recommendation algorithms.

Approach	Name	Technique
Non-personalized	*Rand*	Random function
	AriMean	Arithmetic Mean
Preferred-Content Based	*PFB*	Mapping Rule-Based
	FW-PFB	Mapping Rule-Based and Feature Weighting
Neighbor–based Collaborative Filtering	*NB-CF*	Euclidian Distance varied by number of neighbors

5.3 Experimental Results

The results of each algorithm are presented in the result of PE. The details of results are presented as follows.

5.3.1 The Results of Non-personalized Algorithm

To answer the question in scenario 3, two non-personalized algorithm: random and arithmetic mean algorithm are used to recommend the learning object to learners.

5.3.1.1 Random Algorithm (Rand)

The random algorithm is used to predict the learning object for learner by using random function. It defines the recommended learning object from Rand algorithm as LO_{pd}, when $LO_{pd} = Rand(LO_{001}, LO_{002}, LO_{003}, LO_{004}, LO_{005})$. Eight iteration tests show the stable PE results, so we calculate the mean of them to present the average PE of each group of learners. The PE results are shown in Table 5.2.

Table 5.2: PE results of Random algorithm.

Test No.	Rand PE			
	CS3	IT3	CS4	IT4
1	**0.7742**	0.8333	0.8276	0.7647
2	0.9355	0.9583	0.8966	0.9412
3	0.8710	0.8125	0.7931	0.7941
4	0.8710	0.8125	0.7586	0.8235
5	0.8710	0.8125	0.7931	0.7647
6	0.9032	**0.7083**	0.8621	0.8529
7	0.8387	0.8125	0.8621	0.7941
8	0.8710	0.8125	**0.7586**	**0.7059**
Average PE	**0.8670**	**0.8203**	**0.8190**	**0.8051**

The best result of PE in this experiment is appeared in group of IT4 students, about 0.7059, and the worth result is about 0.9583 in group of IT3 students. For comparing all of PE results in each group of learner shows that there is not much different among various test numbers and learner groups. However, the mean PE in every group were considered and we found that they are not lower than 0.8. Therefore, it seems to be high PE when use the non-personalized approach with Rand algorithm.

5.3.1.2 AriMean Algorithm

As we describe the detail of AriMean algorithm in previous chapter. AriMean algorithm is used to test in all group of learner. The PE results among them are not much different. The best PE result in this experiment is 0.3871 in group of CS3 learners. The range of PE result in the experiment is 0.3871 to 0.5172. In the experimental results, they are better than Random algorithm in every group of learners. The detail of experiment result is presented in Table 5.3.

Table 5.3: PE results of AriMean algorithm.

AriMean PE			
CS3	IT3	CS4	IT4
0.3871	0.4792	0.5172	0.3824

5.3.2 The Results of Preferred feature-based Algorithm

To compute the preference score (PS) of preferred feature-based algorithm for each learning object to learner, we implement the preference score computational program written by JAVA language and export the results in form of XML file to support the agent-based environment. DOM4J is used as the XML parser. The example of the results in this experiment is presented in Figure 5.3 and 5.4. Then, the result of our testing was applied using SPSS software package (SPSS, 2008) for calculating the preference error (PE). The results of two variations, Non-weighting-PFB and Weighted Feature-PFB, are proposed in subsection 5.3.3.1 and 5.3.3.2.

5.3.2.1 Non-weighting Feature PFB (NWF-PFB)

The example output of NWF-PFB is presented in Figure 5.3 and the PE results are shown in Table 5.4.

```
run:
Learner ID >> 001
Active : Mixed : Execise : Simulation : Experiment : 1.0
8 : 9 : Simulation : Experiment : 0.5
5 : 6 : 7 : Definition : Example : 0.5
Video : Image : Animation : 2 : 3 : 4 : Simulation : 1.0
Image : Index : 1.0
=======================
Learner ID >> 002
Active : Mixed : Execise : Simulation : Experiment : 0.5
Expositive : Definition : Algorithm : Example : 0.5
8 : 9 : Simulation : Experiment : 0.5
5 : 6 : 7 : Definition : Example : 0.5
Video : Image : Animation : 2 : 3 : 4 : Simulation : 0.5
Text : Audio : 0 : 1 : Definition : Exercise : 0.5
Text : Audio : 5 : 6 : Exercise : Algorithm : Slide : 0.5
Image : Index : 0.5
=======================
Learner ID >> 003
Active : Mixed : Execise : Simulation : Experiment : 0.5
Expositive : Definition : Algorithm : Example : 0.5
8 : 9 : Simulation : Experiment : 1.0
Video : Image : Animation : 2 : 3 : 4 : Simulation : 1.0
Text : Audio : 5 : 6 : Exercise : Algorithm : Slide : 0.5
Image : Index : 0.5
=======================
 . . .
=======================
Learner ID >> 001
LO_id = 001 >> 6.0
LO_id = 002 >> 1.5
LO_id = 003 >> 3.0
LO_id = 004 >> 4.5
LO_id = 005 >> 4.5

Learner ID >> 002
LO_id = 001 >> 3.5
LO_id = 002 >> 3.5
LO_id = 003 >> 3.0
LO_id = 004 >> 3.0
LO_id = 005 >> 3.5

Learner ID >> 003
LO_id = 001 >> 6.0
LO_id = 002 >> 3.0
LO_id = 003 >> 2.5
LO_id = 004 >> 3.5
LO_id = 005 >> 3.0
 . . .
```

Figure 5.1: Output example of preference score calculating with NFW-PFB algorithm.

Table 5.4: PE result of NWF-PFB algorithm.

NWF-PFB PE			
CS3	IT3	CS4	IT4
0.2903	0.2917	0.2759	0.3235

From Table 5.4, the PE result seems to be decreased when is compared with non-personalized algorithm above. The best result of PE is 0.2759 in group of CS4 learners and the worth result is 0.3235 in group of IT4 learners.

5.3.2.2 Weighted Feature-PFB (WF-PFB)

Weighted Feature-PFB (WF-PFB) is an adjusted variation of NWF-PFB that described in Section 5.3.2.2.

```
run:
Learner ID >> 001
Active : Mixed : Execise : Simulation : Experiment : 1.0
8 : 9 : Simulation : Experiment : 0.5
5 : 6 : 7 : Definition : Example : 0.5
Video : Image : Animation : 2 : 3 : 4 : Simulation : 1.0
Image : Index : 1.0
========================
Learner ID >> 002
Active : Mixed : Execise : Simulation : Experiment : 0.5
Expositive : Definition : Algorithm : Example : 0.5
8 : 9 : Simulation : Experiment : 0.5
5 : 6 : 7 : Definition : Example : 0.5
Video : Image : Animation : 2 : 3 : 4 : Simulation : 0.5
Text : Audio : 0 : 1 : Definition : Exercise : 0.5
Text : Audio : 5 : 6 : Exercise : Algorithm : Slide : 0.5
Image : Index : 0.5
========================
Learner ID >> 003
Active : Mixed : Execise : Simulation : Experiment : 0.5
Expositive : Definition : Algorithm : Example : 0.5
8 : 9 : Simulation : Experiment : 1.0
Video : Image : Animation : 2 : 3 : 4 : Simulation : 1.0
Text : Audio : 5 : 6 : Exercise : Algorithm : Slide : 0.5
Image : Index : 0.5
========================
...
========================
Learner ID >> 001
LO_id = 001 >> 1.8125
LO_id = 002 >> 0.75
LO_id = 003 >> 1.0625
LO_id = 004 >> 1.75
LO_id = 005 >> 1.375

Learner ID >> 002
LO_id = 001 >> 1.0625
LO_id = 002 >> 1.125
LO_id = 003 >> 0.8125
LO_id = 004 >> 1.0625
LO_id = 005 >> 1.3125

Learner ID >> 003
LO_id = 001 >> 1.8125
LO_id = 002 >> 1.0
LO_id = 003 >> 0.8125
LO_id = 004 >> 1.1875
LO_id = 005 >> 0.9375
...
```

Figure 5.2 Output examples of preference score calculating with FW-PFB algorithm.

The example output of WF-PFB is presented in Figure 5.4 and the PE results are shown in Table 5.5.

Table 5.5: PE results of WF-PFB algorithm.

WF-PFB PE			
CS3	IT3	CS4	IT4
0.2258	0.2083	0.2414	0.2353

From the PE results of WF-PFB that shown in Table 5.5, it is quite clearly for better PE result when we compared between NWF-PFB and other non-personalized algorithms. The best PE result is 0.2083 and it is not over than 0.25 in every group of learners.

Both of preferred feature-based algorithms, NFW-PFB and FW-PFB, give the quite good result of PE. The result shows that the selected learning object feature and set of learning object selection rules of proposed approach make the system know more about the learners. Therefore, the algorithm can recommend the compatible learning object to the learner nearly with their actual prefer.

5.3.3 The Results of Neighbor-based Collaborative Filtering Algorithm (NB-CF)

We test all groups of learner with different variations of nearest neighbor (k). The example output shown in Figure 5.6 and the PE results are shown in Table 5.6.

```
run:
Learner id = 001
LO : 001
score = 0.17075318
score = 0.34150636
score = 0.34150636
score = 0.34150636
score = 0.51225954
k-neighbor score = 0.6585

LO : 002
score = 0.0
score = 0.0
score = 0.17075318
score = 0.17075318
score = 0.17075318
k-neighbor score = 0.8975

LO : 003
score = 0.17075318
score = 0.34150636
score = 0.34150636
score = 0.34150636
score = 0.34150636
k-neighbor score = 0.6926

...
Learner ID >> 001
LO_id = 001 >> 0.6585
LO_id = 002 >> 0.8975
LO_id = 003 >> 0.6926
LO_id = 004 >> 0.6426
LO_id = 005 >> 0.6585

Learner ID >> 002
LO_id = 001 >> 0.9317
LO_id = 002 >> 0.6926
LO_id = 003 >> 0.7268
LO_id = 004 >> 0.5219
LO_id = 005 >> 0.6585

Learner ID >> 003
LO_id = 001 >> 0.9158
LO_id = 002 >> 0.6926
LO_id = 003 >> 0.8292
LO_id = 004 >> 0.7268
LO_id = 005 >> 0.7951

...
```

Figure5.3: The example output of NB-CF algorithm when value of k = 3.

Table 5.6: PE results of NBCF algorithm.

Number of Nearest Neighbors(k)	NB-CF PE			
	CS3 (N=31)	IT3 (N=48)	CS4 (N=29)	IT4 (N=34)
k=1	0.6774	0.5484	0.5517	0.4138
k=3	0.4194	**0.4516**	0.5172	0.3448
k=5	0.4194	**0.4516**	**0.4138**	**0.3103**
k=7	**0.3871**	0.5806	0.4483	0.3448
K=9	**0.3871**	**0.4516**	0.4483	0.3448

From the comparison of NB-CF PE result varied by value of k that shown in Table 5.6, the best results of this algorithm in each learner's groups are found when use k=5 and the worth PE results are found when we use k=1. The PE results seem to be as same as in AriMean in some cases (such as the PE result of IT3 learners), but the NB-CF gives better result in general.

5.4 The Comparison of Proposed Algorithms

To do final evaluations among proposed algorithms, the predictive results of each algorithm are compared against actual results. The comparisons of average PE result among recommendation algorithms are shown in Table 5.7.

Table 5.7: The comparison of evaluation results of every algorithm.

Algorithm	Variation	PE				Average PE	Std.	Rank
		CS3	IT3	CS4	IT4			
Rand	-	0.8670	0.8203	0.8190	0.8051	0.8279	0.0270	9
AriMean	-	0.3871	0.4792	0.5172	0.3824	0.4415	0.0673	7
PFB	Non weighting Feature (NWF)	0.2903	0.2917	0.2759	0.3235	0.2954	0.0201	2
	Weighted Feature (WF)	**0.2258**	**0.2083**	**0.2414**	**0.2353**	**0.2277**	**0.0144**	1
NBCF	k=1	0.6774	0.5484	0.5517	0.4138	0.5478	0.1077	8
	k=3	0.4194	0.4516	0.5172	0.3448	0.4333	0.0717	5
	k=5	0.4194	0.4516	0.4138	0.3103	0.3988	0.0613	3
	k=7	0.3871	0.5806	0.4483	0.3448	0.4402	0.1028	6
	k=9	0.3871	0.4516	0.4483	0.3448	0.4080	0.0515	4

Figure 5.4 shows that WF-PFB algorithm has the lowest PE, about 0.2083 in group of CS4 learners, and also has the lowest average PE about 0.2277 when compared with other algorithms.

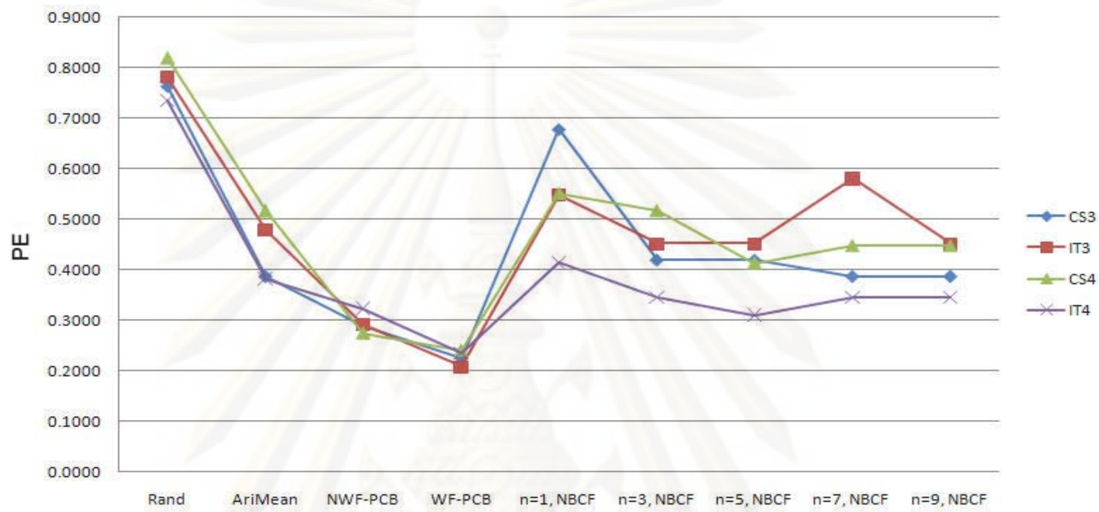


Figure 5.4: Tend of PE of all recommendation algorithms over the same learner group.

As the result shown in Figure 5.5, we note that all learning object recommendations have the same tend of PE in every learner group.

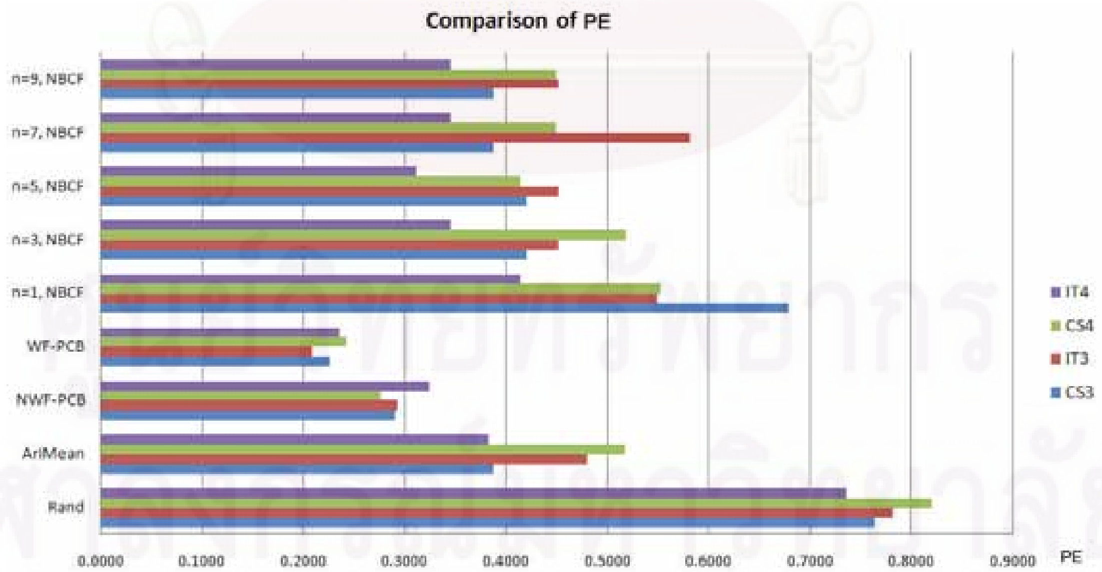


Figure 5.5: Comparison of PE results of all algorithms.

The comparison of PE result in Figure 5.5 presents PE result of all proposed algorithms. For this experiment, we can conclude that the WF-PFB algorithm is the highest accuracy algorithm follow by NFW-PFB, NB-CF and the worth is Rand algorithm.

5.5 Discussion

In this subsection, different recommendation algorithms mentioned above are discussed and analyzed. Five learning objects are personalized candidates which preferred by four group of participate learners in our domain. The two non-personalized algorithms, Rand and AriMean algorithms for learning object recommendation, do not give good results in general, especially Rand algorithm give the worst performance (predict with average PE = 0.8279). Among the various recommendation algorithms studied in this thesis, Weighted Feature-PFB is quite good accuracy in every group of learners (predict with average PE of about 0.2277). Therefore, using Weighted-Feature-PFB algorithm has the highest PE for this evaluation learner data in our domain.

CHAPTER VI

CONCLUSION

In this chapter the work conducted throughout this thesis are summarized and discussed. The first subsection reviews the research results obtained and highlights the main contributions. Next, in section 6.2, the limitations of this work are discussed. Finally, section points towards future work, identifying further research perspectives.

6.1. Synthesis of Main Results

The research is started with a comprehensive literature review, related to adaptive educational hypermedia in general, recommendation system and learning style-based adaptation in particular. We design the model based on multi agent-based, it is strongly based on a continuous interaction among involved agents: such an activity is facilitated by the choice of XML for both representing agent ontologies and handling data exchange. Next we tried to answer the 4 main research objectives that we proposed at the beginning of this thesis.

Firstly, the concept map combination model based on correlation computation can solve the different designs of various instructors and filter the uncompatible learning concept to the group of learners. To evaluate this methodology, Cronbac's α coefficient was used to test the reliability level. The result shows that proposed model has a strongly reliability with α=0.7459. The rest three objectives are related to the recommendation methodology.

Secondly, the five learning object features that will be used to form the learning object mapping rules by using the opinions of learners are identified. There are consisting of format, interactivity type, interactivity level, semantic density and learning resource type.

Then we can describe the learning object in thesis with five identified feature in form of learning object set (LOS).

Thirdly, based on the advantages of learning style those are described as follows:

- Instructor or course developer with an understanding of learners' learning styles are better able to adapt their teaching methods or developing learning objects appropriately.
- Learning style is the implications for learning material preference via learner behaviors.
- Learners who learn about their own style become better learners, they achieve higher grades and have more positive attitudes about their studies, greater self-confidence, learning time reducing and more skill in applying their knowledge in courses.

Information about learning styles can serve as a guide to the design of learning environment that either match, or mismatch, learners' style, depending on whether the instructor's purpose. The assumption, learning style is related to the learning object selection, is proposed in this research and the learner model is created from the relationship between learning style and learning object feature.

Based on the learning object features and the result of learner preference analysis, we can create the learner model that consist of learner style preference set (LSS) and learner preference set (LPS). Both of LSS and LPS set will be used with the criterion in the learning object recommendation algorithms. However, the limitations of learning style approach that explained below are found.

The limitation of learning style seems to be a methodology or tool for discovering the actual learning styles. The results are the incomplete information of learners in learning style approach to define the actual suitable materials for them. Moreover, the learning styles can be changed in different learning environments and situations. However, this point can be solved by the adaptive learner model in adaptive hypermedia educational system. Although the learning style is limited by the reason that is shown above, the educators still believe the learning style can improve the learners' learning for better learning.

Finally, the three approaches of learning object and its variation will be proposed for comparing the preference error (PE) result. To do final evaluation, we found that the preferred feature-based algorithm with weighted feature variation (WF-PFB) has the highest PE result and following by non-weight feature variation (NWF-PFB), Neighbor-based Collaborative Filtering (NB-CF) with k=5. The two non-personalized algorithms seem to be the worth performance, especially Rand algorithm.

As the results in all of research objectives, they give us to know what important process that learning object recommendation need. Being able to identify the learning style of the student is an important step, since it can be used to raise students' awareness regarding their strengths and weaknesses in learning as well as give instructors valuable information regarding the learning preferences they should try to accommodate in their course. In the context of research, learning style diagnosis is the prerequisite for adaptation provisioning. Then, the efficiency learning object recommendation was used to help us for providing the most compatible learning object to learner. As we provide both of content-based and collaborative filtering techniques, so the cold-start problem was solved. Finally, the efficiency of proposed model was proved experimentally, the accuracy of students satisfy is quite high.

To do the learning object recommendation methods that we proposed in this thesis, multi agent-based model was proposed. It provides the design of each agent, modules and databases. LMS developers can use this architecture to implement the complete system.

6.2. Limitations

Limitations of this thesis are represented in three groups of main work as follows:

- **Concept map combination model :**
  - This model is used to build the map of suitable concepts to learner by using only instructor opinions. The learner requirement is not analyzed.
- **Learner modeling:**
  - The only criterion for creating the learner model in proposed model is learning style. In order to allow for generalization, the modeling and adaptation methods should be tested on a wider scale, with user of variable age, field of study, background knowledge and technical experience. However this is a limitation that most studies in the e-learning area suffer from.
  - The styles of learning are scoped by Felder and Silverman model.
  - To demonstrate the learner model building, we use only five learning object features: format, interactivity type, interactivity level, semantic density and learning resource type. The candidate learning object must be filled five complete metadata for supporting the preference score calculation.
  - Mapping rule generation with word analysis technique is done by manual operation and the validation still requires an expert.
- **Learning object recommendation :**
  - In experiment, only five learning object are considered. It does not cover every learning object value space of all features. More learning objects should be defined for more accuracy testing result.

- Fields of study of participants are Computer Science and Information Technology.
- The web-based system is shown in prototype stage, so it does not cover all processes in proposed model.

Furthermore, the laboratory settings could be seen as a limitation. When learners know they are observed, they might alter their normal answer. However, it should be noted that learners were not aware of the purpose or expected outcome of experiment, so it is unlikely that they deliberately tried to confirm researcher's expectations. Nevertheless, it would be interesting to conduct the experiments with undergraduate students in other university, with students learning in the different environment.

6.3. Research Perspectives

In order to allow for such a large scale use of this learning object recommendation, repeating experiment in specific domain should repeat for longer period of time, with the larger number of learners with different background and knowledge levels, and in different study domains. This research is currently at prototype stage, being dedicated mainly to research proposes. It could be extended by adding more tools and functionalities borrowed from LMS, such as: more advanced communication and collaboration tools (as learner surveys suggested), learner involvement tools (student portfolio, bookmarks, etc).

Further support could also be provided for the instructor/ author: while a dedicated course editor is already included, and import / export facility, allowing for conversion between various course formats and standards (e.g. SCORM, IMS LD etc) would be very helpful. It would allow teachers to use exiting courses as they are, which would provide for greater use.

The currently use for analyzing learner is only their learning style. It is outside the scope of this thesis to deal with various learning scenarios but as future work we suggest to analyze the way of learner behavior such as learner interest, time of use, learner action, etc. The mapping rules are created by manual operation. It is better if the developer uses the ontology to do this process automatically or uses the semantic web technology.

To support the self-learning in adaptive hypermedia system, the learning object recommendation from this research can be combined with the course sequencing methodology in order to support the personalization of learning object selection and instructional planning of learners.

The findings and results obtained in the thesis open up many research perspectives for adaptive educational hypermedia system field and learning object development in particular. We believe these future directions to be worthwhile endeavors, since throughout this thesis we showed that we both can and should use learning styles in adaptive web based educational system.

REFERENCES

ADL SCORM. Shareable Content Objects Reference Model (SCORM). Available from: http:www.adlnet.gov/scorm/index.aspx [2008, January 27].

Adomavicius, G. and Tuzhilin, A. Toward the Next Generation of Recommendation System: a Survey of the state-of-art and possible extension. IEEE Transactions on Knowledge and Data Engineering 17(6) (June, 2005): 734-749.

Alderman, F. L. and Barritt, C. Creating a Reusable Learning Objects Strategy: Leveraging Information and Learning in a Knowledge Economy. Pfeifer Publishing, 2002.

Allinson, C. and Hayes, J. The Congnitive Style Index. Journal of Management Studies 33 (1996): 119-135.

Apter, M. J. Motivation Styles in Everyday Life: a Guide to Reversal Theory. Washington DC: American Psychological Association (2001): 87-109.

ATutor. ATutor [Online] . Available from: http://www.atutor.ca [2008, January 20].

Backboard. Backboard [Online]. Available from: http://www.blackbord [2008, February 3].

Balabanovic, M. and Shoham, Y. FAB: Content-based Collaborative Recommendation. Communications of the ACM 40(3) (1997): 66-72.

Bates, J. The role of emotion in believable agents. Communications of the ACM37(7) (1994):122-125.

Beshuizen, J. J. and Stoutjesdijk, E. T. Study Strategies in a Computer Assisted Study Environment. Learning and Instruction 9 (1999): 281-301.

Bresse, J. S., Heckerman, D., Kadie, C. Empirical Analysis of Predictive Algorithm for Collaborative Filtering. Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI' 98) (1998): 43-52.

Brown, E., Brailsford, T., Fisher, T., Moore, A. and Ashman, H. Reappraising Cognitive Styles in Adaptive Web Applications. Proceedings of the 18th Conference on Hypertext and Hypermedia, ACM, New York (2007): 57-66.

Brusilovsky, P. Adaptive Hypermedia. User Modeling and User-Adapted Interaction 11 (2001): 87-110.

Brusilosky, P., Farzan, R. and Ahn, J. Comprehensive Personalize Information Access in an Educational Digital Library. Proceedings of IEEE/ACM Joint Conference on Digital Libraries (2005): 9-18.

Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4):331-370.

CanCore. CanCore Guidelines Version 1.9: Educational Category. [Online]. Available from: http://www.cancore.ca/ guidelines/1.9/CanCore_guidelines_Educational_1.9.pdf. http://www.atutor.ca [2009, September].

Chen, L. & Sycara, K. WebMate: Personal Agent for Browsing and Searching. Proceedings of the Second International Conference on Autonomous Agents, ACM Press (1998):132-139.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D and Sartin, M. Combining Content-based and Collaborative Filters in an Online Newspaper. ACM SIGIR Workshop on Recommender Systems, Berkeley, 1999.

Cronbach, L. J. Coefficient aphla and the internal structure of tests. Psychometrika 16 (1981): 297-334.

Coffield, F., Moseley, D., Hall, E. and Ecclestone, K. Learning Styles and Pedagogy in Post-16 Learning. A Systematic and Critical Review. Learning and Skills Research Centre, UK (2004): 54-64.

Cox, D. and Greenberg, S. Supporting Collaborative Interpretation in Distributed Groupware. Proceedings of ACM Conference on Computer Supported Collaborative Work (2000): 289-298.

Curry, L. An Organization of Learning Styles Theory and Constructs. Annual Meeting of the American Educational Research Association. (1983).

Dahlen, B. J. , Konstan, J., Herlocker, J., Good, N. Borachers, A., and Riedl, J. Jump-Starting MovieLens: User Benefits of Starting a Collaborative Filtering System with 'dead data', Technical Report, TR 98-017. University of Minnesota, 1998.

Davis, J. and Huttenlocher, D. Shared Annotation for Cooperative Learning. Proceedings of Computer Supported Collaborative Learning (CSCL'95) (1995): 84-88.

DCMI. Dublin Core Metada Initiative [Online] . Available from: http://dublincore.org [2008, January].

dotLRN. dotLRN LMS [Online] . Available from: http://dotlrn.org [2008, January 20].

Dunn, R. and Griggs, S. Synthesis of the Dunn and Dunn Learning Styles Model Research: Who, What, When, Where and So What- The Dunn and Dunn Learning Styles Model and Its Theoretical Cornerstone. New York. St John's University, 2003.

Educause. [Online] . Available from: http://www.educause.edu [2007, November 15].

El-Beltagy, S. An Agent Based Framework for Navigation Assistance and Information Finding in Context. PhD thesis, University of Southampton, 2001.

Entwistle, N. J. Improving Teaching through Research on Student Learning, University teaching: international perspectives. New York, 1998.

Felder, R. M. and Silverman, L. K. Learning and Teaching Styles in Engineering Education. Engineering Education 78 (7)(1988) : 674-681.

Felder, R.M. and Soloman, B. A. Index of Learning Styles [Online]. Available from: http:www.ncsu.edu/felder-public/ILSpage.html. [2007, June].

Felder, R. M. and Spurlin, J. Application, Reliability and Validity of the Index of Learning Styles. International Journal on Engineering Educational 21(1) (2005): 103-112.

FIPA. The Foundation for Intelligent Physical Agents (FIPA) [Online]. Available from: http://www.fipa.org/ [2007, February 2].

Finin, T., Labrou, Y. & Mayfield, J. KQML as an Agent Communication Language. MIT Press, 1997: 291-316.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. Combining Collaborative Filtering with Personal Agents for Better Recommendations. Proceedings of IEEE/ACM Joint Conference on Digital Libraries (ACM/IEEE JCDL'03) (1999): 439-446.

Graf, S., Rita Viola, S, Leo, T. In-Depth Analysis of the Felder-Silverman Learning Style Dimensions. Journal of Research on Technology in Education 40(1) (2007):79-93.

Gregorc, A. F. Learning/Teaching Styles: Potent Forces behind Them. Educational Leadership 36(4) (1979).

Gregorc, A. F. Style Delineator: A Self-Assessment Instrument for Adults. Columbia. Gregorc Associates Inc., 1985.

Han, H., Giles, C. L., Manavoglu, E. and Zha, H. Automatic Document Metadata Extraction using Support Vector Machines. Proceedings of 2001 Canadian Artificial Intelligence Conference (2003): 37-48.

Herlocker, J., Konstan, J., Terveen, L. and Riedl, J. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information System (TOIS) 22(1) (January, 2004): 5-53.

Herlocker, J. and Konstan, J. An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of the 22th Annual International ACM SIGIR, Berkeley (1999): 230-237.

Herrmann, N. The Whole Brain Business Book. McGraw-Hill, 1996.

Hewitt, C. Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence, 8(3) (1977): 323-364.

Honey, P. and Mumford, A. The Learning Styles Helper's Guide. Maidenhead: Peter Honey Publications Ltd., 2000.

IBM. Aglets [Online]. Available from: http://www.trl.ibm.co.jp/aglets/ [2007, May 14].

IEEE LOM. IEEE Learning Object Metadata [Online]. Available from: http://ltsc.ieee.org/wg1 [2008, January 20].

IMS. Learner Information Package (LIP) Final Specification v1.0. IMS Global Learning Consortium Inc., 2001.

Jackson, C. Manual of the Learning Styles Profiler, 2002.

JADE. Java Agent Development Framework [Online]. Available from: http://jade.tilab.com/ [2008, February 3].

Jennings, N. R. On Agent-Based Software Engineering. Artificial Intelligence 117(2)(2000): 277-296.

Kay, J. Learner Control. User Modeling and User Adapted Interaction 11 (2001): 111-127.

Keefe, J. Learning Style: An Overview. NASSP's Student Learning Styles: Diagnosing and Prescribing Programs (2003): 1-17.

Keefe, J. and Ferrell, B. Developing a Defensible Learning Style Paradigm. Educational Leadership 48(2) (2002): 28-35.

Kolb, D. A. The Kolb Learning Style Inventory, Version 3. Hay Group, Boston, 1990.

KQML. The Knowledge Query and Manipulation Language (KQML). Available from: http://www.cs.umbc.edu/kqml/ [2007, February 22].

Langley, P., Iba, W. andThompson, K. An Analysis of Bayesian Classifiers. Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), AAAI Press/MIT Press (1992):223-228.

Leake, D., Maguitman, A., Reichherzer, T. Understanding Knowledge Models: Modeling Assessment of Concept Importance in Concept Maps. Proceedings of CogSci 2004. Mahwah, NJ: Erlbaum. In press, 2004.

Lawrence, S., Giles, C. L., and Bollacker, K. Digital Libraries and Autonomous Citation Indexing. IEEE Computer 32(6) (1999): 67-71.

Maes, P. Agents that Reduce Work and Information Overload. Communications of the ACM 37(7) (1994): 31-40.

MASIF. The Mobile Agent System Interoperability Facility (MASIF) [Online]. Available from: http://www.fokus.gmd.de/research/cc/ecco/masif/ [2007, February 22].

McKay, B. D., Royle, G. F., Wanless, I. M., Oggier, F. E., Sloane, N. J. A.; and Wilf, H. Acyclic Digraphs and Eigenvalues of (0,1)-Matrices. Journal of Integer Sequences 7, Article 04.3.3 1-5 (2004).

McNee, S., Riedl, J. and J. A. Konstan. Being Accurate is Not Enough: how Accuracy Metric have hurt Recommendation Systems. The Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems (CHI 2006) (2006): 1097-1101.

Melville, P., Mooney, R. J. and Nagarajan. R. Content-boosted Collaborative Filtering for improved Recommendations. Proceedings of National Conference on Artificial Intelligence (AAAI' 2002) (2002): 187-192.

Mladenic, D. Personal WebWatcher: Implementation and Design. Report Number: IJS-DP-7472, Department of Intelligent Systems, J.Stefan Institute, Slovenia, 1996.

Mooney, R. J. and Roy, L. Content-based Book Recommending using Learning for Text Categorization. Proceedings of IEEE/ACM Joint Conference on Digital Libraries (ACM/IEEE JCDL' 2000) (2000): 195-204.

Moodle. Moodle LMS [Online]. Available from: http://moodle.org [2008, January 20].

Myer, I. B. and McCaulley, M. H. Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator. Palo Alto, CA: Consulting Psychologists Press, 1985.

Nwana, H. Software Agents. Knowledge Engineering Review 11(3)(1996):1-40.

Novak, J. and Caňas, A. The Theory Underlying Concept Maps and How to Construct them, Technical Report IHMC CmapTools 2006-01. Florida Institute for Human and Machine Cognition Revised December 12, 2006.

Object Space. ObjectSpace Voyager 2.0, User Guide [Online]. Available from http://www.objectspace.com/developers/voyager/ [2007, May 14].

Pananikolaou, K. A., Mabbott, A., Bull, S. and Grigoriadou, M. Designing Learner Controlled Educational Interactions Based on Learning/Cognitive Style and Learner Behaviors. Interacting with Computer 18 (2006):438-442.

Pazzani, M. and Billsus, D. Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning 27 (1997): 313-331.

Pukkhem, N., Vatanawood, W. Using Ontological Modeling in Multi-Expert Guiding based Learning Object Recommendation. Proceedings of International Conference on Computer Engineering and Application. (June, 2009):73-77.

Recker, M., Walker, A. and Lawless K. What Do You Recommended? Implementation and Analyses of Collaborative Information Filtering of Web Resources for Education. Instructional Science 31 (2003): 299-316.

Riding, R. J. and Rayner, R. J. Cognitive Styles and Learning Strategies: Understanding Style Differences in Learning and Behavior. David Fulton Publishers, 1998.

Riedl, J. and Dourish, P. Introduction to the Special Section on Recommender Systems. ACM Transactions on Computer-Human Interaction 12(3) (2005): 371-313.

Ritter, A. L. Combining Two Contrasting Philosophies to Instructional Design [Online]. Available from: http://www.syllabus.com/news_issue.asp?id=154&IssueDate=8/18/2004#view [2007, September].

Roure, D. D., Hall, W., Reich, S., Hill, G., Pikrakis, A. and Stairmand, M. MEMOIR - An Open Framework for Enhanced Navigation of Distributed Information. Information Processing and Management 37 (2001): 53-74.

Sangineto, E., Capuano N., Gaeta M., and Micarelli, A. Adaptive Course Generation through Learning Styles Representation. Journal of Universal Access in the Information Society. Springer (2007): 223-240.

Schein, A., Popescul, A., Ungar, L. H. and Pennock, D. CROC: A New Evaluation Criterion for Recommender Systems. Electronic Commerce Research, Special Issue: World Wide Web Electronic Commerce, Security and Privacy 5(1) (2002): 253-260.

Sebastiani, F., Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1) (2001):1-47.

Shyu, H. Y., Hsieh, S. H. and Chou,Y. H. Integrating Concept Map into Designing a Courseware Management System. Proceedings of the Learning Objects 2003 Symposium. Honolulu, Hawaii, USA, (June 2003): 51-60.

Sternberg, R. J. Thinking Styles. Cambridge: Cambridge University Press, 1999.

Sumner, T., Khoo, M., Recker, M. and Marlino, M. Understanding Educator's Perceptions of 'Quality' in Digital Libraries. Proceedings of IEEE/ACM Joint Conference on Digital Libraries (ACM/IEEE JCDL' 2003) (2003): 269-279.

Tang, T. Y., and McCalla, G.I. Paper Annotations with Learner Models. Proceeding of the 12th International Conference on Artificial Intelligence in Education (2005): 654-661.

Triantafillou, E., Pomportsis, A. and Demetriadis, S. The design and the formative evaluation of an adaptive educational system based on cognitive styles. Computers & Education 41 (2003): 87-103.

Vermunt, J. D. The Regulation of Constructive Learning Processes. British Journal of Educational Psychology 68 (1998): 149-171.

Wagner, D. N. Liberal Order for Software Agents? An economic analysis. Journal of Artificial Societies and Social Simulation 3(1) (2000).

Weibel, S. The Dublin Core: a Simple Content Description Format for Electronic Resources. NFAIS Newsletter 40(7): 117-119.

Wikitionnary. Wikitionnary [Online]. Available from: http://www.wiktionary.org/ [2009, September].

Wiley, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy," in D. A. Wiley, ed., The Instructional Use of Learning Object (2002).

Woodruff, A., Gossweiler, R., Pitkow, J., Chi, E. and Card, S. Enhancing a Digital Book with a Reading Recomender. Proceedings of ACM Conference on Human Factors in Computing Systems (ACM CHI'00) (2000): 153-160.

Zywno, M.S. A Contribution to Validation of Score Meaning for Felder-Soloman's Index of Learning Styles. Proceedings of the 2003 American Society for Engineering Annual Conference and Exposition (2003): 78-84.

APPENDICES

APPENDIX A
QUESTIONNAIRES

A.1 Index of Learning Styles Questionnaire (Thai Version)

ชื่อ-นามสกุล...เพศ ☐ช / ☐ญ
สาขาวิชา...................................ชั้นปี.....

แบบทดสอบเพื่อจำแนกลักษณะการเรียนรู้ (Index of Learning Styles Questionnaire*)

คำแนะนำ : กรุณาวงกลมรอบคำตอบเพียงหนึ่งคำตอบ **(ก)** หรือ **(ข)** ที่บ่งบอกถึงลักษณะของท่านมากที่สุดในแต่ละคำถาม และให้ตอบคำถามทุกข้อ โดยคำถามมีทั้งหมด 44 ข้อ

1. ข้าพเจ้าจะเข้าใจสิ่งต่าง ๆ ได้ดีขึ้นเมื่อ
   (ก) ทดลองปฏิบัติ
   (ข) ได้ใช้เวลาคิดพิจารณาไตร่ตรอง

2 . ข้าพเจ้าอยากจะถูกมองว่าเป็นคนที่
   (ก) อยู่กับความจริง
   (ข) มีความล้ำสมัย/มีความคิดสร้างสรรค์

3 . เมื่อข้าพเจ้าได้ย้อนคิดถึงสิ่งที่ได้ทำไปเมื่อวาน ข้าพเจ้ามักจะคิดออกมาในรูปแบบของ
   (ก) ภาพเหตุการณ์
   (ข) คำพูด

4. ข้าพเจ้ามีแนวโน้มที่
   (ก) เข้าใจในรายละเอียดของแต่ละหัวข้อแต่มักจะคลุมเคลือในเรื่องของภาพรวมทั้งหมด
   (ข) เข้าใจภาพรวมทั้งหมดแต่มักจะคลุมเคลือในเรื่องของรายละเอียดในแต่ละหัวข้อ

5 . เมื่อข้าพเจ้าได้เรียนรู้สิ่งใหม่ ๆ มันจะช่วยข้าพเจ้าให้สามารถ
   (ก) พูดคุยแลกเปลี่ยนความรู้ในเรื่องนี้กับคนอื่น
   (ข) คิดพิจารณาอย่างถี่ถ้วนในเรื่องนี้ด้วยตนเอง

6 . ถ้าข้าพเจ้าเป็นอาจารย์ ข้าพเจ้าต้องการที่จะสอนเนื้อหาโดยเน้น

(ก) การให้ได้คิดถึงหลักการข้อเท็จจริงและสถานการณ์ที่เกิดขึ้นในชีวิตจริง

(ข) การให้ได้คิดถึงแนวคิดและหลักทฤษฎี

7 . ข้าพเจ้าชอบที่จะรับรู้ข้อมูลใหม่ ๆ จาก

(ก) รูปภาพ ไดอะแกรม กราฟ หรือ แผนที่

(ข) การแนะนำในรูปแบบการเขียนหรือข้อมูลที่มีการบรรยายเป็นคำพูด

8 . เมื่อข้าพเจ้าเข้าใจ

(ก) ทุกส่วนประกอบของสิ่งนั้น ข้าพเจ้าก็จะเข้าใจสิ่งนั้นได้อย่างสมบูรณ์

(ข) สิ่งนั้นอย่างสมบูรณ์ ข้าพเจ้าก็จะรู้ว่าส่วนต่างๆของสิ่งนั้นประกอบกันอย่างไร

9 . ในการทำงานเป็นกลุ่มในปัญหาที่มีความยาก ข้าพเจ้าชอบที่จะ

(ก) มีส่วนร่วมและแสดงความคิดเห็น

(ข) นั่งด้านหลังกลุ่มและฟัง

10 . ข้าพเจ้าค้นพบว่ามันง่ายกว่า เมื่อข้าพเจ้า

(ก) เรียนรู้ข้อเท็จจริง

(ข) เรียนรู้แนวคิด

11 . ในหนังสือที่ประกอบไปด้วยรูปภาพ ชาร์ต (chart) จำนวนมาก ข้าพเจ้าชอบที่จะ

(ก) พิจารณารูปภาพและชาร์ตนั้นอย่างระมัดระวัง

(ข) เน้นการอ่านข้อความที่เขียนบรรยาย

12 . เมื่อข้าพเจ้าต้องการแก้ปัญหาทางคณิตศาสตร์

(ก) ข้าพเจ้ามักจะเลือกทำตามวิธีการของข้าพเจ้าเพียงขั้นตอนเดียว ณ เวลานั้น

(ข) ข้าพเจ้ามักจะมองหาหลาย ๆ วิธี จากนั้นต้องมาหาวิธีจัดการให้เป็นลำดับขั้นตอนเพื่อแก้ปัญหานั้น

13 . ในการเรียนในชั้นเรียน

(ก) ข้าพเจ้ามักจะรู้จักเพื่อนร่วมชั้นเรียนเป็นจำนวนมากอยู่เป็นประจำ

(ข) ข้าพเจ้ามักไม่ค่อยจะรู้จักเพื่อนร่วมชั้นเรียนมากนัก

14. ในการอ่านบทความที่ไม่ใช่นวนิยาย ข้าพเจ้าต้องการได้

(ก) บางสิ่งที่สอนข้าพเจ้าถึงข้อเท็จจริงใหม่ ๆ หรือบอกข้าพเจ้าถึงวิธีการทำสิ่งต่าง ๆ

(ข) บางสิ่งที่ให้ความคิดใหม่ๆให้ข้าพเจ้าได้พิจารณา

15 . ข้าพเจ้าชอบผู้สอนที่

(ก) สอนโดยการใช้ภาพไดอะแกรมหลาย ๆภาพบนกระดาน

(ข) ใช้เวลาส่วนใหญ่ในการอธิบายเป็นคำพูด

16 . เมื่อข้าพเจ้ากำลังวิเคราะห์เรื่องราวต่าง ๆ หรือ นิยายหรือหนังสืออ่านเล่น

(ก) ข้าพเจ้าคิดถึงเหตุการณ์ต่างๆและพยายามประมวลเข้าด้วยกันเพื่อหาแนวคิดของเรื่อง

(ข) ข้าพเจ้ารู้แนวคิดของเรื่องเมื่ออ่านจบและต้องย้อนกลับไปหาเหตุการณ์ต่างๆที่บ่งชี้แนวคิดของเรื่อง

17 . เมื่อข้าพเจ้าต้องการแก้โจทย์ของการบ้าน ข้าพเจ้ามักจะ

(ก) เริ่มด้วยการแก้ปัญหาทันที

(ข) พยายามทำความเข้าใจกับปัญญาอย่างถี่ถ้วนก่อน

18 . ข้าพเจ้าต้องการแนวคิดที่

(ก) เป็นสิ่งที่แน่นอน

(ข) เป็นทฤษฎี

19 . ข้าพเจ้าจะจดจำได้ดีที่สุดเมื่อ

(ก) ข้าพเจ้าได้เห็นอะไร

(ข) ข้าพเจ้าได้ยินอะไร

20 . มันเป็นสิ่งที่มีความสำคัญมากสำหรับข้าพเจ้า ถ้าผู้สอน

(ก) มีการวางโครงสร้างบทเรียนที่มีลำดับชัดเจน

(ข) ทำให้ข้าพเจ้ามองเห็นถึงภาพรวมและความสัมพันธ์กับเนื้อหาในวิชาอื่น ๆ

21 . ข้าพเจ้าชอบที่จะเรียนรู้แบบ

(ก) เป็นกลุ่ม

(ข) ตามลำพัง

22 . ข้าพเจ้าชอบที่จะถูกมองว่า

(ก) ระมัดระวังในรายละเอียดของงาน

(ข) คิดสร้างสรรค์ว่าจะพัฒนางานขึ้นมาอย่างไร

23 . เมื่อข้าพเจ้าต้องการคำแนะนำในการไปสถานที่ที่ไม่เคยไป ข้าพเจ้าต้องการ

(ก) แผนที่

(ข) การเขียนอธิบายการเดินทาง

24 . ข้าพเจ้าเรียนรู้

(ก) ในระดับปกติ ถ้าข้าพเจ้าเรียนอย่างหนัก ข้าพเจ้าก็จะทำได้

(ข) แบบขึ้นๆลงๆ ข้าพเจ้าจะสับสนก่อน และแล้วก็เรียนรู้ได้ในทันที

25 . ข้าพเจ้ามักจะทำเหตุการณ์นี้เป็นอันดับแรก ในการแก้ปัญหา

(ก) ทดลองทำจริง

(ข) คิดไตร่ตรองก่อนว่าควรจะทำอย่างไร

26 . เมื่อข้าพเจ้ากำลังอ่านงานเพื่อความบันเทิง ข้าพเจ้าต้องการให้ผู้เขียน

(ก) อธิบายถึงความหมายของสิ่งที่ต้องการสื่ออย่างชัดเจน

(ข) นำเสนอด้วยวิธีสร้างสรรค์และน่าสนใจ

27 . เมื่อข้าพเจ้าเห็นไดอะแกรมหรือมีการวาดรูปในชั้นเรียน ข้าพเจ้าชอบที่จะจดจำ

(ก) รูปภาพ

(ข) สิ่งที่ผู้สอนพูดหรืออธิบาย

28 . เมื่อพิจารณาเนื้อหาสาระของข้อมูลข้าพเจ้าชอบที่จะ

(ก) เน้นเรื่องรายละเอียดและข้ามภาพรวมไป

(ข) พยายามเข้าใจภาพรวมก่อนเข้าไปสู่รายละเอียด

29 . ข้าพเจ้าจะจดจำสิ่งต่าง ๆ ได้ง่ายขึ้นเมื่อสิ่งนั้น

(ก) เป็นสิ่งที่ข้าพเจ้าได้ทำแล้ว

(ข) เป็นสิ่งที่ข้าพเจ้าได้คิดทบทวนไตร่ตรองอย่างมากมาย

30 . เมื่อข้าพเจ้าลงมือทำงานใดข้าพเจ้าต้องการ

(ก) ทำงานให้สำเร็จด้วยวิธีที่ดีที่สุดวิธีเดียว

(ข) ทำงานให้สำเร็จด้วยวิธีใหม่ๆหลายวิธี

31 . เมื่อมีผู้ใดมานำแสดงข้อมูลแก่ข้าพเจ้า ข้าพเจ้าต้องการข้อมูลนั้นในรูปแบบ

(ก) ชาร์ท (chart) หรือ กราฟ

(ข) ข้อความที่เป็นผลสรุป

32 . เมื่อข้าพเจ้าเขียนบทความ ข้าพเจ้าชอบที่จะ

(ก) คิดหรือเขียนเรื่องตั้งแต่เริ่มต้นไปจนจบ

(ข) คิดหรือเขียนเรื่องในส่วนต่าง ๆ แล้วจึงนำส่วนต่าง ๆ ของเรื่องมาเรียงกัน

33 . เมื่อข้าพเจ้าต้องทำงานเป็นกลุ่ม ประการแรกข้าพเจ้าต้องการ

(ก) มีการ "ระดมความคิดแบบเป็นกลุ่ม" โดยทุกคนร่วมกันแสดงความคิดเห็น

(ข) ต่างคนต่างคิด แล้วหลังจากนั้นจึงค่อยจัดกลุ่มเอาความคิดนั้นมาเปรียบเทียบกัน

34 . ข้าพเจ้าคิดว่าคนลักษณะเช่นนี้สมควรได้รับการชื่นชม

(ก) มีเหตุผล

(ข) มีจินตนาการ

35 . เมื่อข้าพเจ้าได้พบผู้คนในงานเลี้ยงสังสรรค์ ข้าพเจ้ามักจะจดจำ

(ก) ลักษณะหน้าตาการแต่งตัวของเขา

(ข) สิ่งที่พวกเขาพูดเกี่ยวกับตัวเขาเอง

36 . เมื่อข้าพเจ้ากำลังเรียนรู้รายวิชาใหม่ ข้าพเจ้าาชอบที่จะ

(ก) มุ่งเน้นที่รายวิชานั้น และเรียนรู้ให้มากที่สุดเท่าที่ทำได้

(ข) พยายามสร้างการเชื่อมต่อระหว่างรายวิชานั้นกับรายวิชาอื่น ๆ ที่เกี่ยวข้อง

37 . ข้าพเจ้าอยากจะถูกมองว่าเป็นคนที่

(ก) เข้าสังคมเก่ง

(ข) สงวนท่าที

38 . ข้าพเจ้าชอบรายวิชาที่ให้ความสำคัญหรือเน้น

(ก) เนื้อหาที่เป็นรูปธรรม (ข้อเท็จจริง, ข้อมูล)

(ข) เนื้อหาที่เป็นนามธรรม (แนวคิด, ทฤษฎี)

39 . สำหรับการบันเทิง ข้าพเจ้าชอบที่จะ

(ก) ดูโทรทัศน์

(ข) อ่านหนังสือ

40 . ผู้สอนบางท่านเริ่มต้นบทเรียนโดยการให้โครงสร้างบทเรียนที่ครอบคลุมเนื้อหา โครงสร้างนั้น

(ก) เป็นตัวช่วยให้ข้าพเจ้าได้เรียนรู้บ้าง

(ข) เป็นตัวช่วยให้ข้าพเจ้าได้การเรียนรู้เป็นอย่างมาก

41 . แนวความคิดของการทำการบ้านเป็นกลุ่ม แล้วให้คะแนนเท่ากันทั้งกลุ่ม ข้าพเจ้าคิดว่า

(ก) ดึงดูดใจสำหรับข้าพเจ้า

(ข) ไม่ดึงดูดใจสำหรับข้าพเจ้า

42. เมื่อข้าพเจ้าต้องคำนวณโจทย์ปัญหายาว ๆ

(ก) ข้าพเจ้ามีแนวโน้มที่จะทบทวนทุกขั้นตอนและตรวจสอบงานของข้าพเจ้าอย่างระมัดระวัง

(ข) ข้าพเจ้าคิดว่าการตรวจสอบเป็นงานที่น่าเบื่อหน่ายและพยายามบังคับตนเองให้ทำ

43. ข้าพเจ้ามีแนวโน้มที่จะอธิบายสถานที่ต่าง ๆ ที่เคยไปได้อย่าง

(ก) ง่ายและค่อนข้างมีความแม่นยำ

(ข) ยากลำบากและไม่ค่อยมีรายละเอียด

44. เมื่อมีการแก้ปัญหาในกลุ่มข้าพเจ้ามักจะ

(ก) คิดถึงทุกขั้นตอนในกระบวนการแก้ปัญหา

(ข) คิดถึงผลลัพธ์ที่เป็นไปได้หรือการใช้วิธีการแก้ปัญหาในแนวกว้าง

---

* แบบสอบถามชุดนี้แปลจาก Richard M. Felder and Barbara A. Soloman, *Index of Learning Styles*, http://www.ncsu.edu/felder-public/ILSpage.html

A.2 Learner Rating for Learning Object Feature Selection Questionnaire

ชื่อ-นามสกุล...เพศ ☐ ช / ☐ ญ
สาขาวิชา...................................ชั้นปี.....

<u>คำอธิบาย</u>: ในสถานการณ์ของการเรียนรู้แบบออนไลน์ ท่านเห็นด้วยกับคุณลักษณะของวัตถุการเรียนรู้ (Learning object feature) ในการส่งผลต่อความชอบที่จะเลือกเรียนวัตถุการเรียนรู้ใด ๆ ของท่านเอง ในระดับเท่าใด โดย 5=เห็นด้วยอย่างยิ่ง, 4=เห็นด้วย, 3 = ค่อนข้างเห็นด้วย, 2 =ไม่เห็นด้วย และ 1= ไม่เห็นด้วยอย่างยิ่ง

กรุณาทำเครื่องหมาย X ในช่องของระดับที่ท่านต้องการในทุกหัวข้อ

คุณลักษณะของวัตถุการเรียนรู้ (Learning Object Feature)		1 ไม่เห็นด้วยอย่างยิ่ง	2 ไม่เห็นด้วย	3 ค่อนข้างเห็นด้วย	4 เห็นด้วย	5 เห็นด้วยอย่างยิ่ง
1. หมวดทั่วไป (General)	1.1 ชื่อวัตถุการเรียนรู้ (title) เช่น “การค้นหาแบบไบนารี่”					
	1.2 ภาษา (language) เช่น “Thai” , “Eng”					
	1.3 คำอธิบาย (description) : อธิบายวัตถุประสงค์การเรียนรู้					
	1.4 โครงสร้าง (structure) เช่น “แบบบรรยาย” “แบบชั้นลำดับ”					
	1.5 ความซับซ้อน (aggregation) : ระดับการรวมกันของวัตถุการเรียนรู้ชิ้นย่อยไปเป็นชิ้นใหญ่					
2. หมวดเทคนิค (Technical)	2.1 รูปแบบ (format): ประเภทของไฟล์ เช่น “วีดีโอ”, “รูปภาพ”					
	2.2 ขนาด (size) : ขนาดไฟล์ของวัตถุการเรียนรู้					
	2.3 แหล่งจัดเก็บเพื่อเข้าไปเรียนรู้ (location) เช่น เว็บไซต์ผู้ให้บริการวัตถุการเรียนรู้					

คุณลักษณะของวัตถุการเรียนรู้ (Learning Object Feature)		1 ไม่เห็นด้วยอย่างยิ่ง	2 ไม่เห็นด้วย	3 ค่อนข้างเห็นด้วย	4 เห็นด้วย	5 เห็นด้วยอย่างยิ่ง
3. หมวดการศึกษา (Educational)	3.1 ประเภทของการปฏิสัมพันธ์กับผู้เรียน (interactivity type): เช่น “แบบสื่อผสม” , “แบบบรรยาย”					
	3.2 ประเภทของวัตถุการเรียนรู้ (learning resource type) : เช่น “อัลกอริทึม”, “สไลด์”, “แบบจำลอง”, “คำจำกัดความ”, “แบบฝึกหัด”					
	3.3 ระดับของการปฏิสัมพันธ์กับผู้เรียน (interactive level): ระดับของการให้ผู้เรียนมีส่วนร่วมกับการเรียนรู้ เช่น การตอบคำถาม การเปลี่ยนแปลงค่าพารามิเตอร์ การโต้ตอบกับวัตถุการเรียนรู้					
	3.4 ระดับการสื่อความหมายของวัตถุการเรียนรู้ (semantic density) : เช่น “สื่อความหมายได้ดี”, “สื่อความหมายได้ปานกลาง”					
	3.5 บริบท (context) : สภาพแวดล้อมโดยรวมของวัตถุการเรียนรู้					
	3.6 ระดับความยาก (difficulty) : ระดับของความยากง่ายของวัตถุการเรียนรู้					
	3.7 ความดังของเสียง (audio loudness)					
	3.8 ความสว่างของสี (color brightness)					
	3.9 ความซับซ้อนของการใช้สี (color complexity)					
	3.10 รายละเอียดของการใช้เสียง (detail of sound) : เช่น ความชัดเจนของเสียง					
	3.12 รายละเอียดของข้อความ (detail of text): เช่น มีการอธิบายข้อความอย่างละเอียด					

คุณลักษณะของวัตถุการเรียนรู้ (Learning Object Feature)		1 ไม่เห็นด้วยอย่างยิ่ง	2 ไม่เห็นด้วย	3 ค่อนข้างเห็นด้วย	4 เห็นด้วย	5 เห็นด้วยอย่างยิ่ง
	3.13 รายละเอียดของประโยค (detail of sentence) : เช่น เป็นประโยคที่มีการอธิบายความแบบสมบูรณ์					

*************ขอขอบคุณนิสิตทุกท่านที่ให้ความร่วมมือในการตอบแบบสอบถาม***************

A.3 Instructor's opinion for the concept map combination reliability questionnaire

Major ………………………………………………….. Teaching Experience ……..years

Please write **X** on the level of agreement that you think about the concept map combination methodology (see the detail in attached paper).

Question	1 Very disagree	2 disagree	3 Neutral	4 agree	5 Very Agree
Q1. Do you agree concept map combination methodology on the quality of the e-learning system?					
Q2. Do you agree various designs of instructor should be considered in concept map in e-learning system?					
Q3. Do you agree the concept map combination model can be used to personalize the learning object selection?					
Q4. Refer to the quality concept map development; do you agree the total quality combination model is helpful for instructors and users to develop his/her course?					
Q5. Do you agree the proposed concept map combination methodologies are complete?					
Q6. Do you agree this proposed methodology can be used as a preprocessing for improving quality of a learning object recommendation system?					
Q7. Do you agree the proposed concept map combination should provide in learning management system?					

APPENDIX B

Extended Experimental Results

B.1 Extended Learning Object Feature Rating Results

Table B.1: The rating results of learning object feature selection by 31 learners.

LO Feature	Leaner ID																															∑	Normalized
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Title	2	1	1	2	3	4	1	2	2	1	4	2	1	1	2	3	2	2	3	2	1	1	2	2	2	1	3	4	5	2	3	67	0.4323
Language	4	5	5	5	5	4	5	5	5	4	5	4	5	5	5	4	5	5	5	4	5	5	3	4	3	3	5	5	4	4	5	140	0.9032
Description	5	4	3	5	5	4	2	3	4	5	5	4	5	5	4	4	4	4	3	5	3	3	3	4	2	2	3	4	4	4	4	119	0.7677
Structure	3	3	2	1	4	1	1	2	3	1	5	4	1	2	2	2	3	4	5	5	2	3	3	2	1	1	2	3	4	5	3	83	0.5355
Aggregation	1	1	2	2	2	3	2	3	3	3	4	1	1	2	3	2	3	3	2	2	1	1	1	2	2	3	2	2	1	3	4	67	0.4323
Format	5	5	5	5	5	4	5	3	5	5	5	5	3	4	5	3	4	5	5	5	2	5	5	4	4	4	5	2	2	2	3	129	0.8323
Size	1	1	1	2	2	3	3	2	5	4	2	2	2	1	1	1	1	2	2	2	1	2	1	2	1	2	1	3	4	1	2	60	0.3871
Location	1	2	1	3	1	1	3	1	2	1	2	4	4	1	1	2	2	2	3	2	2	2	1	2	3	2	1	1	2	2	4	61	0.3935
Interactivity Type	5	4	2	3	4	5	4	3	5	3	3	3	4	2	2	3	3	5	5	4	2	3	4	5	5	4	4	4	4	2	2	111	0.7161
Learning Resource Type	4	5	5	5	4	5	4	5	5	5	5	5	5	5	5	5	4	5	5	5	4	5	4	5	5	5	4	4	5	3	3	143	0.9226
Interactivity Level	4	5	4	3	5	3	2	5	4	2	2	2	3	2	4	5	4	3	5	3	3	3	3	2	4	5	5	3	2	5	4	109	0.7032
Semantic Density	2	2	3	3	5	4	4	4	5	4	3	5	3	2	3	2	3	3	3	4	4	4	4	3	5	3	5	4	5	5	3	112	0.7226
Context	1	3	1	2	1	2	4	4	1	2	2	1	2	1	2	1	3	2	2	3	2	2	1	3	3	3	4	2	2	4	3	69	0.4452
Difficulty	1	2	3	1	5	4	1	2	2	2	2	3	2	4	5	2	2	3	3	5	2	5	5	4	3	3	3	4	1	1	2	87	0.5613
Auditory Loudness	4	3	5	3	3	3	4	2	2	3	1	1	2	3	2	3	3	2	2	2	2	3	2	3	3	2	4	5	5	3	4	89	0.5742
Color Brightness	2	1	3	2	2	3	5	3	3	3	4	2	5	4	2	2	2	1	2	2	2	3	2	4	2	3	3	5	5	4	4	90	0.5806
Color Complexity	2	3	3	2	5	3	4	2	2	3	5	4	2	2	2	1	2	3	2	4	5	2	3	3	3	4	1	2	2	3	2	86	0.5548
DetailOfSound	5	4	3	5	3	3	1	2	4	5	4	2	2	2	1	3	3	3	3	2	3	3	3	1	1	3	1	2	2	3	2	84	0.5419
DetailOfText	3	2	5	4	2	2	2	1	3	2	2	3	2	3	3	2	2	2	3	3	4	2	3	4	2	2	2	5	4	4	2	85	0.5484
DetailOfSentense	3	3	3	4	4	2	3	3	2	5	3	2	3	2	2	3	3	2	5	4	2	2	3	4	5	3	2	3	3	3	5	96	0.6194

B.2 Extended Non-weighting Feature Preferred feature-based (NWF-PFB) Results

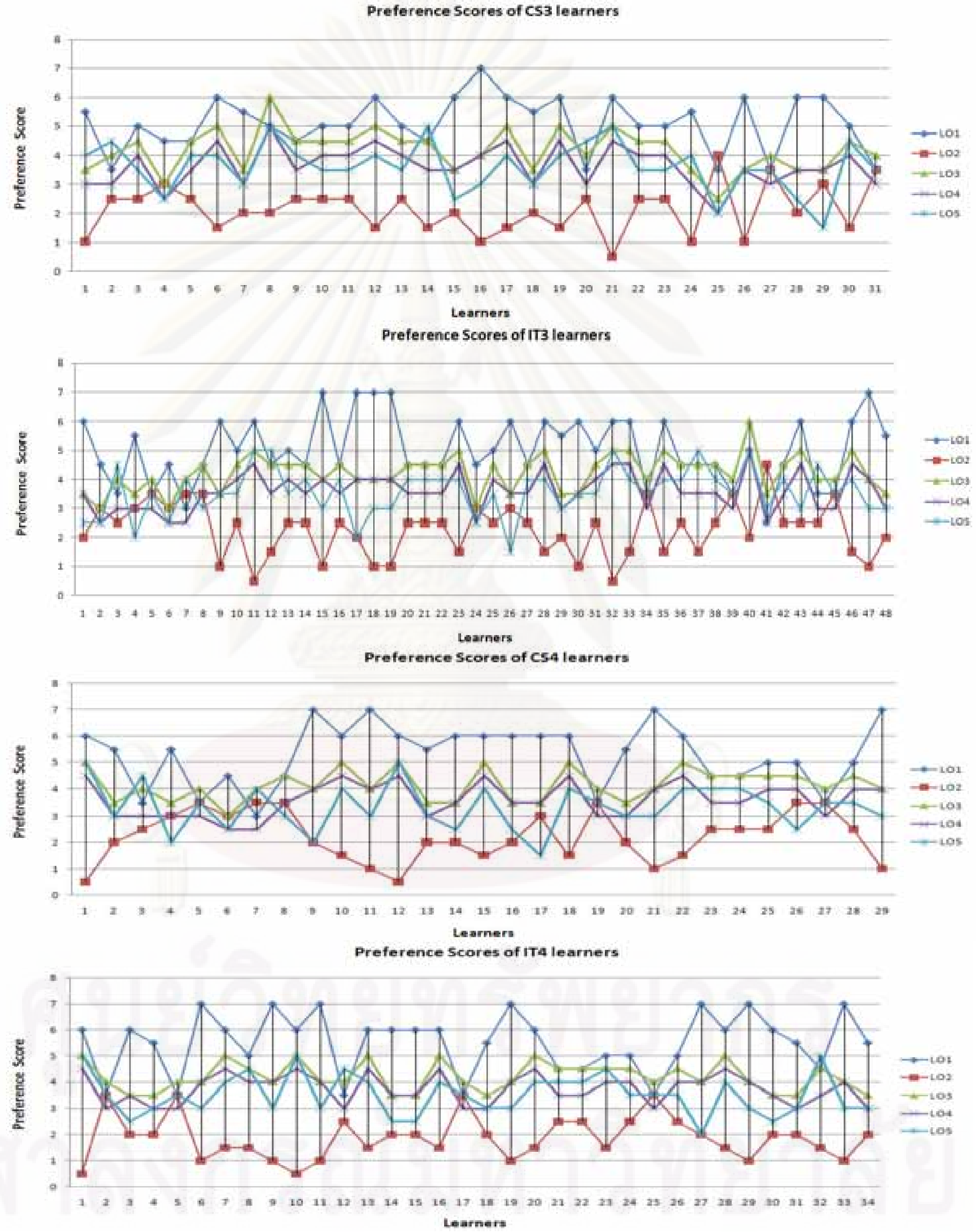


Figure B.1: The non–weighting feature preference scores of learners categorized by major.

B.3 Extended Weighted Feature Preferred feature-based (WF-PFB) Results

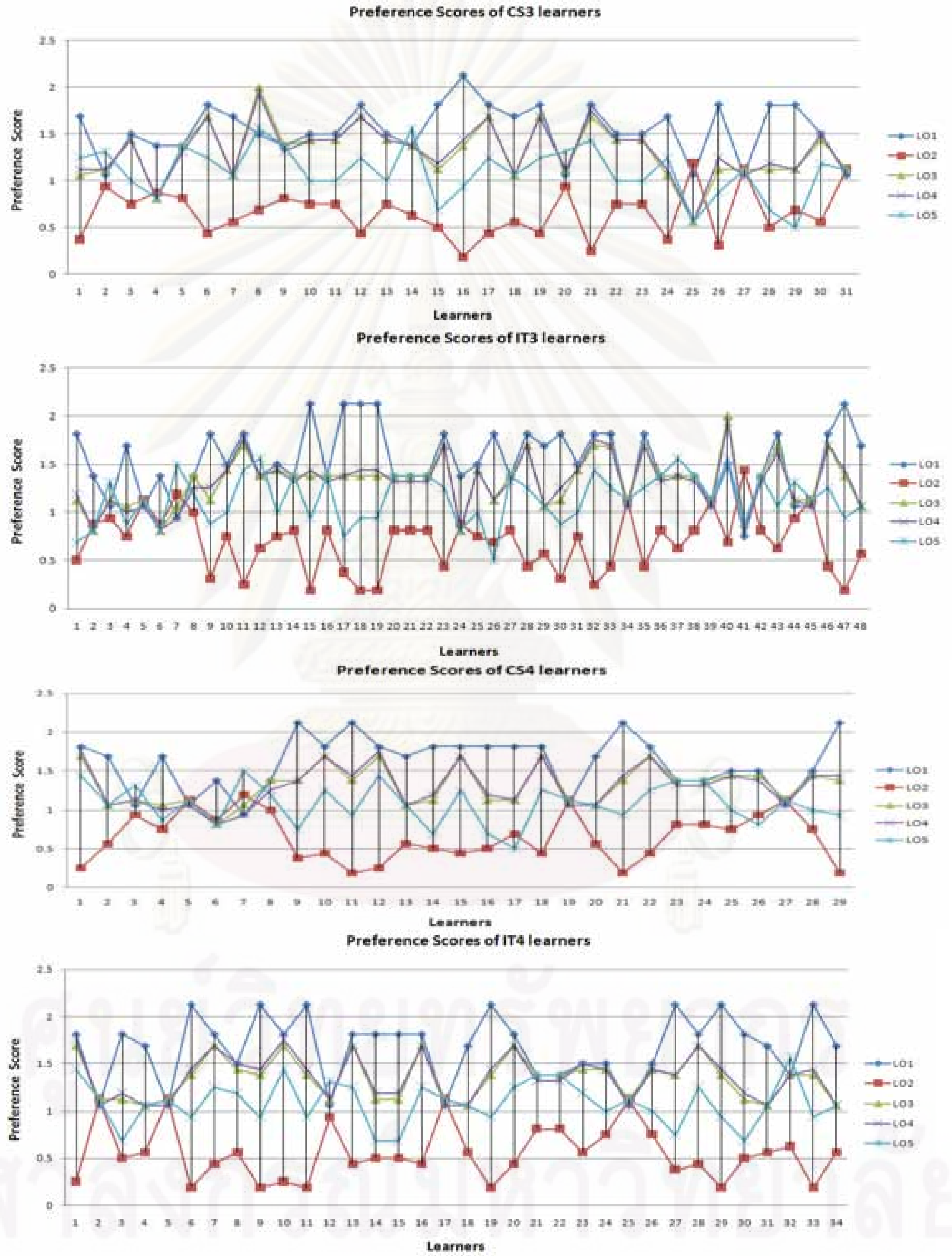


Figure B.2: The weighted feature preference scores of learners categorized by major.

B.4 Extended Neighbor-based Algorithm Results

Table B.2: The Results of Neighbor-based algorithm classified by learner's major.

NBCF-CS3

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
1	3	1	3	1	2	2	1	3	1	1	1	1	1	1	0	0	0	0	0
2	2	2	2	2	2	2	1	3	1	3	1	1	1	1	1	0	0	0	0
3	2	2	2	2	3	1	2	2	5	1	1	1	1	1	1	1	1	1	1
4	2	2	3	1	2	2	2	2	1	2	1	5	1	1	1	0	1	0	0
5	3	1	2	2	2	2	2	2	3	1	1	1	1	1	1	1	1	1	1
6	3	1	2	2	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
7	3	1	3	1	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
8	3	1	1	3	3	1	2	2	3	3	1	1	1	1	0	1	1	1	1
9	3	1	2	2	2	2	2	2	1	3	1	3	1	1	1	0	1	0	0
10	2	2	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0
11	2	2	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0
12	3	1	2	2	3	1	2	2	1	3	1	1	1	1	1	0	0	0	0
13	2	2	2	2	3	1	2	2	3	5	1	1	1	1	1	1	1	1	1
14	3	1	2	2	2	2	1	3	5	1	1	1	1	1	1	1	1	1	1
15	2	2	3	1	3	1	2	2	2	5	1	1	1	1	1	1	1	1	1
16	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
17	3	1	2	2	3	1	2	2	1	3	1	1	1	1	1	0	0	0	0
18	3	1	3	1	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
19	3	1	2	2	3	1	2	2	1	3	3	1	1	1	1	1	0	0	0
20	2	2	2	2	2	2	1	3	3	1	1	1	1	1	1	1	1	1	1
21	3	1	2	2	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0
22	2	2	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0

173

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
23	2	2	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0
24	3	1	3	1	2	2	1	3	1	1	1	1	1	1	0	0	0	0	0
25	1	3	3	1	2	2	2	2	2	1	1	5	5	1	1	1	1	1	1
26	2	2	3	1	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0
27	2	2	2	2	2	2	2	2	5	1	1	5	1	1	1	1	0	1	1
28	2	2	3	1	3	1	2	2	3	2	1	1	1	1	1	1	1	1	1
29	2	2	3	1	3	1	3	1	1	2	1	1	1	1	1	0	0	0	0
30	2	2	2	2	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0
31	2	2	2	2	2	2	2	2	5	5	1	1	1	1	0	1	1	1	1
														PE	0.6774	0.4194	0.4194	0.3871	0.3871

NBCF-IT3

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
1	2	2	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
2	2	2	3	1	2	2	2	2	5	1	2	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	1	3	5	5	2	3	2	2	0	1	1	1	1
4	3	1	3	1	2	2	3	1	1	3	1	1	1	1	1	0	0	0	0
5	2	2	2	2	2	2	2	2	2	3	2	3	1	1	1	0	1	1	1
6	2	2	3	1	2	2	2	2	1	5	3	1	1	1	1	1	0	0	0
7	3	1	2	2	1	3	2	2	3	3	1	1	1	1	0	1	1	1	1
8	3	1	2	2	2	2	3	1	3	3	1	1	1	1	0	1	1	1	1
9	2	2	3	1	3	1	1	3	2	1	1	1	1	1	1	1	1	1	1
10	2	2	2	2	3	1	2	2	1	1	1	3	1	1	0	0	1	0	0
11	3	1	2	2	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0

174

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
12	3	1	2	2	2	2	1	3	3	3	1	1	1	5	0	1	1	1	1
13	2	2	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
14	3	1	2	2	2	2	2	2	3	2	1	1	3	1	1	1	1	0	1
15	3	1	3	1	3	1	2	2	1	4	1	1	1	1	1	0	0	0	0
16	3	1	2	2	2	2	2	2	2	3	1	1	3	1	1	1	1	1	1
17	3	1	3	1	3	1	3	1	1	1	1	1	1	1	0	0	0	0	0
18	3	1	3	1	3	1	2	2	4	1	1	1	1	1	1	1	1	1	1
19	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
20	3	1	2	2	2	2	2	2	1	3	3	1	3	1	1	1	0	1	0
21	3	1	2	2	2	2	2	2	5	3	1	1	3	1	1	1	1	1	1
22	3	1	2	2	2	2	2	2	4	3	1	1	1	1	1	1	1	1	1
23	3	1	2	2	3	1	2	2	1	3	1	1	1	1	1	0	0	0	0
24	2	2	3	1	2	2	2	2	1	5	3	1	1	1	1	1	0	0	0
25	2	2	2	2	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
26	2	2	3	1	3	1	3	1	1	1	1	1	1	1	0	0	0	0	0
27	3	1	2	2	2	2	2	2	3	3	1	1	1	1	0	1	1	1	1
28	3	1	2	2	3	1	2	2	2	1	1	1	1	1	1	1	1	1	1
29	3	1	3	1	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
30	2	2	3	1	3	1	1	3	1	2	1	1	1	1	1	0	0	0	0
31	2	2	2	2	3	1	2	2	1	1	1	3	1	1	0	0	1	0	0
32	3	1	2	2	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0
33	3	1	2	2	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
34	2	2	2	2	2	2	2	2	3	2	2	2	1	2	1	1	1	1	1
35	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
36	3	1	2	2	2	2	2	2	1	3	1	1	3	1	1	0	0	1	0

175

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
37	3	1	2	2	2	2	1	3	5	3	1	1	1	1	1	1	1	1	1
38	3	1	2	2	2	2	2	2	1	3	1	1	3	1	1	0	0	1	0
39	2	2	2	2	2	2	2	2	2	2	2	2	1	1	0	0	0	1	1
40	3	1	1	3	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
41	1	3	2	2	2	2	2	2	2	2	2	2	1	2	0	0	0	1	0
42	3	1	2	2	2	2	2	2	4	3	1	1	1	1	1	1	1	1	1
43	3	1	2	2	3	1	3	1	1	1	1	1	1	1	0	0	0	0	0
44	2	2	2	2	2	2	1	3	5	3	2	2	2	2	1	1	1	1	1
45	2	2	2	2	2	2	2	2	2	2	2	3	1	1	0	0	1	1	1
46	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
47	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
48	3	1	3	1	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
														PE	0.5484	0.4516	0.4516	0.5806	0.4516

NBCF-CS4

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
1	3	1	3	1	2	2	1	1	5	1	1	1	0	1	0	0	0	1	3
2	3	1	2	2	2	2	5	5	1	1	1	1	0	1	1	1	1	2	3
3	2	2	2	2	1	3	5	3	1	1	1	1	1	1	1	1	1	2	2
4	3	1	2	2	3	1	4	1	1	1	1	1	1	1	1	1	1	1	3
5	2	2	2	2	2	2	5	3	3	1	1	1	1	1	1	1	1	2	2
6	3	1	2	2	2	2	1	5	5	1	1	1	1	1	0	0	0	2	3
7	2	2	1	3	2	2	3	1	1	3	1	1	1	1	0	1	1	1	2
8	2	2	1	3	3	1	1	1	1	3	1	1	0	0	1	0	0	1	2

176

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
9	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	2	2
10	3	1	3	1	3	1	3	1	1	1	1	1	1	1	1	1	1	2	3
11	2	2	2	2	2	2	1	3	1	1	1	1	1	0	0	0	0	1	2
12	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	1	2
13	3	1	2	2	2	2	1	5	5	1	1	1	1	1	0	0	0	1	3
14	3	1	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0	3	3
15	2	2	3	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	2
16	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	2	3
17	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	2	3
18	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	1	2
19	2	2	2	2	2	2	3	3	1	1	1	1	0	1	1	1	1	2	2
20	3	1	2	2	2	2	5	1	5	1	1	1	1	0	1	1	1	1	3
21	3	1	3	1	2	2	4	1	1	1	1	1	1	1	1	1	1	1	3
22	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0	1	2
23	2	2	2	2	2	2	3	1	1	1	1	1	1	1	1	1	1	1	2
24	2	2	2	2	2	2	4	1	1	1	1	1	1	1	1	1	1	1	2
25	2	2	3	1	1	3	1	1	1	1	1	1	0	0	0	0	0	2	2
26	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	2	2
27	2	2	2	2	2	2	3	1	1	3	1	1	1	1	0	1	1	2	2
28	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0	2	2
29	3	1	3	1	3	1	1	5	1	1	1	1	1	0	0	0	0	1	3
														PE	0.5517	0.5172	0.4138	0.4483	0.4483

177

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
3	2	2	3	1	3	1	2	2	1	3	1	1	1	1	1	0	0	0	0
4	3	1	3	1	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
5	2	2	2	2	2	2	2	2	5	2	1	2	1	1	1	1	1	1	1
6	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
7	3	1	2	2	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
8	2	2	2	2	3	1	1	3	1	1	5	1	1	1	0	1	0	0	0
9	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
10	3	1	2	2	3	1	1	3	1	5	1	1	1	1	1	0	0	0	0
11	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
12	2	2	2	2	2	2	1	3	5	2	5	2	1	1	1	0	1	1	1
13	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
14	2	2	3	1	3	1	2	2	3	1	1	1	1	1	1	1	1	1	1
15	2	2	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
16	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
17	2	2	2	2	2	2	2	2	2	2	5	5	1	1	0	1	1	1	1
18	3	1	3	1	2	2	2	2	1	5	1	1	1	1	1	0	0	0	0
19	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
20	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
21	3	1	2	2	2	2	2	2	4	1	3	1	1	1	1	1	1	1	1
22	3	1	2	2	2	2	2	2	3	4	1	1	1	1	1	1	1	1	1
23	2	2	2	2	3	1	1	3	1	1	5	1	1	1	0	1	0	0	0
24	2	2	2	2	3	1	2	2	1	5	1	1	1	1	1	0	0	0	0
25	2	2	2	2	2	2	2	2	3	2	5	5	1	1	1	1	1	1	1
26	2	2	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
27	3	1	3	1	3	1	3	1	1	1	1	1	1	1	0	0	0	0	0

Table B.2: The Results of Neighbor-based algorithm classified by learner's major. (Cont.)

LID	A_L	R_L	S_L	I_L	U_L	B_L	Q_L	G_L	Actual Preferred LO	NBCF-Predictive LO					Preference Error				
										k=1	k=3	k=5	k=7	k=9	k=1	k=3	k=5	k=7	k=9
28	3	1	2	2	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
29	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
30	2	2	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
31	3	1	3	1	2	2	2	2	5	1	1	1	1	1	1	1	1	1	1
32	3	1	2	2	2	2	1	3	5	4	5	5	1	1	1	0	0	1	1
33	3	1	3	1	3	1	2	2	1	1	1	1	1	1	0	0	0	0	0
34	3	1	3	1	2	2	2	2	4	1	1	1	1	1	1	1	1	1	1
														PE	**0.4138**	**0.3448**	**0.3103**	**0.3448**	**0.3448**

179

APPENDIX C

System Prototype

C.1 Web-based Learning Object Recommendation

In order to demonstrate the learning object recommendation based on learning styles, we implement a web-based learning object recommendation system called "LOS: Learning Style-based Learning Object Recommendation", which assists the learner in process of learning style assessment and learning object recommendation.

The system was implemented as a web-based tool, using JSP and XML, XML DOM technologies based on Struts 1.2.9 framework and GlassFish V2 as application server.

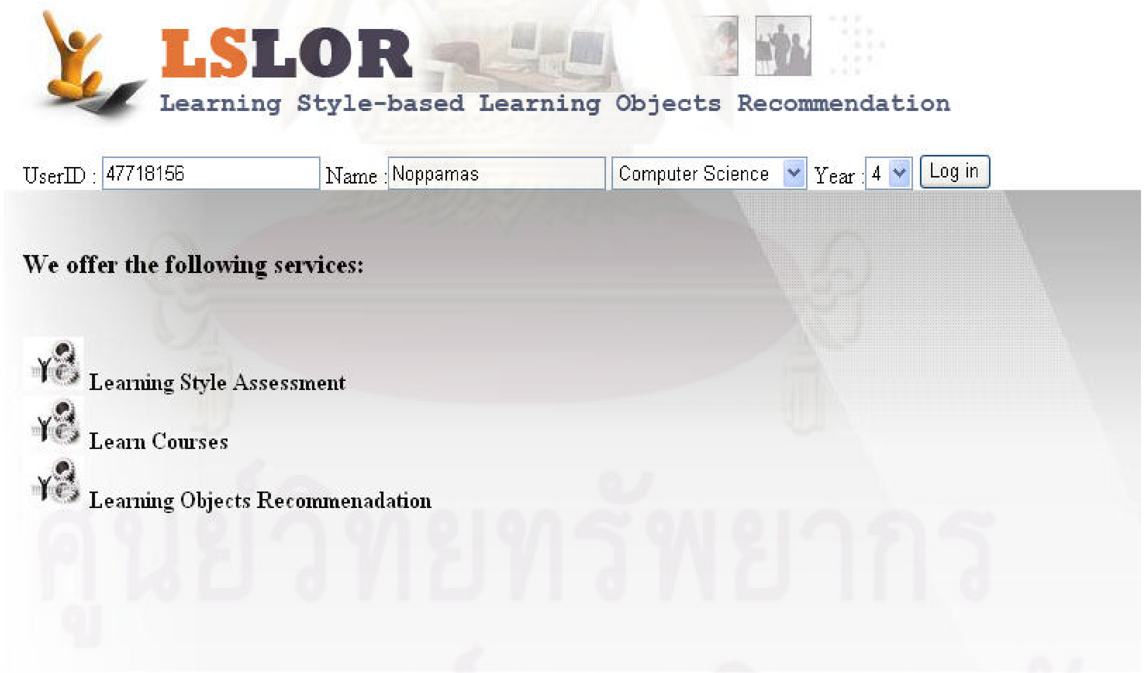


Figure C.1: A snapshot of the LSLOR – Login page.

After logging into the system (Figure C.1), the new learner is offered to answer the 44-questions of Index of Learning Style questionnaire (Figure C.2). The results of learning style are shown in Figure C.3.

Figure C.2: A snapshot of the LSLOR – Learning style assessment page.

Figure C.3: A snapshot of the LSLOR – Learning style scores results.

The preference scores are shown in eight preferences based on type of learning styles: Active, reflective, sensing, intuitive, visual, verbal, sequential and global. Once this process is finished, a learner preference set (LPS) base on learners' learning style will be created. Next, if the learner request to learner the course, learn course link in navigate window is selected and the select course page is activated. Figure C.4 shows the course selection process to provide available course in the system to learner.

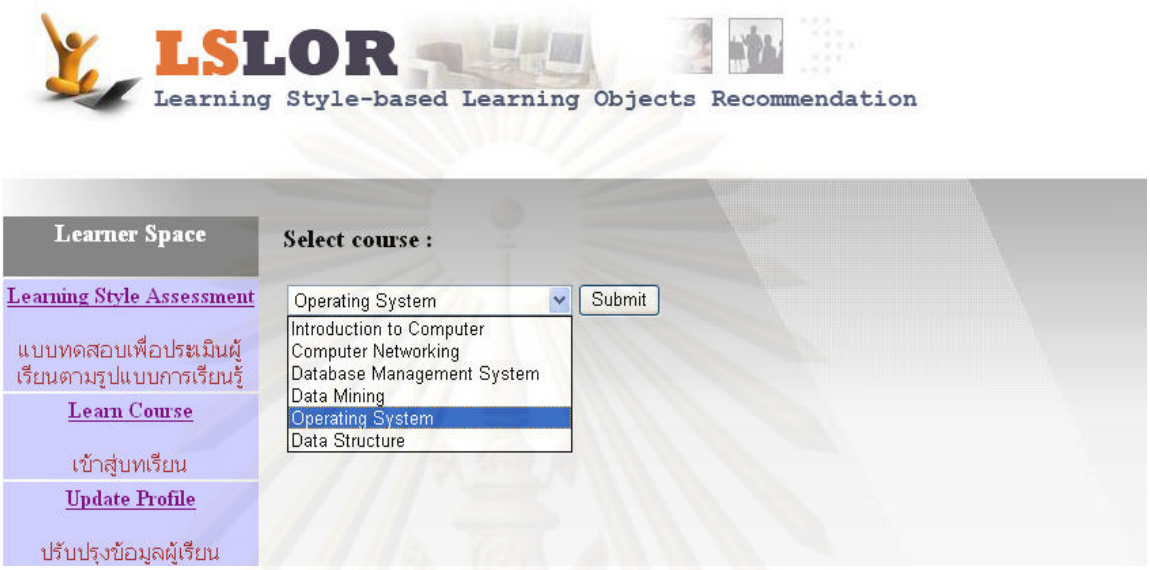


Figure C.4: A snapshot of the LSLOR – Select course.

Next, the learner may choose the lesson (Figure C.5) and views all topics in selected lesson (Figure C.6).

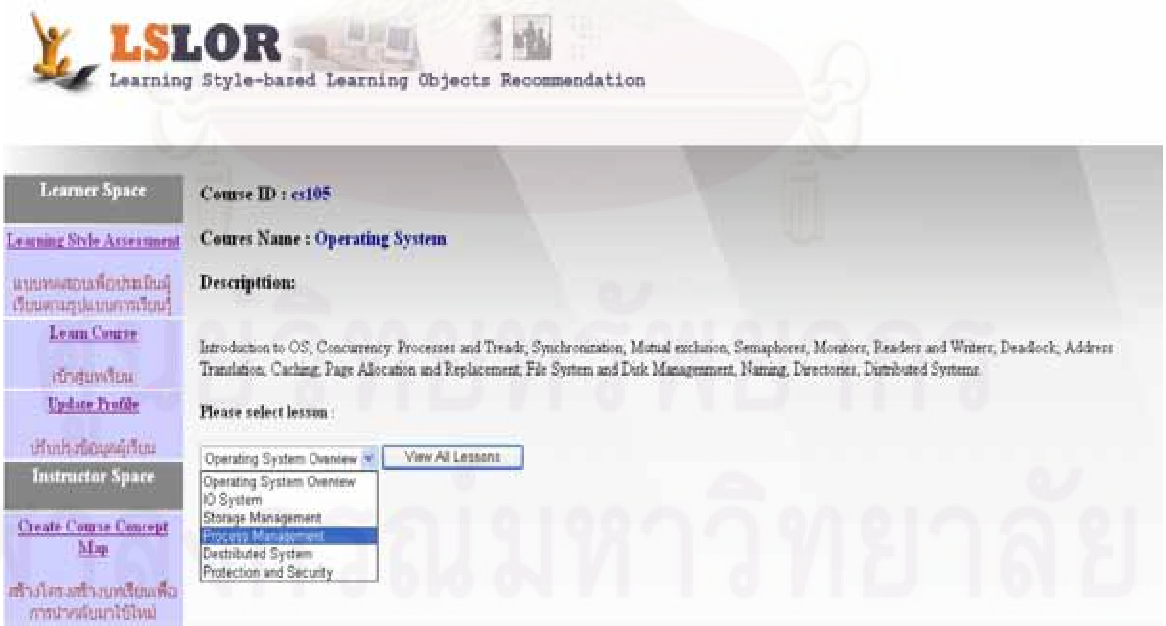


Figure C.5: A snapshot of the LSLOR – Select lesson.

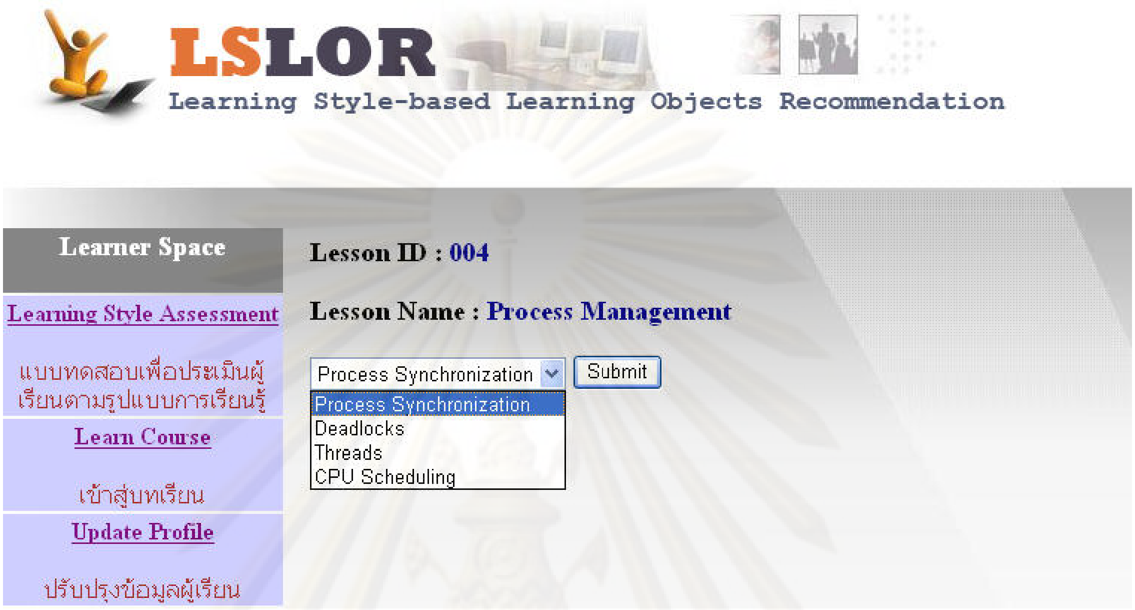


Figure C.6: A snapshot of the LSLOR – Select topic.

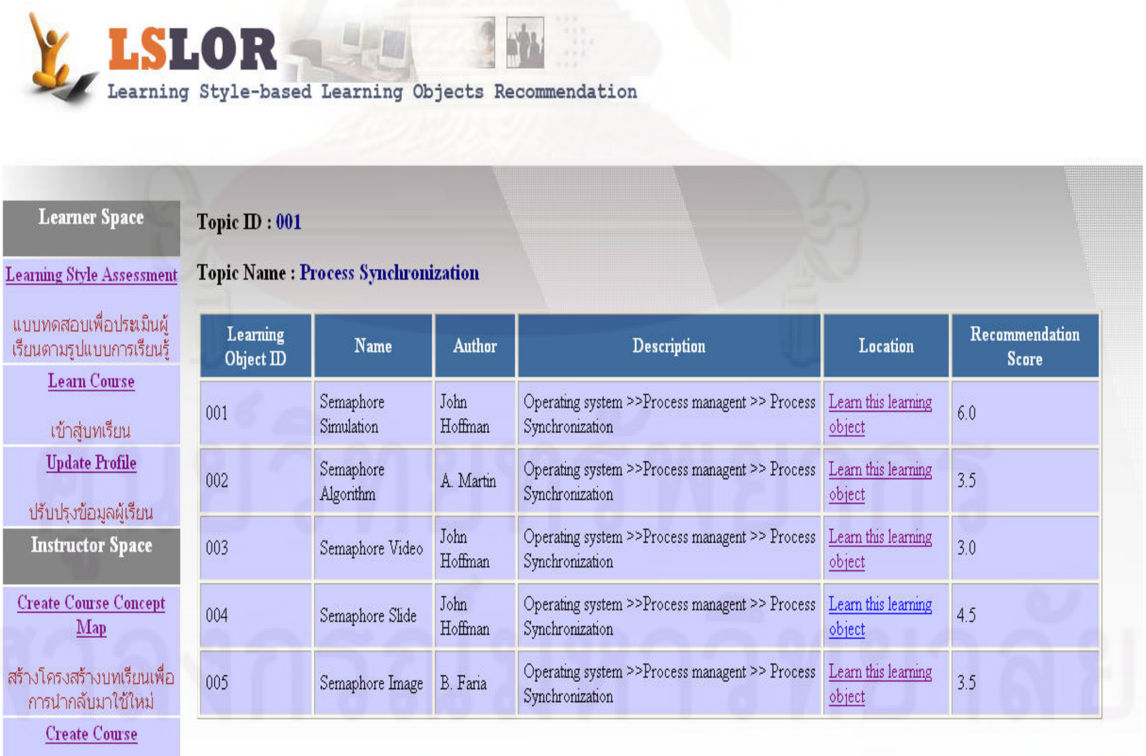


Figure C.7: A snapshot of the LSLOR – Recommend learning objects.

After learner views all topics and submits the topic that he/she wants to learn, the learning objects recommendation page will be appeared (Figure C.7).

The highest preference score shows the most compatible learning object based on learners' learning styles. The location is provided the accessible way to learn the recommended learning object from the system (Figure C.8). In general of implementation, location of learning object may define as LMS courseware URL.

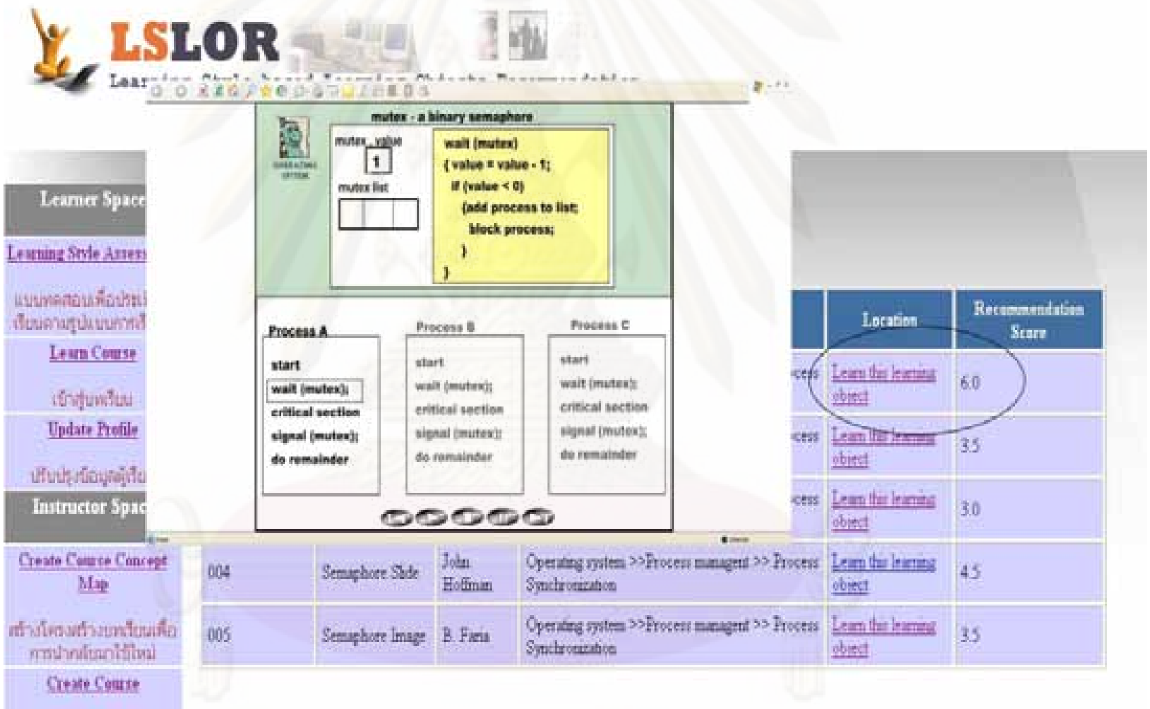


Figure C.8: A snapshot of the LSLOR – Link to learning object.

C.2 Lorecommend Package

```
/*
 * This package is used to extract the LOM features of learning object, convert LSS set to LPS set,
 * calculate preference score of learning object and to generate an output in XML format
 */

package lorecommend;

import java.util.Vector;
import java.io.File;
import java.io.FileWriter;
import org.dom4j.Document;
import org.dom4j.DocumentHelper;
import org.dom4j.DocumentException;
import org.dom4j.io.SAXReader;
import org.dom4j.Element;
import org.dom4j.Attribute;
import org.dom4j.io.XMLWriter;
import java.util.Iterator;

/**
 *
 * @author  Noppamas  Pukkhem
 */
public class FWPreferenceBase {

   Vector<LearnerStyle> learnerStyleVector = new Vector<LearnerStyle>();
   Vector<LearnerPreferSet> learnerPreferVector = new Vector<LearnerPreferSet>();
   Vector<LoM> loVector = new Vector<LoM>();
   Vector<LearnerPreferLo> loScoreVector = new Vector<LearnerPreferLo>();
   String learnerStyleFile;
   String lomFile;
   String loScoreFile;
   private Document doc;

   public void loadLearnerStyle(){

            try{
        File aFile = new File("C:\\lorecommend\\xml\\LearnerStyle.xml");
        SAXReader xmlReader = new SAXReader();
        this.doc = xmlReader.read(aFile);
        Element root = this.doc.getRootElement();
        Iterator elementIterator = root.elementIterator();
        while(elementIterator.hasNext()){
           Element element = (Element)elementIterator.next();
           Iterator learner = element.elementIterator();
           LearnerStyle ls = new LearnerStyle();

           Element id = (Element)learner.next();
           ls.setId((String)id.getData());
           Element name = (Element)learner.next();
           ls.setName((String)name.getData());
           Element major = (Element)learner.next();
```

```
            ls.setMajor((String)major.getData());
            Element year = (Element)learner.next();
            ls.setYear((String)year.getData());
            Element style = (Element)learner.next();
            Iterator attStyle = style.attributeIterator();

            int i = 0;
            while (attStyle.hasNext()){
               Attribute attr = (Attribute) attStyle.next();
               float score = Float.parseFloat((String) attr.getData());
               ls.setStyle(i, score);
               i++;
            }
            learnerStyleVector.add(ls);
         }
            }catch(DocumentException e){
                     e.printStackTrace();
            }
}

public void loadLoM(){
   // load lo xml --> vector

   try{
      File aFile = new File("C:\\lorecommend\\xml\\LO.xml");
      SAXReader xmlReader = new SAXReader();
      this.doc = xmlReader.read(aFile);
      doc.toString();
      Element root = this.doc.getRootElement();
      Iterator elementIterator = root.elementIterator();
      while(elementIterator.hasNext()){

         LoM lom = new LoM();

         Element element = (Element)elementIterator.next();
         Iterator lo = element.elementIterator();
         Attribute loAttr = element.attribute(0);
          lom.setId((String)loAttr.getData());

         Element name = (Element)lo.next();
         lom.setName((String)name.getData());
         Element author = (Element)lo.next();
         lom.setAuthor((String)author.getData());
         Element des = (Element)lo.next();
         lom.setDes((String)des.getData());

         Element lomRoot = (Element)lo.next();
         Iterator lomIt = lomRoot.elementIterator();

            Element techFormat = (Element) lomIt.next();
            Attribute tfAttr = techFormat.attribute(0);
            lom.setTechFormat((String)tfAttr.getData());
            Element interactType = (Element) lomIt.next();
            Iterator interactTypeAttr = interactType.attributeIterator();
            Attribute it = (Attribute) interactTypeAttr.next();
```

```
            lom.setInteractType((String) it.getData());
            Attribute il = (Attribute) interactTypeAttr.next();
            lom.setInteractLevel((String) il.getData());
            Attribute sd = (Attribute) interactTypeAttr.next();
            lom.setSemanticDens((String) sd.getData());
            Attribute rt = (Attribute) interactTypeAttr.next();
            lom.setResourceType((String) rt.getData());

            loVector.add(lom);
          }

   } catch(DocumentException e){
                  e.printStackTrace();
          }
}

public void p(Object m){
          System.out.println(m);
}

public void printLSV(){
   for (int i=0;i<learnerPreferVector.size();i++){
      LearnerPreferSet preferSet= new LearnerPreferSet();
          preferSet = (LearnerPreferSet)learnerPreferVector.get(i);
          p("Learner ID >> " + preferSet.getId());
          for (int j=0;j<8;j++){
                  if (preferSet.featureSet[j].getWeight()!=0){
                          for (int k=0;k<preferSet.featureSet[j].featureVector.size();k++){
                                  System.out.print(preferSet.featureSet[j].getFeature(k)+" : ");
                          }
                          p(preferSet.featureSet[j].getWeight());
                  }
          }
          p("======================");
   }
}

public void printLScV(){
   for (int i=0;i<loScoreVector.size();i++){
      LearnerPreferLo preferLo = new LearnerPreferLo();
          preferLo = (LearnerPreferLo)loScoreVector.get(i);
          p("Learner ID >> " + preferLo.getId());
          for (int j=0;j<preferLo.preferScore.size();j++){
         System.out.print("LO_id = " + preferLo.preferScore.elementAt(j).getId()+ " >> ");
         System.out.println(preferLo.getPreferScore(j));
          }
            p("");
          p("======================");
   }

}

public void applyRule(){
// module for convert the learner style set (LSS) to learner preference set (LPS)
```

```
//read LSS.xml
for (int i=0;i<learnerStyleVector.size();i++){
  LearnerStyle style = new LearnerStyle();
  LearnerPreferSet preferSet= new LearnerPreferSet();

  style = (LearnerStyle)learnerStyleVector.get(i);
  preferSet.setId(style.getId());
  if(style.getWeight(0)!=0){
    preferSet.setWeight(0, style.getWeight(0));
    preferSet.addFeature(0, "Active");
    preferSet.addFeature(0, "Mixed");
    preferSet.addFeature(0, "Execise");
    preferSet.addFeature(0, "Simulation");
    preferSet.addFeature(0, "Experiment");
  }
  if(style.getWeight(1)!=0){
    preferSet.setWeight(1, style.getWeight(1));
    preferSet.addFeature(1, "Expositive");
    preferSet.addFeature(1, "Definition");
    preferSet.addFeature(1, "Algorithm");
    preferSet.addFeature(1, "Example");
  }
  if(style.getWeight(2)!=0){
    preferSet.setWeight(2, style.getWeight(2));
    preferSet.addFeature(2, "8");
    preferSet.addFeature(2, "9");
    preferSet.addFeature(2, "Simulation");
    preferSet.addFeature(2, "Experiment");
  }
  if(style.getWeight(3)!=0){
    preferSet.setWeight(3, style.getWeight(3));
    preferSet.addFeature(3, "5");
    preferSet.addFeature(3, "6");
    preferSet.addFeature(3, "7");
    preferSet.addFeature(3, "Definition");
    preferSet.addFeature(3, "Example");
  }
  if(style.getWeight(4)!=0){
    preferSet.setWeight(4, style.getWeight(4));
    preferSet.addFeature(4, "Video");
    preferSet.addFeature(4, "Image");
    preferSet.addFeature(4, "Animation");
    preferSet.addFeature(4, "2");
    preferSet.addFeature(4, "3");
    preferSet.addFeature(4, "4");
    preferSet.addFeature(4, "Simulation");
  }
  if(style.getWeight(5)!=0){
    preferSet.setWeight(5, style.getWeight(5));
    preferSet.addFeature(5, "Text");
    preferSet.addFeature(5, "Audio");
    preferSet.addFeature(5, "0");
    preferSet.addFeature(5, "1");
    preferSet.addFeature(5, "Definition");
    preferSet.addFeature(5, "Exercise");
```

```
            }
            if(style.getWeight(6)!=0){
                preferSet.setWeight(6, style.getWeight(6));
                preferSet.addFeature(6, "Text");
                preferSet.addFeature(6, "Audio");
                preferSet.addFeature(6, "5");
                preferSet.addFeature(6, "6");
                preferSet.addFeature(6, "Exercise");
                preferSet.addFeature(6, "Algorithm");
                preferSet.addFeature(6, "Slide");
            }
            if(style.getWeight(7)!=0){
                preferSet.setWeight(7, style.getWeight(7));
                preferSet.addFeature(7, "Image");
                preferSet.addFeature(7, "Index");
            }
            // apply all rule
            learnerPreferVector.add(preferSet);
        }
        printLSV();        // apply rule from style vector to prefer vector
    }

    public void computeScore(){
        // compute score from prefer vector and lom vector to score vector
        for (int i=0;i<learnerPreferVector.size();i++){
            LearnerPreferSet preferSet = new LearnerPreferSet();
            preferSet = (LearnerPreferSet)learnerPreferVector.get(i);
            LearnerPreferLo preferLo = new LearnerPreferLo();
            preferLo.setId(preferSet.getId());
            for (int j=0;j<loVector.size();j++){
                LoM lom = new LoM();
                lom = (LoM)loVector.get(j);
                double loScore = 0;
                double  f1=0.25, f2=0.5, f3=0.5, f4=0.5, f5=0.125;
                for (int k=0;k<8;k++){

                    float weight = preferSet.featureSet[k].getWeight();
                    if(weight!=0.0){
                        for(int x=0;x<preferSet.featureSet[k].featureVector.size();x++){
                            String feature = preferSet.featureSet[k].getFeature(x);
                            if (lom.getTechFormat().equals(feature)){
                                loScore = loScore + f1*weight;
                            }
                            if (lom.getInteractType().equals(feature)){
                                loScore = loScore + f2*weight;
                            }
                            if (lom.getInteractLevel().equals(feature)){
                                loScore = loScore + f3*weight;
                            }
                            if (lom.getSemanticDens().equals(feature)){
                                loScore = loScore + f4*weight;
                            }
                            if (lom.getResourceType().equals(feature)){
                                loScore = loScore + f5*weight;
                            }
```

```
                }
              }
            }
            LoPreferScore preferScore = new LoPreferScore();
            preferScore.setId(lom.getId());
            preferScore.setScore(loScore);
            preferLo.addPreferScore(preferScore);
          }
          loScoreVector.add(preferLo);
        }
      }

      public void storeLoPreferScore(){
        Document document = DocumentHelper.createDocument();
        Element root = document.addElement( "PreferenceBaseScore" );

        for (int i=0;i<loScoreVector.size();i++){
          LearnerPreferLo preferLo = new LearnerPreferLo();
           preferLo = (LearnerPreferLo)loScoreVector.get(i);
          Element learner = root.addElement("learner");
          Element id = learner.addElement("id")
          .addText(preferLo.getId());
               p("Learner ID >> " + preferLo.getId());
          Element los = learner.addElement("los");

               for (int j=0;j<preferLo.preferScore.size();j++){
             Element lo = los.addElement("lo");
             Element loId = lo.addElement("LO_id")
                   .addText(preferLo.preferScore.elementAt(j).getId());
             System.out.print("LO_id = " + preferLo.preferScore.elementAt(j).getId()+ " >> ");
             Element pfScore = lo.addElement("PreferenceScore")
                   .addText(String.valueOf(preferLo.getPreferScore(j)));
             System.out.println(preferLo.getPreferScore(j));
              }
            p("");
        }
        try{
          XMLWriter writer = new XMLWriter(new FileWriter( "C:\\lorecommend\\xml\\FWPreferScoreOutput.xml" ));
          writer.write( document );
          writer.close();
          p("write file");
        }  catch (Exception e){
             e.printStackTrace();
        }
      }

      public void run(){
        this.loadLearnerStyle();
        this.loadLoM();
        this.applyRule();
        this.computeScore();
        this.storeLoPreferScore();
      }

    }
```

APPENDIX D

PUBLICATIONS

C.1 International Journals

1. Pukkhem, N., Vatanawood, W. (2009)" The Algorithms for Collaborative "Course Concept Map" Decision Making in Learning Object Recommenders", *An International Journal of Research and Surveys*, Vol. 3, No. 3, September, 2009, pp757-763.
2. Pukkhem, N. and Vatanawood, W. (2005) "A Multi-Instructor Cooperative Model for Supporting Learning Objects Aggregation based on XML-Based Planning Strategy," *WSEAS Transactions on Computer*, Vol.4, Iss. 10, October 2005, pp1390-1398.

C.2 International Conferences

1. Pukkhem, N., Vatanawood, W. (2009)" An Evidential Reasoning Approach for Learning Object Recommendation with Uncertainty", *In Proceedings of Fourth International Conference on Innovative Computing, Information and Control*, December, Kaohsiung, Taiwan, 2009.
2. Pukkhem, N., Vatanawood, W. (2009)"Using Ontological Modeling in Multi-Expert Guiding based Learning Object Recommendation", In Proceedings of International Conference on Computer Engineering and Applications, Manila, Philippines, pp.73-77.
3. Pukkhem, N., Evens, M.W. and Vatanawood, W.(2006) "An Estimation Function for Selecting the Suitable Learning Objects in Adaptive Education Systems", *In Proceedings of the 4th International Conference on Education and Information systems, Technologies and Applications: EISTA'06),* July 20-23, Orlando, Florida, USA, 2006.

4. Pukkhem, N., Evens, M.W. and Vatanawood, W.(2006) "The Concept Path Combination Model for Supporting a Personalized Learning Path in Adaptive Educational Systems", *In Proceedings of the 2006 International Conference on e-Learning, e-Business, Enterprise Information Systems, e-Government, and Outsourcing (EEE'06),* June 26-29, Las Vegas, USA, 2006.
5. Pukkhem, N. and Vatanawood, W.(2006) "The Sequence Pattern Combination Model for Solving Various Content Aggregation Designs Problem", *In Proceedings of the International Conference on e-Learning (ICEL 2006)*, June 22-23, Montreal, Canada, 2006.
6. Pukkhem, N. and Vatanawood, W. (2005) "A Course Blueprint Designs using Learning Objects Approach and XML-Based Planning for Multi-Instructors Learning Systems," *In Proceedings of the 5th WSEAS International Conference on Distance Learning and Web Engineering*, Corfu Island, Greece, August 23-25, 2005
7. Pukkhem, N. and Vatanawood, W. (2005) "Instructional Design using Component-Based Development and Learning Object Classification," *In Proceedings* of *the 5th IEEE International Conference on Advanced Learning Technologies*, Taiwan, July 2005, pp492-494.

BIOGRAPHY

Name	Noppamas Pukkhem
Sex	Female
Marital Status	Married
Date of Birth	November 25, 1977
Place of Birth	Nakhonsithammarat
Permanent Address	10 Moo 1, Nearn-Khow Village, Soi 10, KuanLang, Hadyai, Songkla, 90110

Education:

2004 – Present	Ph.D. studies, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
2003	M.Sc. in Computer Science, Prince of Songkla University *Funding source*: - Thai Government Scholarship
2000	B.Sc. in Computer Science, Prince of Songkla University