STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING USING A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY

SUMALEE SONAMTHIANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
(SCIENCE AND TECHNOLOGY EDUCATION)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2012

Thesis
entitled

STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING USING A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY

……………………………….
Miss Sumalee Sonamthiang,
Candidate

……………………………….
Assoc. Prof. Kanlaya Naruedomkul,
Ph.D.
Major advisor

……………………………….
Prof. Nick Cercone,
Ph.D.
Co-advisor

………………………………
Assoc. Prof. Booncharoen Sirinaovakul,
Ph.D.
Co-advisor

……………………………..…
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……………………………….
Lect. Piyachat Jittam, Ph.D.
Program Director
Doctor of Philosophy Program in
Science and Technology Education
Institute for Innovative Learning
Mahidol University

Thesis
entitled

STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING USING A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Doctor of Philosophy (Science and Technology Education)
on
July 25, 2012

……………………………….
Miss Sumalee Sonamthiang,
Candidate

………………………………
Assoc. Prof. Wanchai Rivepiboon,
Ph.D.
Chair

……………………………….
Assoc. Prof. Kanlaya Naruedomkul,
Ph.D.
Member

………………………………
Prof. Nick Cercone,
Ph.D.
Member

……………………………….
Assoc. Prof. Booncharoen Sirinaovakul,
Ph.D.
Member

………………………………
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

………………………………
Assoc. Prof. Wannapong Triampo,
Ph.D.
Director
Institute for Innovative Learning
Mahidol University

ACKNOWLEDGEMENTS

The success of this thesis would not have been possible without the support, hard work, and efforts of a large number of individuals and institutions. I am particularly grateful for my major supervisor Assoc. Prof. Kanlanya Naruedomkul who gave me superior support and guidance. Her endless efforts and patience are invaluable to me. She has opened windows to this student and she will always be my role model in both work and life. It is an honor for me to thank Prof. Nick Cercone. He has made available his support in a number of ways. Prof. Cercone hosted and provided partial of financial support during my research at Dalhousie University and York University, Canada (2006-2007). His inspiring words gave me many initial ideas to this study. He is very special person who is not only best supervisor, but also a best uncle and best friend of mine. I would like to thank Assoc. Prof. Booncharoen Sirinaovakul, co-supervisor, for his guidance from the proposal until the reading proof of this dissertation.

I would like to express my sincere gratitude to Dr. Bhinyo Panijphan and Dr. Pintip Ruenwongsa for their supervision and guidance during my first years at Institute for Innovative Learning, Mahidol University. I am also grateful Dr. Sutha Lualamai for his kindness helps.

My sincere gratitude is also expressed to my colleagues at AI LAB and the first alumni in Science and Technology Education Program for their support, care, share, and give invaluable ideas and hands all along.

At last, my deepest gratitude goes to my beloved father, mother, brother, dearest husband, and my father and mother in law for their endless love and support.

Sumalee Sonamthiang

STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING USING A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY

SUMALEE SONAMTHIANG 4637886 ILSE/D

Ph.D. (SCIENCE AND TECHNOLOGY EDUCATION)

THESIS ADVISORY COMMITTEE: KANLAYA NARUEDOMKUL, Ph.D. (COMPUTER SCIENCE), NICK CERCONE, Ph.D. (COMPUTING SCIENCE), BOONCHAROEN SIRINAOVAKUL, Ph.D. (ENGINEERING)

ABSTRACT

The lack of proper prior knowledge affects students' ability to learn a new concept. Therefore, teachers need to know the student's prior knowledge before designing the instruction plan that best suits each student. Unfortunately, determining the prior knowledge of individual students is time consuming. In addition, designing and implementing the instruction plans to serve individual students with different prior knowledge is quite difficult. In this thesis, we propose the approach to group the students based on their similar prior knowledges and if their prior knowledge is inadequate, then the required prerequisite concept will be mapped to each student. We designed and developed a student clustering and prerequisite concept requirement mapping tool called RoughClust. RoughClust applied a rough set-based granular concept hierarchy (GCH) and domain dependency in clustering the students' pretest data. Then, the group's characteristics were used as criteria to map the students to the groups' prerequisite concept requirement. RoughClust can also provide the teachers with instructional materials for convenient use in teaching. RoughClust was evaluated for its clustering accuracy, coverage, and user satisfaction. The evaluation results showed that RoughClust provides acceptable accuracy and coverage, and satisfies the users' needs. Moreover, the teachers expressed the desire to use RoughClust in instruction planning.

KEY WORDS: GRANULAR CONCEPT HIERARCHY/ ROUH SETS/ STUDENT CLUSTERING/ PREREQUSITE CONCEPT REQUIREMENT

127 pages

การจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่จำเป็นโดยใช้ระเบียบวิธีการจัดกลุ่มข้อมูลอย่างเป็นลำดับชั้นด้วยราฟเซต

STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING USING A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY

สุมาลี โสน้ำเที่ยง 4637886 ILSE/D

ปร.ด. (วิทยาศาสตร์และเทคโนโลยีศึกษา)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์: กัลยา นฤดมกุล, Ph.D. (COMPUTER SCIENCE), นิค เซอโคน, Ph.D. (COMPUTING SCIENCE), บุญเจริญ ศิริเนาวกุล, Ph.D. (ENGINEERING)

บทคัดย่อ

การขาดความรู้ที่จำเป็นต่อการเรียนรู้ส่งผลกระทบโดยตรงต่อการเรียนรู้หน่วยความรู้ใหม่ของนักเรียน ดังนั้นจึงจำเป็นอย่างยิ่งที่ครูต้องทราบถึงความรู้พื้นฐานของนักเรียนในการวางแผนการจัดการเรียนการสอนในชั้นเรียน นอกจากนี้ความรู้พื้นฐานของนักเรียนที่แตกต่างกันยังผลให้การจัดการเรียนการสอนให้เหมาะกับผู้เรียนแต่ละคนเป็นไปได้ยาก การวัดความรู้พื้นฐานของนักเรียนเป็นรายบุคคลนั้นใช้เวลามาก งานวิจัยนี้จึงได้เสนอวิธีการการจัดกลุ่มนักเรียนตามความรู้พื้นฐาน ค้นหาความรู้พื้นฐานที่จำเป็นที่นักเรียนในกลุ่มต้องได้รับการซ่อมเสริม และเตรียมสื่อการเรียนรู้สำหรับกลุ่ม เพื่อเป็นแนวทางให้ครูใช้วางแผนการจัดการเรียนการสอนให้เหมาะสมกับความรู้พื้นฐานของนักเรียนทั้งแบบกลุ่มและแบบรายบุคคลได้อย่างสะดวกและลดเวลา ในการวิจัยนี้ ผู้วิจัยได้ออกแบบพัฒนาระบบการจัดกลุ่มนักเรียนออนไลน์ ชื่อ RoughClust โดยใช้ระเบียบวิธีการจัดกลุ่มข้อมูลอย่างเป็นลำดับชั้นด้วยราฟเซตและข้อมูลคำตอบของนักเรียนจากแบบทดสอบความรู้พื้นฐาน ในการจัดกลุ่มนักเรียนตามความรู้พื้นฐาน และใช้ความสัมพันธ์แบบขึ้นต่อกันของหน่วยการเรียนรู้ในการค้นหาหน่วยการเรียนรู้ที่จำเป็นต้องได้รับการซ่อมเสริม ผู้วิจัยได้วัดและประเมินผล RoughClust ในเรื่องความถูกต้องและครอบคลุมของการจัดกลุ่มนักเรียนและความพึงพอใจในการใช้งานระบบ ผลการประเมินพบว่า RoughClust สามารถใช้จัดกลุ่มนักเรียนได้อย่างถูกต้องและครอบคลุมตามความต้องการของครู และครูมีความพึงพอใจ อีกทั้งประสงค์จะนำ RoughClust ไปใช้ประกอบการวางแผนการสอนในชั้นเรียนอีกด้วย

127 หน้า

CONTENTS

CONTENTS (cont.)

CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (cont.)

Figure **Page**

LIST OF ABBREVIATIONS

Abbreviation	
AI	Artificial Intelligence
AHS	Adaptive Hypermedia System
DST	Dempster-Shafer Theory
GCH	Granular Concept Hierarchy
GrC	Granular Computing
ITS	Intelligent Tutoring System
SRI	Standard Rough Inclusion
RST	Rough Set Theory

CHAPTER I
INTRODUCTION

"Students must learn mathematics with understanding, actively building new knowledge from experience and prior knowledge."

From the Principles for School Mathematics of NCTM standards, copyright NCTM 2000-2004.

1.1 Motivation

Students bring preconceptions into learning situations. If their prior understanding is not engaged, they may fail to grasp new concepts. In 1968, Ausubel stated: "the most important single factor influencing learning is what the learner already knows. Ascertain this and teach them accordingly" (Ausubel, 1968). A number of researches also showed the essence of students' existing knowledge prior to instruction in science learning (Hewson & Hewson, 1983; Harrison et al., 1999). Clark (2003) stated that "prior experience as the most significant learner characteristic influencing learning." Furthermore, Mayer (2004) claimed that expert teachers who hold a more complex conception of prior knowledge than novice teachers make use of their students' prior knowledge in significant ways during instruction.

It has been proved that applying student prior knowledge can improve student learning achievement in learning English Language as surveyed in *the report* of *NCAC Background Knowledge Instruction and the Implications for UDL Implementation*.

> "…Prior knowledge has a large influence on student performance, explaining up to 81% of the variance in posttest scores (Dochy, Segers, & Buehl, 1999). And there is a well established correlation between prior knowledge and reading comprehension (Langer, 1984; Long, Winograd, & Bridget, 1989; Stevens, 1980). Irrespective of students' reading ability, high prior knowledge of a subject area or key vocabulary for a text often

means higher scores on reading comprehension measures (Langer, 1984; Long et al., 1989; Stevens, 1980). In addition, high correlations have been found between prior knowledge and speed and accuracy of study behavior (reviewed in (Dochy et al., 1999) as well as student interest in a topic (Tobias, 1994). Thus, prior knowledge is associated with beneficial academic behaviors and higher academic performance….”

Research has suggested a number of reasons in support of taking into account the student prior knowledge in instruction. Christen and Murphy (1991) proposed the three levels of prior knowledge as *much*, *some*, or *little* prior knowledge. In this research, students can be grouped into these three categories. Based on the student prior knowledge levels, the teacher can make a specific instruction to build, correct, or enhance the prior knowledge of the learners. Kalyuga, Ayres, Chandler and Sweller (2003) stated that inappropriate student prior knowledge level and new information in instruction materials affected the student learning. It results in students failing to connect and apply mathematical concepts to solve real world problems in their daily life. Therefore, understanding of the student prior knowledge is an influential factor to student learning achievement. Teachers need as much knowledge about student prior knowledge of learning concepts as physicians have about the causes of illnesses. As a result, it is worthy to investigate the student prior knowledge and engage in instruction planning.

Mathematics is one of the learning subjects that student prior knowledge is essential since math composed of incremental and sequential concepts such as cumulating of knowledge from arithmetic to algebra concepts suggested in (Pillay, 1998). Math can become easier to learn when math foundation already set in place (Ashlock, 2010). Only when students have fuller understanding of the foundation, they can effectively move forward to more complicated concepts. Without sequences of knowledge and skills, a student cannot understand and learning the concepts well (Clarke et. al., 2005). Ashlock (2010) also suggested that, in learning mathematics, the best instruction should target each student's specific misunderstanding prior to learning, whether it is computational or conceptual. Moreover, research on presentation sequencing found that learning was enhanced if supportive information was presented sequentially before or during practice (Kester et al., 2004; Kester et. al., 2001). As a result, learning math concepts should be in sequential order to establish student prior knowledge foundation.

An example of prerequisite concepts that the student should hold prior learning fraction is shown as the triangle in Figure 1.1. From the figure, the prerequisite concepts for learning Fractions are listed based on the Basic Education Core Curriculum B.E. 2551 (A.D. 2008) Mathematics. The triangle shows the fundamental knowledge in the lower levels of the triangle which affect learning more specific concepts in the upper levels of the triangle. For example, to compute a result of adding two fractions with different denominators, there are many concepts and skills essential such as Least Common multiple (L.C.M.), Greatest Common Divisor (G.C.D.), multiplication, division, addition, subtraction, and whole number. An example from classrooms illustrates the need of prior knowledge in instruction. If a student fails to compute adding two fractions with different denominators as shown in Figure 1.2, it is unclear about which granular prerequisite concepts or skills are premature and required to be treated. To find which prerequisite required, the teacher must retest the students in order to backtrack to find the underlying granular prerequisite knowledge that the student needs to be treated specifically.

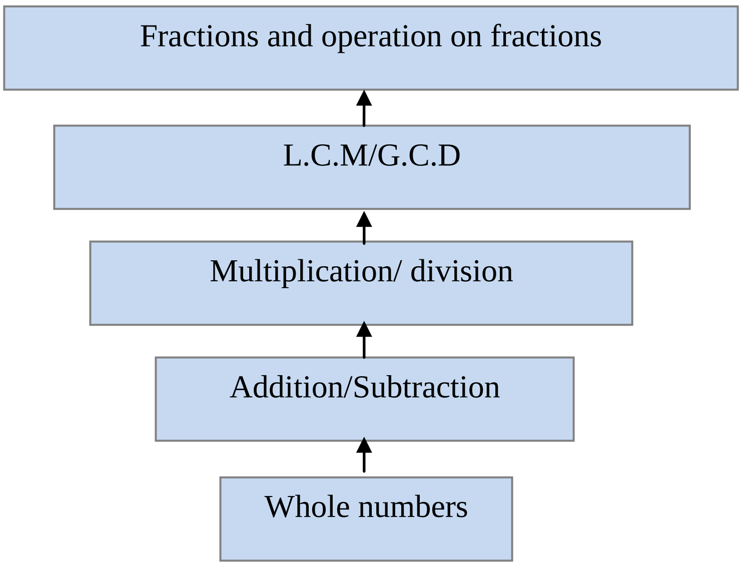


Figure 1.1 Prerequisite concepts triangle for learning fraction concept

A students' common answer of adding two fractions:

$$\frac{2}{5}+\frac{2}{7} = \frac{4}{7}$$

Figure 1.2 A common students' answer of adding two fractions

To encourage students to learn mathematical concepts, a student needs proper instruction accordingly to their prerequisite knowledge. If a student tried to solve a problem that he/she has a lack of prerequisite knowledge, he could fail and continue to accumulate the wrong concepts. Failing to solve mathematics problems often can develop negative attitude to not only the subject but also to the teachers. If the students' granular prerequisite knowledge are not revealed and treated properly, their learning problem will be added up constantly.

In order to develop a proper instruction plan, the teachers must understand the student prior knowledge. Christen et al. (1990) suggested that teachers may choose to do some of the following activities to acquire what prior knowledge exists:

1) Brainstorm a topic. Write all the information solicited from the students on the chalkboard, a piece of paper, or transparency.

2) Ask specific and/or general questions about the topic. See what responses are given.

3) Post a problem or a scenario then find out what the students know about the idea presented.

These three activities usually provide descriptive information of the students reflect about what they already known. Unfortunately, unknown knowledge and misconception of the specific prerequisite concepts may not be uncovered. Moreover, if possible to perform such the mentioned activities, it is uneasy and time consuming task to understanding and treating individual student needs in a classroom setting. One way to reveal the prerequisite concept requirement is to give student a test prior to undertaking a course. The pretest result can be used to find the real causes of the problems about which underlying granular prerequisite knowledge that the student needs to be treated specifically. However, it will add more workload for the teachers in analyzing the pretest to understand each student's prior knowledge.

As a result, there are efforts to automatically acquire student prior knowledge such as a student modeling in Intelligent Tutoring Systems (ITSs) (Sleeman & Brown, 1982; Anderson et al., 1985) and diagnostic testing systems (Panjaburee et al., 2010). Student model is seen as the heart of an ITS since it represents student cognition (and affective traits). The student's prior knowledge and learning characteristics is initiated as the primary student model. However, acquiring student prior knowledge through student modeling is limited for teachers to extend the student prior knowledge for the more granular concepts. Therefore, an automatic system that provides the teachers a simpler way to acquire student prior knowledge granularly is required. In this thesis, therefore, we find an approach that is able to:

1) Group the students who hold common knowledge error together with their characteristics.

2) Map to student's granular prerequisite knowledge in order to help teachers find the right granular prerequisite knowledge and treat them properly either individual or group treatment.

3) Suggest appropriate and handy instruction materials such as handouts, exercises, and a set of test items related to each granular learning concept.

4) Articulate the granular prerequisite knowledge level as deep as the teachers desire.

We believe that a mapping approach from a student cluster to prerequisite concepts structure can reveal the group's learning difficulty and prerequisite concept requirement. The group's prerequisite concept requirement can be useful for instruction administrators and teachers in order to plan appropriate course outline, prepare handouts, or arrange special classes for each group of students. Moreover, teachers can apply the "within-class ability grouping" (Kulik & Kulik,1989) and "re-grouping" (Slavin, 1987) for appropriate classroom instructions. The "within-class ability grouping" allows each group to work on different materials specific to their requirement and abilities. In the "regrouping" approach, students are pre-assessed and grouped according to their ability in the subjects (usually in mathematics and language subject). Then the students attend specific classroom for the subjects with their like-ability group and return to their regular classroom for other subjects. Therefore, curriculum and instruction plan based specific requirement of the group can be developed.

Even though the “within-class ability grouping” and “regrouping” approach have proved their effective instruction to serve group-based requirement (Kulik & Kulik, 1982; Kulik & Kulik,1989; Slavin, 1987), identifying the group or individual student prior knowledge is a time consuming task and difficult to implement. Therefore, an automatic approach to group students and find the group-based prior knowledge requirement is worthy.

In order to automatically grouping (or clustering) students, a test result will be collected to generate an input data set to the clustering system. However, students’ test data can be uncertain for many reasons. For example, it is very vague to model a person knowledge as the only way to perceive about the students’ knowledge is limited by answers of a test and a questionnaire. Students use various strategies to perform the test which cause uncertainties such as guessing, unintentional answering, or skipping (Sonamthiang, Naruedomkul, & Cercone, 2006). To cope with uncertain knowledge about the students, a soft computing approach called rough sets (Pawlak, 1991; Pawlak, 1998) is investigated. Using the rough set approach is capable of clustering under situation involving uncertainty. Results of rough set clustering are approximations which may not reach total accuracy but is acceptable or sufficient to solve problems. Sufficient details and accuracy of knowledge (or clusters) is very crucial and inspired the state of the art in Granular Computing (GrC) field (Pedrycz, Skowron & Kreinovich, 2008; Yao, 2009).

In education, the hierarchy of granular concepts naturally underlies in topics within a course, prerequisite concepts, the Concept-Effect Relationship model (Hwang, 2003), book chapters, structured problem solving such as structured writing (Yao, 2007); for example. As a result, in this study we applied our approach of granular concept hierarchy mapping based on rough sets (Sonamthiang, Naruedomkul, & Cercone, 2007) with the domain conceptual hierarchy to student clustering and articulating granular prerequisite concept requirement.

1.2 Thesis statement

Understanding students' granular prerequisite knowledge towards a teaching concept is uneasy and time consuming task. Moreover, students in the same classroom are different in ability levels, interests, and background knowledge; hence, instruction such a classroom is difficult for the teachers to accommodate all students' requirement. This thesis presents an alternative approach to student clustering which takes into account the student pretest result. Groups of similar students are discovered together with information of student's learning difficulty for group or individual students as desired. The discovered student clusters and the clusters' information can be benefited for managerial classroom and instruction planning. In addition to the student clustering, an approach for the prerequisite concept requirement mapping is contributed by using the students' pretest and the domain prerequisite. We designed and developed an automatic system called RoughClust as a tool for teacher to acquire the student groups based on their prior knowledge and plan an instruction that suits each group requirement.

The clustering approach presented in this thesis is a formal method of granular concept hierarchy approximation based on Rough Set Theory (RST) which integrates soft computing to education applications. The approach is domain independent; therefore, it is not limited to student data sets only. The approach takes into account the multilevel external relationships amongst objects (e.g. students) in an information system format; or, a tabular format. The multilevel external relation is extracted based on a dominance degree of attribute subset when the domain dependency hierarchy is not available. For clustering students based on the test result, the domain dependency hierarchy is used to control partitioning at each level. A rough set is used to define a concept which permits crisp and vague concept definition possible. As a result, using rough sets to partition and define concept provide the concepts with more semantic and more accessible comparing with traditional clustering methods that are based on only similarity measurement clustering. Moreover, experiments were conducted to evaluate the performance of the innovative clustering approach.

1.3 Objectives

Objectives of this study are two-fold. The primary objective of this thesis is to find an approach to cluster students into groups that students in the same group hold similar learning characteristics. The clustering approach should concern domain dependency of the learning subjects together with students' background knowledge and learning performance. A system for student clustering is to be developed in order to automatically discover student clusters and provide student clusters' description as information for instructional administrators and teachers to corporately manage classrooms and instruction to meet the group/individual students' requirement.

The secondary objective is to use the discovered student cluster's description for mapping to the prerequisite concept requirement for each group. The prerequisite concept requirement supports teacher's understanding about his students' prior knowledge so that the teacher can plan proper instruction to enhance group-based learning achievement. In addition, RoughClust also provided the teacher with appropriate learning materials for the groups for convenient use such as handouts, exercises, lesson plans, and pretests.

1.4 Contributions

Contributions to Computer Science:

Our main contribution is an alternative approach for granular concept hierarchical clustering based on RST. The rough set-based hierarchical clustering provide clusters with more semantic and more accessibility for cluster mapping comparing to traditional clustering methods. Our hierarchical concept clustering approach can be applied to an information system both with or without the domain dependency guidance. If the domain dependency available, the dependency structure can be used to control multi-level partitioning. If the domain dependency is unknown, *most dominant attribute subset*, *reducts*, and *core attribute* are applied for selecting attribute subset for each level partitioning.

An automatic granular concept hierarchy (GCH) clustering system for domain independent hierarchical concept clustering based on rough sets was developed as a prototype. Input to the system is a data set in information system format and the output is a granular concept hierarchy. Users can determine appropriate level of granular clusters to solve a particular problem using heuristics criteria.

Contributions to Education Field:

Our secondary contribution is an application of clustering student data to discover groups of students who share similar difficulties in learning a subject. RoughClust was developed as a tool for student clustering and prerequisite concept requirement mapping. The discovered clusters and clusters' information provides insight for administrators and teachers to manage instruction and resources for each group suitably. Moreover, teachers are provided with handy instruction material that is suitable for each group/individual student instruction. Using the system, teachers are also allowed to define more granular prerequisite knowledge as desire.

1.5 Benefits

Using RoughClust system, the instruction administrators and teachers are provided with student groups with their similar learning characteristics in domain. Thus, classroom management such as "within-class grouping" or "regrouping" based on the groups can be implemented.

Students, teachers, and parents will benefits from RoughClust as follows.

1. More comfortable classroom learning environment is promoted. Students complacently learn with their learning pace, performance, materials, and like-ability peers.

2. Teacher' work load is reduced in understanding the students' prior knowledge and preparing instruction since teachers can use group-based prerequisite concept requirement to construct a group-based curriculum and plan an instruction. In addition, teachers can use the provided materials such as hand outs, exercises, and tests to use in the instruction planning.

3. Cost reduction for schools as there is not necessary to arrange special classes for special needed and gifted students. Diversities of student learning ability can be accommodated with group-based curriculum within classroom learning.

4. Once students are treated as they need at the school, supplementary tutoring in out-of-school is reduced. Parents' expense for their children tutoring can be

lower or cut off. More importantly, students are able to spend more time for other activities rather than attending the tutoring classes after school or at weekends.

1.6 Thesis outline

Chapter 2 presents literature reviews of existing study in automatic system to diagnose student prior knowledge and explores the studies and approaches to define such granular prerequisite concepts of student's prior knowledge. Then, background in RST and framework of our approach to granular concept hierarchical clustering is presented in Chapter 3. We also illustrate the granular concept hierarchy construction algorithms in this chapter. Chapter 4 presents the design and implementation of RoughClust followed by its user interface and output interpretation. Chapter 5 illustrates an evaluation and discusses the results of clustering an artificial data set and student data sets. We also present an evaluation of implementing the group-based instruction in real classroom setting and an evaluation of user satisfaction in this chapter. The thesis is concluded in Chapter 6 which is followed by discussion and further research.

CHAPTER II
LITERATURE REVIEWS

New knowledge is acquired by extending and revising prior knowledge; new ideas acquire meaning when they are presented in a coherent relationship to one another.

Richard F. Elmore, Education for Judgment: The Artistry of Discussion Leadership, xii.

Students have different experiences and different prior knowledge. Teachers and educators must serve each student with different instruction plan so that they can correct their misconception or learn new concepts. However, understanding individual student prior knowledge is a time consuming task and it would not be possible for a teacher to implement an instruction to simultaneously suite all student prior knowledge in a classroom. As a result, grouping students that hold similar prior knowledge is essential for the teachers and educators to provide more chance to implement the instruction that accommodates the group's knowledge requirement. Finding an approach to group the students based on their similar prior knowledge in learning a concept domain is worthy to investigate. Moreover, a mapping approach to find the student prior knowledge defined as "granular prerequisite concept requirement" for both groups and individual students is studied.

This chapter gives literature reviews and related studies to both research in finding student's prerequisite concept requirement and hierarchical clustering approaches. We reviews related field of our study which inspires our approach which is Granular Computing (GrC) in Section 2.1, then review on systems for finding student granular prerequisite concept requirement is presented in Section 2.2. Previous works in hierarchical clustering and conceptual clustering are discussed in Section 2.3. Finally, we summarize the role of our approach in student learning in Section 2.4.

2.1 Granular prerequisite concept requirement

Knowledge is organized in hierarchical network of concepts and attributes (Ericsson & Polson, 1988). Concepts are related to one another and the hierarchical network is formed by the relational propositions from granular simpler concepts to more complex concepts. Examples of the hierarchical knowledge structure are such as the definable structure of granular knowledge and solution to solve mathematical problems defined in (Pillay et al., 1998) and problem solving skill development in programming (Weber & Specht, 1997). Moreover, in human problem solving, a human categorization skill is essential to mental life and results in organizations of hierarchy knowledge structure (Yao, 2010.)

The hierarchical structure of the domain knowledge has shown a framework to modeling the student prior knowledge (Hwang, 2003; Panjaburee, 2010). By using the hierarchical relationships of a learning concept domain, student prior knowledge requirement such as *poorly-learn paths* can be revealed (Hwang, 2003; Panjaburee, 2010). However, only the hierarchical structure cannot answer the question of "how should we articulate the hierarchy to identify the real cause or the most granular concept that the student should master in order to well learn the learning concept?" A human-inspired granular computing research suggested multilevel problem solving approach by focusing on the most appropriate level of granular knowledge (Yao, 2010). To find the most appropriate level of granularity in multilevel problem solving, switching between the levels of granularity is necessary in order to disregarding the irrelevant knowledge and refining for the sufficient granular knowledge as necessary.

GrC provides us perspectives of computing of the hierarchical network structure for granular information in helping student learning. GrC is an emerging field of study in information processing. In 1997, Professor T.Y. Lin introduced the term "granular computing" (Lin, 1997) which defines the study the processing of complex information entities called information granules in order to derive granular data abstraction and knowledge from the information. To process such complex information in respect of concept granularity, two main set-oriented theories, fuzzy sets and rough sets, were proposed. In 1979, Professor Lotfi A. Zadeh introduced information granulation and suggested fuzzy set theory for computing with uncertain data. In 1982, Professor Zdzislaw Pawlak proposed the Rough Set Theory (RST) (Pawlak, 1982) which provides a framework to hierarchical information granulation. Yao (2000) collected and presented the following quotations to help us better understanding GrC.

> "Granulation of an object A leads to a collections of granules of A, with a granule being a clump of points (objects) drawn together by indistinguishability, similarity, proximity or functionality." (Zadeh, 1997a)
>
> "The theory of fuzzy information granulation (TFIG) is inspired by the ways in which humans granulate information and reason with it." (Zadeh, 1997a)
>
> "TFIG builds on the existing machinery of fuzzy information granulation in fuzzy logic but takes it to a significantly higher level of generality, consolidates its foundations and suggests new directions." (Zadeh, 1997a)
>
> "GrC is a superset of the theory of fuzzy information granulation, rough set theory and interval computations, and is a subset of granular mathematics." (Zadeh, 1997b)

Sonamthiang, Cercone and Naruedomkul (2007a) presented a GrC approach based on rough sets to model a granular knowledge hierarchy. This approach has been applied to hierarchical data clustering. The hierarchical clusters provide various levels of granularity that are accessible by the clusters' intensions in the logical rules.

In applying GrC to examine the student's prior knowledge, we defined a specific term which is *granular prerequisite concept requirement* as the following:

> "Granular prerequisite concept requirement of a specific concept is composed of a set of the sufficient and relevant concepts that must be well learnt by the student to learn a new concept or correct a misconception."

In this study, we apply the hierarchical data clustering approach (Sonamthiang, Cercone & Naruedomkul, 2007a) to cluster students' test data in order to group students who share common knowledge error. Moreover, a granular approach is applied to map to the prerequisite concept requirement for the group or individual student. This approach will be used in developing a tool for teachers to categorize their students into groups and find the group prerequisite concept requirement.

2.2 Automatic systems for finding student granular prerequisite concept requirement

2.2.1 Student prior knowledge in intelligent tutoring systems

Advanced computer and information communication technology nowadays provide more channels to promote education. Intelligent tutoring system or ITS is an advanced computer-based learning systems to conduct one-to-one tutoring. Katie Hafner defined an ITS in The New York Times (September 16, 2004) that "Broadly defined, an intelligent tutoring system is educational software containing an artificial intelligence component. The software tracks students' work, tailoring feedback and hints along the way. By collecting information on a particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work." ITSs are generally developed as multi-component system (Sonamthiang, Cercone & Naruedomkul, 2007a). Artificial intelligence is generally embedded in ITS in four main components: domain knowledge model, student model, pedagogical model, and user interface model.

The first ITS, SCHOLAR, was introduced in 1970 (Carbonell, 1970). Since then, the use of ITSs has increased in the past twenties years. Various subjects that have adopted ITSs are exemplified in Table 2.1. These systems embedded AI techniques to improve their performance on tutoring context. For instance, a Bayesian network approach was used in tutoring Newtonian Physics of Andes tutor (Conati et al., 1997) by modeling the student's plan in problem solving and the network model is used to predict the students' actions. PAT tutor (Koedinger et al., 1997) has been used by a great number of schools and colleges to teach Algebra. PAT tutors by modeling current cognitive status of student by using the predefined if-then production rules of both correct and incorrect steps generally taken by students. Moreover, PAT tutor applies a Bayesian estimation to compute the students' strengths and weaknesses according to the student's cognitive model. These widely use of ITS presents the success of adopting AI to help instructions in computer tutors.

Table 2.1 Examples of ITSs in various domains.

Subject	ITSs
Physics	Andes (Conati et al., 1997)
Mathematics	PAT Algebra (Koedinger et al., 1997), Cognitive Tutor(Aleven& Koedinger, 2002) ActiveMath (Melis & Siekmann, 2004) WEST (Burton & Brown, 1982)
Language	ICALL (Virvou & Tsiriga, 2001), CAPIT (Mayo et al., 2000)
Formal Language	GET-BIT (Devedzic, 2003)
Computer programming	LISP tutor (Corbett et al., 1990), ASSERT (Baffes & Mooney, 1996), PROUST (Johnson & Soloway, 1985), SQL Tutor (Mitrovic, 2003), ELM-ART II (Weber and Specht, 1997)
Medicine	SlideTutor (Crowley et al., 2003)
Industrial training	ABIT (Capuano et al., 2000)
Military training	SimBionic (http://www.simbionic.com/)
Law	LITES (Span, 1993)

The domain knowledge model of many ITSs is built on prior knowledge, so some of the knowledge components used in the ITSs may be mastered by students before they take the course. However, the ITSs cannot just assume that all students will have mastered all the prerequisites in the student model. Some student must learn the prerequisites as they learn every knowledge component in the course. Thus, the ITSs should estimate mastery of the prerequisite knowledge components along with the others in the student model. Many ITSs investigated student prior knowledge by giving a student a pretest and questionnaire, then use the test result to initialize the student model.

The student modeling approaches are categorized as follows.

- **Stereotype model** uses constraints to classify students that have common characteristics into groups or classes. Stereotype approach was introduced in User Modeling field by (Rich, 1989). Machine learning techniques such as K-nearest neighbor algorithms are used to group students (Virvou & Tsiriga, 2003). However, the class constraints are predefined in which the constraints may be inappropriate with real student prior knowledge during learning situation.
- **Overlay model** collects and represents corrections of students' knowledge as a subset of the ideal knowledge model of the expert. An overlay student model is easy to implement but we cannot find misconceptions that might occur in student knowledge via this model. The systems that applied overlay model are (Brusilovsky ,1996) and (Kavčič, 2004), for example.
- **Differential model** considers two categories of student's knowledge which are knowledge that already known and knowledge that the student cannot expect to know and then compares of the two knowledge categories. Similar to the overlay model, both types of knowledge are represented as subset of the expert model. Therefore, the coaching mechanism determines difference between the two types of student knowledge model about when it is appropriate to interrupt the student with advice. An example of ITS that uses the student differential model is WEST tutor (Brown & Burton, 1982).
- **Perturbation model or bug model** is the student model that includes both correct knowledge and incorrect knowledge of the student. The incorrect knowledge is separately store in the system knowledgebase called bug library (Greer & McCalla, 1994). ITSs that deployed this type of student modeling are, e.g., CIRCSIM tutor (Kim et al., 1989), ASSERT tutor (Baffes & Mooney, 1996).
- **Constraint-based model** represents a student model using two sets of constraints in problem solving solutions of the student: constraints that the student correctly and incorrectly performs. Therefore, the expert solutions are predefined as the knowledgebase of constraints in order to compare the

acceptable and unacceptable solution steps taken by the student in problem solving. An example of tutors that implemented using this technique in student modeling is SQL tutor (Mitrovic, 2003).

- **Model tracing** is a well known method. It has been use in PAT tutor, SlideTutor, LISP tutor, Cognitive tutor, and Andes tutor. For example, the model tracing in LISP tutor is to model the student actions in LISP programming process rather than recognition of the entire student's algorithm. Two sets of predefined rules must be collected for the expert and the novice programming behaviour. Then the student action is mapped to these sets of predefined rules to obtain the student model. As a result, the student model is obtained through tracing the student programming actions and assistance can be provided when the student performs action taken by novice.
- **Episodic models** are especially used in adaptive hypermedia systems (AHSs). An example of the system is ELM-ART (Weber & Brusilovsky, 2001) which teaches LISP programming. In ELM-ART, an episode of a student programming contains description of concepts and rules required to programming coded by a particular student. Therefore, the descriptions are stored as episodes of programming.

Although there are various approaches to model student knowledge, the main concerns are to model current status of cognitive knowledge or to trace the student solution based on predefined problem solving paths. Prior knowledge of students is pre-assumed by representing the tutoring domain cocepts, tutoring methods, common or possible mistakes, and common misconceptions in the tutor knowledge. However, these ITSs did not quite concern student's granular prior knowledge of the tutoring domain and his/her learning behaviour in order to provide proper tutoring sessions.

To accommodate the granular prior knowledge and behaviour of the student during learning with an ITS, the tutor knowledge must be decomposable as well as a the tutoring strategies. Namely, hierarchical structure of the tutor knowledge should be able to be refined for teaching basic skills before higher-level concepts as the student prior knowledge requirement. For decomposable tutoring strategies, the

ITS must provide the details of basic and simple teaching strategies and medias for the lower-level concepts based on the student prior knowledge. In order to present granular teaching strategies, granular prerequisite requirement of a student must be first known by the tutor.

In addition to study of students' prior knowledge in ITSs, there has been intensive research on automatic system to diagnose student prior knowledge as reviewed in the next section.

2.2.2 Diagnostic systems for finding student prior knowledge

While there are various methods to assess students' prior knowledge, using a pretest is commonly used for direct prior knowledge assessment. However, conventional testing approaches usually assign only a score to each student, and this means that students are unable to realize their own strength and weaknesses. Moreover, teachers usually summarize and report the test result using basic statistical data of students' score (e.g., *Max, Min*, X, S^2). Only the statistical report does not provide sufficient information for teachers to adjust curriculum and instruction plan to serve students who need specific instruction based on their strength or weakness of prior knowledge. Teachers may manually and intensively diagnose the test results of individual student to uncover his/her prior knowledge requirement. However, diagnosing all individual pretest results is a time-consuming task and increases the teacher's workload.

One of existing automatic diagnosis systems for finding student prior knowledge is presented in our previous research (Sonamthiang, Cercone & Naruedomkul, 2006). The research presented an approach for diagnosis of student prior knowledge which can be applied to initialize the student model for an ITS development. The approach uses Demspter-Shafer Theory to prior knowledge cognitive estimation for all prior concepts. Once student registers to the ITS, he or she will be asked to take the pretest and answer a set of questionnaire. Student knowledge of all background concepts will be estimated (not merely the summary score of the pretest). The pretest has the following specification (Sonamthiang, Cercone & Naruedomkul, 2006).

1) Each item is independent from answering other items.

2) All item-concept relationships must be defined.

3) Weight of each item-concept relationships must be defined.

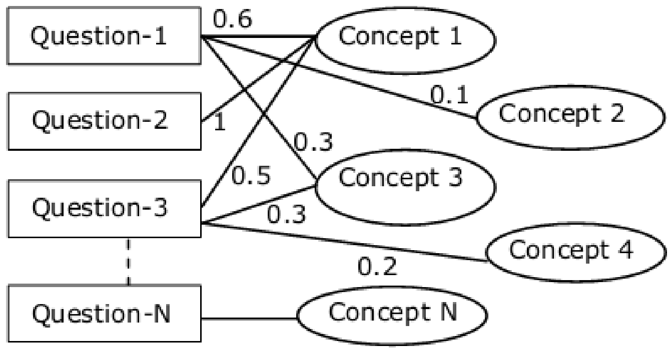


Figure 2.1 Relationships between test items and concepts with their weights

One or more related concepts are applied to answer a question. Therefore, each pretest item is responsible for perceiving student knowledge of a concept. However, related concepts may not have equal effect on answering an item. It is necessary to define weight of item-concept relationships as exampled in Figure 2.1. Students must have knowledge of *concept1*, *concept2*, and *concept3* to answer *Question1* correctly (assume no lucky guessing). Each relation of item-concept has normalized weight sum to 1, but not necessary to the converse relation. For instance, *concept1* has a weight of 0.6 (or 60 percent) of effecting on answering *question1*. The weight of the related concepts that affect answering each question can be defined by a voting result from the experts of the domain.

After define all the requirements, a student pretest result and the item-concept relationships with their weights are input and processed using Dempster-Shafer Theory. This study mainly concerns uncertainty of student taking the pretest. The reliability of student prior knowledge diagnosed by this method is not well studied. Moreover, the approach lacks of suggestions such as suitable learning path or how the ITS or teachers should proceed based on the diagnostic result.

According to Hwang et al. (2003) research on the Concept Effect Relationship (CER) model, the prerequisite relationships amongst concepts of a course are represented via a directed graph model (see Figure 2.2). For example, to learn the concept *Proper fraction,* a student must first learn basic concept of *Fraction*. To learn *Mixed number*, the student is required to first learning *Proper fraction* and *Improper fraction*. This model was used in diagnosing student learning problems. For example, if a student fails to answer most of the test items concerning *Mixed number*, the learning problem may be because of the student has not mastered *Mixed number* or its prerequisite concepts (such as *Proper fraction* or *Improper fraction*).

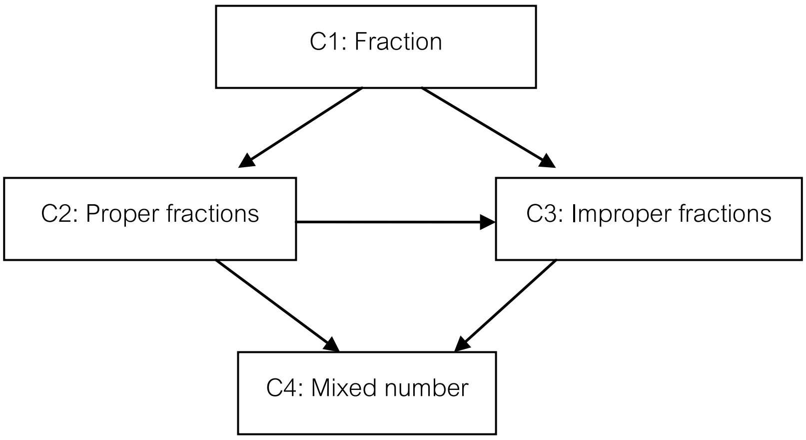


Figure 2.2 An example of Concept Effect Relationships in Fraction

Besides the CER model, the student's answers of the test are required to diagnose student knowledge error. The student's answers are used to define the Error Rate (ER) on a test and a threshold θ for the acceptable ER value must be defined. By tracing all of the possible learning paths and determining the student's ER, the *poorly-learned paths* of a student can be obtained. Therefore, the system can provide suggestion for each student about what prerequisite concepts are required in order to learn a particular concept. The CER model is capable of defining the domain knowledge of a testing diagnosis system to assist teachers in identifying weakness of student's learning as poorly-learned path. Therefore, in this thesis the CER model is used in defining prerequisite relationships amongst the learning concepts to map to the student's prerequisite concept requirement.

The CER model extended in (Panjaburee et al., 2010) is an automatic system for testing and diagnostic learning problems (TDLP). The diagnostic system deploys an approach to eliciting and integrating the weightings of test item-concept relationships from multi experts on top of the CER model. Input to the system is relationships amongst concepts, weights, and certainty degree given by multi experts to define the test. Once a student takes the test, the student answers are used to calculate the ER and diagnose the poorly-learned paths. Output of the system does not only show a student performance level but also suggestion of prerequisite concepts that should be revisited. However, the system does not allow teachers to define their own granular level of prerequisite concept hierarchy to the test structure. What if the students fail to understand all the concepts defined in the CER model? Where should the students go back start?

In addition to (Panjaburee et al., 2010), there is recent study which is a testing-based diagnosis system to help teachers in identifying and improving a student's prior knowledge before implementing an instruction (Lin et al., 2011). The authors present a prior knowledge diagnosis or PKD model. The model requires two input data which are the testing information and the student's answers information. The test information is represented by the relationships of concept-test item and the relationship amongst the concepts. This information is defined by teachers. The student's answers information is derived from relationships between their answers and the test item. PKD uses linear equations to compute the student strength and weakness based on those defined relationships. The authors claimed that the diagnostic system based on testing approach is effective to diagnose student prior knowledge comparing with the expert approach. Like TDLP, the PKD approach is limited to the predefined concept-test item relation and not yet concern granular prerequisite concept as specific as need and the two studies are capable of diagnosing only one-by-one student. As a result, the information of each student strength and weakness will be scattered and become difficult for teachers to summarize overall strength and weakness of their students and plan for classroom instruction.

While students have their individual differences in background knowledge and learning performance, students usually perform common errors. Ashlock (2010) addresses students' common errors and misconceptions in mathematics at primary and secondary level. The author suggested that most of these misconceptions are built up and accumulated from the student's prior knowledge. An example of a common misconception is multiplication of a whole number by a fraction less than one. Many students automatically assume that the multiplication result will be larger than the whole number since they have prior knowledge of multiplying of two whole numbers always results in a larger number. If the common errors are identified, an instruction to build connection of prior knowledge and new knowledge can be designed. Hence, the teachers will able motivate the students to learn new concepts and correct the students' misconceptions easier. However, it is challenged for the teachers to address the students' common errors in a large and diverse classroom.

We, therefore, present an approach to use student test on prior knowledge requirement as input data and then automatically cluster students into granular groups of student who have common errors. If students grouped by their common errors are revealed, teacher can make use of this information for instructional planning for group as well as for individual as necessary. The next section provides a survey on hierarchical data clustering approaches which will be used to clusters the students.

2.3 A survey on hierarchical data clustering

2.3.1 Classical data clustering

Categorization skill is considered as an intelligent behavior of a human. In categorization, one differentiates objects being determined as well as related objects together based on the objects description. This human learning skill was applied to machine learning called categorizing or clustering. Data clustering is a method to group objects that are similar in particular characteristics to discover meaningful relations among objects or abstraction in a dataset, with or without the predefined class. Popular applications of data clustering are amongst pattern recognition, information technology, image processing, biology, psychology, and marketing.

Data objects to be clustered is most frequently represented as an attribute-value system such as tables in a relational database, information systems, formal contexts, and decision tables. These representations structure the space of data in a tabular format which each column represents attribute of objects and each row represents individual object in the data set together with the object's values of the corresponding attributes. The tabular format represents data in flat and unconnected structure. The meaningful relations amongst objects are underlying in the pile of data since an object may be related to other objects in the problem space. Discovering of the meaningful relations is intensive studied in data classification and data clustering.

Existing clustering methods can be categorized based on the types of input data (e.g., spatial data, text, and numerical data), the hardware and software, the size of the dataset, and output cluster structure. By the output cluster structure, we can divide two types of clustering methods as follows.

- The non-hierarchical methods. In a dataset, an object is assigned to belong to a cluster based on the degree similarity. This type of clustering partitions a dataset of N objects into M clusters where $M \leq N$ by using a similarity measurement or distance measurement. Therefore, the similarity threshold must be defined to in determining which cluster is most similar to the determined object.
- The hierarchical methods. This type of clustering produces a set of nested clusters. Nested clusters can be induced by top down and bottom up manners. In the top down manners, the cluster hierarchy is constructed by dividing or partitioning a dataset into nested clusters by considering both global and local relationships of objects and clusters. The nested clustering will process the data until all individual objects belong to a cluster. In bottom up data clustering, pairs of objects is grouped to into a cluster considering local similarity of the object pair.

Classical data clustering provides information about cluster members with some similarity measurement of intra-cluster and/or inter cluster are computed, e.g., the K-means algorithm (Lloyd, 1982), and the Expectation–Maximization or EM algorithm (Dempster, Laird, & Rubin, 1977). However, a semantic conceptual description of clusters is not defined. In clustering student data set, only similarity measurements is not adequate since teachers also want to know descriptive characteristics about each group of students for further use in tailoring instruction plan and classroom management. Without the conceptual description of clusters, the discovered student groups may not be meaningful or useful for the users.

2.3.2 Conceptual clustering

Conceptual clustering (CC) is a special field of clustering which not only partitions objects into groups as concepts but also provides concepts description (Fisher, 1987; Biswas, Weinberg, Fisher & Douglas, 1998). A concept's description is useful for rule generation and indexing concepts or clusters. CCs build a structure out of data by two manners which are partition and agglomeration. Thus, hierarchical conceptual clustering for concept hierarchy is achieved. The concept hierarchies permit class-subclass relations of clusters to describe knowledge structure in an object space. The class-subclass relations can be defined using conditional probability (Fisher, 1987), for example.

2.3.3 Hierarchical conceptual clustering systems

CC algorithms were usually developed based on similarity measurement of internal inclusion called cohesion and external. One of the well known CC algorithms is COBWEB (Fisher, 1987). COBWEB incrementally organizes objects into a classification tree using a probabilistic concept to describe distributions of objects attribute-values. However, using probabilistic description of a concept fails to convey semantics of the concepts and their relations. An improved version of COBWEB is ITERATE (Biswas, Weinberg, Fisher & Douglas, 1998). ITERATE relaxes the effect of random ordering of objects' attributes by using a feature selection algorithm, then, it iteratively redistributes the objects' attributes to maximize the cluster cohesion. The UNIMEM's algorithm (Lebowitz, 1987) is an incremental CC that uses weights of attribute to reorganize the concept hierarchy.

A clustering algorithm can be viewed as a search algorithm that looks for the "best" groupings of objects among a multitude grouping structures. Therefore, a heuristic function to evaluate certain groupings is necessary to select the best grouping structure. A sequence of selected attribute(s) for partitioning affects the clusters' content, cluster's cohesion, and the hierarchy structure. Gluck & Corter (1985) and Corter & Gluck (1992) presented a probabilistic criterion function called category utility to evaluate such groupings. It is used by COBWEB and ITERATE. The function is used to determine the usefulness or utility of a category. However, the cluster utility function favors large size of clusters over small clusters. If the data contain consecutive similar objects, they tend to go to the same cluster, and as the cluster size increases other less similar object are going to the attracted to this over size cluster, causing skewed cluster structure.

In ID3 algorithm (Quinlan, 1986), a decision tree construction algorithm, a feature selection is achieved through an information gain. COBWEB, ITERATE, and ID3 select one attribute at a time for partitioning. In partitioning based on rough set theory, a subset of attribute selection is allowed. A flexible number of attribute subset selection was studied in (Sonamthiang, Cercone & Naruedomkul, 2007) which will be applied to clustering student data sets in RoughClust.

2.4 Role of student clustering and granular prerequisite concept requirement in student learning

Students in the same classroom can be very diverse and result in instruction to serve student diversity is difficult to achieve. If we apply a hierarchical clustering technique to student clustering which takes into account the student prior knowledge, groups of similar students can be discovered together with information of their prior knowledge requirement for each domain concept. Our clustering approach (Sonamthiang, Cercone & Naruedomkul, 2007) overcomes traditional CC approaches in the sense that the granular concept hierarchical clustering provides not only rich description of student clusters but also the multilevel of granularity of student grouping as the teachers require. The discovered student groups and the groups' information can be benefited for managerial classroom and instruction planning. In addition to student clustering, an approach for student granular prerequisite knowledge requirement mapping is contributed by using the student groups' information with domain prerequisite relations.

In summary, our proposed approach embraces the followings answers the questions:

1. What's new? – Granular prerequisite knowledge requirement mapping.
2. What's different?- Granular solution to uncover students common errors and groups of similar learning difficulty.

3. What’s better?- Our approach is better than existing testing diagnostic systems since it provide granular information in two point of views. The first point is granular group of students based on specific common errors. The second point is granular domain concept hierarchy of the prerequisite knowledge requirement.

4. What’s significant? – Based on the group’s common errors and the domain concept hierarchy, we can provide instruction materials such as handouts, exercises, and tests that are suitable for each group and ready to use by the teachers.

This chapter presents a review of existing research in student prior knowledge diagnosis. Granular computing approach and hierarchical clustering are discussed as alternative approach to uncover student prerequisite knowledge requirement for group or individual students. We close the chapter with an interesting application in student clustering. Clustering students are to find the groups of similar students with common prerequisite requirement. Therefore, teachers and educators can utilize student clusters to improve group or individual students learning achievement. The next chapter, backgrounds in RST which is the theory applied to divisively partition and to define concepts (or clusters) and our rough set-based granular concept hierarchy clustering approach are presented.

CHAPTER III
A ROUGH SET BASED GRANULAR CONCEPT HIERARCHY CLUSTERING APPROACH

As we aim to apply granular approach to cluster students into groups of similar learning difficulty and mapping to granular prerequisite concept requirement, a theory to model such a granular concept of student group and granular prerequisite structure is required. Amongst granular computing techniques (e.g., fuzzy sets, near sets, and shadowed sets) rough set theory (RST) is widely used because of its potential in modeling uncertain knowledge with the simplicity and understandability. This chapter provides background of RST in Section 3.1. We present a hierarchical conceptual clustering approach which is a rough set-based approach to cluster data hierarchically in Section 3.2. Finally, a mapping approach for prerequisite concept requirement is detailed in Section 3.3.

3.1 Background theory on rough sets

This section provides background of RST used in our GCH clustering approach. Sets are one of the most fundamental concepts in mathematics. In conventional sets or crisp sets, an object of a universe is clear to belong or not belong to the sets. As shown in a Venn diagram in Figure 3.1, set *A* and *B*, the intersection of the two sets is made up of the objects contained in both sets. The set of objects that contain in A but do not contain in *B* is represented as *A*-*B*. However, for uncertain, imprecision, fuzzy, or imperfect data sets, defining uncertain objects that may belong to a set is not allowed. In rough sets, certain objects are defined by using a lower approximation and possible objects are defined by an upper approximation of a rough set. Therefore, a boundary region that contains uncertainty objects can also be defined by the differences between the upper and the lower approximation. Figure 3.2 presents an illustrative example of a rough set X in which the set contains a boundary region in the gray area while the lower approximation is in black area. The basic notions of RST that will be referred to in the latter chapters are described as follows.

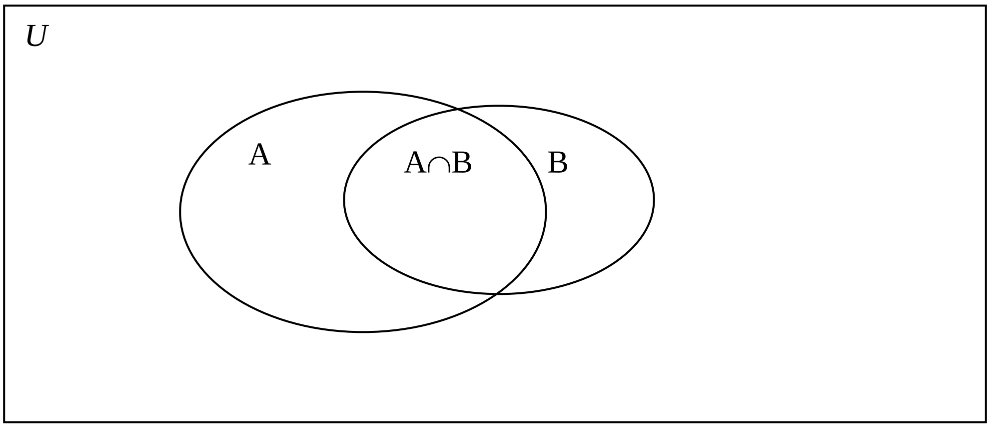


Figure 3.1 A Venn diagram for crisp sets.

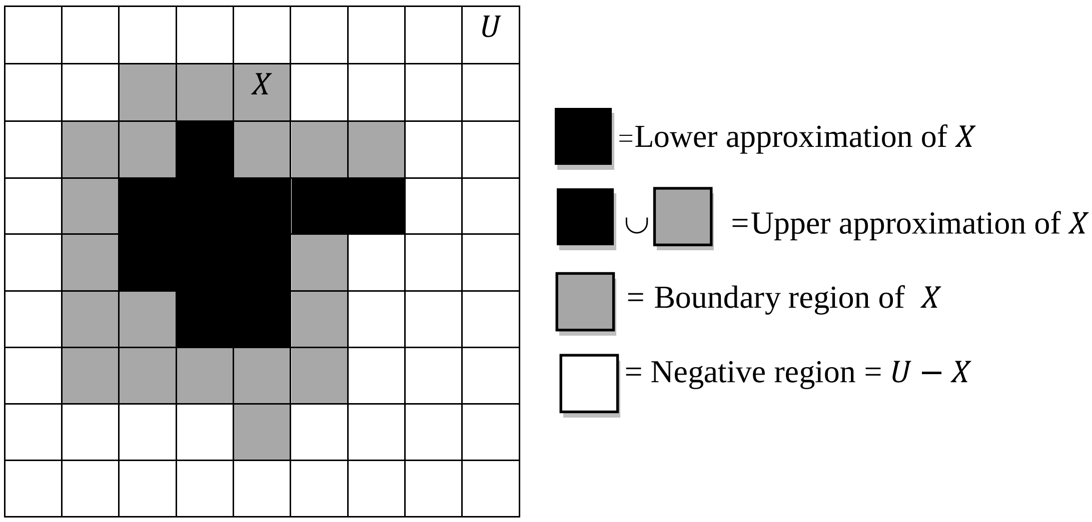


Figure 3.2 A pictorial example of a rough set X

RST was proposed by Professor Zdzislaw Pawlak (1926-2006) in 1982. The RST is to model indiscernible objects and forms a basic granule of knowledge about a domain, based on given observations (Peters, 2008; Skowron & Peters, 2008). However, the observations can be imperfect: inconsistent, insufficient and uncertain. These characteristics of observations, consequently, cause basic granules being rough which are defined as rough sets. Defining rough sets does not require priori probabilistic information about data. Moreover, the rough sets permit induction of rules about uncertainty (Grzymala-Busse, 2004), namely, the certain classes as certain rules, and the uncertain classes as possible rules.

RST provides a formal framework that focuses on both internal relations between attributes' values within an object and external relations between objects. The indiscernibility relation expresses the external relations between objects and the relation can be used to form a granular concepts. In our study, a granular concept represents sufficient information to solve a problem at hand. *How coarse or how specific should a granular concept be to convey such sufficient information?* Some of data clustering techniques such as K-means (Hartigan & Wong, 1979) use a distance matrix to measure similarity of objects within cluster. However, they did not identify how meaningful of each cluster towards target concept that the cluster represent. As a result, it becomes uneasy to interpret and use the cluster semantic in order to solve particular problem. Using rough sets to define a cluster provide insights to what concept is being projected and how accurate of a cluster based on its target concept such as a decision attribute's value.

In information analysis based on RST (Pawlak, 1982; Pawlak, 1991), observations are represented by an information system. An information system S is defined by:

$$S = < U, A, (Va)a \in A, f >, \tag{3.1}$$

where U is a finite and non-empty set of instances called universe. A is a finite and non-empty set of attributes and V_a is a non-empty set of *values* of an attribute a. There is a function f that maps from an attribute of an instance x to a value v_a of V_a such that $<x, a> \in U \times A, f(x, a) \in Va$. A decision table D is a form of an information system S which there is a designated attribute for making decision about objects. Hence, $A = C \cup \{d\}$, where C is called condition attribute set and d is called decision attribute. Formally,

$$D = < U, A, (Va)a \in A, f >. \tag{3.2}$$

The values from mapping *f*(*x*, *d*) are referred to as decision values.

The following notions are applicable for both information system and decision table. Let *x* be an instance of the universe and *X* be a subset of *U*. *X* is called an information granule and is obtained by granulating the set *U* using the indiscernibility relation on an attribute subset *B*, *IND*(*B*), into disjoint subsets of granules. The indiscernibility relation is a relation on $U \times U$ defined for $x, y \in U$ as:

$$(x, y) \in IND(B) \Leftrightarrow f(x, a) = f(y, a), \forall a \in B, B \subseteq A \tag{3.3}$$

The relation *IND*(*B*) is called a *B*-indiscernibility relation and it indicates that x and y are indiscernible by the attribute subset B.

The partitions (or granules) of the universe induced by a subset of attribute *B* is called the equivalence classes of the *B*-indiscernibility relation which is denoted by $[x]_B$. Therefore, an information granule *X* can be characterized by a pair of lower and upper approximations as follows:

$$LOWER(X) = \bigcup [x]_B | x \in U, [x]_B \subseteq X \tag{3.4}$$

$$UPPER(X) = \bigcup [x]_B | x \in U, [x]_B \cap X = \emptyset \tag{3.5}$$

The lower approximation of a granule *X*, or the positive region, is the union of all equivalence classes in $[x]_B$ which are contained in the target set *X*. The upper approximation of a granule *X* is the union of all equivalence classes in which have non-empty intersection with the target set *X*.

The main characteristic that distinguishes a rough set from a conventional set (crisp set) is allowing a set to have a boundary region. The boundary region of a rough set is defined as:

$$BND(X) = UPPER(X) - LOWER(X). \tag{3.6}$$

The accuracy of rough approximation is given by:

$$\alpha(X) = \frac{|LOWER(X)|}{|UPPER(X)|}. \tag{3.7}$$

$|X|$ denotes the cardinality of a set X. The approximation accuracy is in the range of $0 \leq \alpha(x) \leq 1$ and $\alpha(\emptyset) = 1$.

In a decision table, if there is a minimal subset of attributes that is sufficient to describe the decision attribute, this subset of attributes is called a *reduct* (*RED'*).

$$RED \subseteq A \mid [x]_{RED'} = [x]_A, \forall RED' \subset RED. \tag{3.8}$$

The equivalence classes induced by *RED* is the same as the equivalence class induced by full attribute set *A*. There is a non-empty set of attribute that is common to all reducts, this attribute set is defined as *core*.

Core is computed by:

$$CORE = \bigcap \forall RED' \tag{3.9}$$

The core cannot be removed from an information system without effects on the equivalence class structure. One can assume that the core contains essential information to define fine-grained concepts. For this reason, the essence of the core is applied to construct hierarchical granular concepts of the domain. Many studies to compute attributes reduction based on RST have been proposed (Li, Qing, Yang, & Xu, 2004; Bazan & Szczuka, 2005). Different rough set based software may provide multiple reduct generations. In our study, we apply Rough Set Exploration System: RSES (Bazan & Szczuka, 2005) for reducts generation which is suitable in cases of larger data sets for generating representative reducts.

3.2 A rough set-based granular concept hierarchy clustering approach

3.2.1 Formal definition of a granular concept hierarchy

A granular concept hierarchy (GCH) is a hierarchical granular knowledge organization that provides multi-level granular knowledge units, evaluation of knowledge, and knowledge mapping mechanism.

A GCH is formally defined as a quadruple $GCH =< G; R; T; \alpha >$, where G is a non-empty set of nodes, and the nodes themselves are non-empty set.

R is a binary relation of *parent-child* and *child-parent* relation on g. If < g, g' > ∈ R then g is the parent of g' and g' is a child of g. There is a designated element r of G called *root*. The root r holds the universe of elements such that $\neg r = \phi$. A branch BR = $g_0, g_1, g_2, \ldots, g_n$ is the maximal sequence of element of G such that $g_0 = r$ and for every $i \geq 0, < g_i, g_{i+1} > \in R$. Nodes g which $R(g) = \emptyset$ are called leaves. The level of g, denoted by $||g||$, is defined by n if and only if there is a branch $BR = g_0, g_1, g_2, \ldots, g_n$; where $g = g_n$. Obviously, $||g|| = 0$.

T is the target concept of granule which is defined by a set of the decision attribute values.

The α is knowledge evaluation of a granule g. The knowledge evaluation in our approach is defined by accuracy of rough approximation:

$$\alpha(g) = \frac{|LOWER(g)|}{|UPPER(g)|}$$

$LOWER(g)$ is lower approximation and $UPPER(g)$ is upper approximation of a granule g induced by a subset of attribute. $|X|$ denoted the cardinality of a set X. The approximation accuracy is in the range of $0 \leq \alpha(X) \geq 1$, and $\alpha(\emptyset) = 1$.

A GCH comprises of nodes in which the coarsest concept is represented at the root level whereas the most specific concept is represented at the leaf levels. We articulate a concept by using the idea of the most dominant attribute subset: the more dominant degree attribute subset, the more gravity to draw the objects into concepts by that subset. Once a concept is granulated by the most dominant attributes subset, we obtain the more specific concepts which are drawn by common attribute subset. The common attribute subset forms the indiscernibility relations amongst the concept's extension. This structure allows mapping of appropriate granular knowledge in order to solve a problem at hand. The essences of GCH knowledge organization are.

- In order to map to an appropriate granular knowledge, the machine problem solver must identify satisfaction criterions. One of satisfaction criterion is that the granular knowledge is evaluated by sufficient knowledge for solving a particular problem. If the problem is to find decision rules to predict unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them have smaller boundary regions. If the problem is to predict missing values of condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. One may define a satisfaction criterion by setting precision tolerance of applying the granular knowledge. This criterion permits reducing cost of computation where precisiation is expensive or unavailable.

- According to the GCH provides multilevel of granular knowledge ranging from the coarsest level at the root and the most specific level at the leaves, GCH structure provides system of granular knowledge mapping through a tree traversal. Searching for a granular concept in a GCH can be achieved through several techniques such as the depth first search and breadth first search.
- Core attributes are essential to form the more specific concepts since they contain specific characteristics of an object. In GCH construction, core attributes are preserved to retain such specific concepts until the latest granulation.

We shall define the syntax and semantics of GCH and present algorithms to construct a GCH in the next subsection.

3.2.2 Syntax and semantic of a granular concept hierarchy

This section explains what knowledge is represented in the GCH and how to interpret and evaluate knowledge in a granular concept. The section is started by definitions of basic notions, followed by syntax and semantic of a granular concept.

Definition 1. *Let g be a node in a hierarchy of granular concept hierarchy G and g is a decision table, $g \subseteq D$. A common attribute of g is the attribute that forms the indiscernibility relation on $g \times g$. The set of common attribute s is denoted by CA where $CA \subseteq A$.*

Definition 2. *The set of target concepts of g, denoted by* $\tau(g)$, *is defined by the set of decision values in the decision attribute of* $x,\ x \in g$.

$$\tau(g) = \cup\, v_d \mid < x, d > = v_d, \forall x \in g \tag{3.10}$$

Definition 3. *The most dominant target concept,* $\tau^{\wedge}(g)$, *is defined by the decision value of the largest decision class in g.*

Definition 4. *A granular concept description phrase of g, denoted by* $\pi(g)$, *comprises of atomic predicates. A predicate is defined by a pair of common attribute's name and a value of the attribute. Each predicate is conjuncted by the* $\wedge$ *operator to form a phrase.*

$$\pi(g) = ca_0(V_{ca0}) \wedge ca_1(V_{ca1}) \wedge ca_2(V_{ca2}) \wedge \ldots . \wedge ca_n(V_{ca_n}) \tag{3.11}$$

Where $ca_i \in CA$ *and* $|CA| = n, 1 \leq i \leq n$.

Definition 5. *Granular concept description language of g is denoted by* $\lambda(g)$. *The language* $\lambda(g)$ *is generated by traversing H form g0 to g. The phrase of the traversed granules are* $\wedge$ *conjuncted successively to form* $\lambda(g)$.

$$\lambda(g) = \pi(g0) \wedge \pi(g1) \wedge \pi(g2) \wedge \ldots . \pi(g) \tag{3.12}$$

Note that λ(g) is the granular concept's intension.

Definition 6. *Syntax of a granular concept g is denoted by a pair:*

$$\psi < \phi(g), \lambda(g) > \text{ if and only if } x \vDash \lambda(g), \forall x \in g. \tag{3.13}$$

$\phi(g) = \{x | x \in g\}$ is called concept's extensions and every member of ϕ(g) is understood by λ(g).

Definition 7 *Semantics of a granular concept is the accuracy of rough approximation of the granule toward the most dominant target concepts and the concept's intention. The semantics of* g *is denoted by* $\alpha(g)$ *in which*

$$\alpha(g) = \frac{LOWER\ (g)}{UPPER(g)}, \tag{3.14}$$

where

$$LOWER(X) = \bigcup [g]_B \mid x \in g, [g]_B \subseteq g,$$

$$UPPER(X) = \bigcup [g]_B \mid x \in g, [g]_B \cap g \ \neq \emptyset,$$

$$[g]_B = \bigcup \{[< a, v >] | a \in B, B = CA \cup \{d\}\}, f(x, a) = \ \tau\hat{}.$$

A granular concept is indexed by its intension. The granular concept conveys a semantic of being a target concept. To interpret the concept's semantic, one can measure the rough approximation accuracy toward the target concept based on the granular concept intension. Example 3.1 provides an illustrative explanation in the domain of Flu and Example 3.2 introduces an application to student data clustering.

Table 3.1 Flu diagnosis case base

Case	Temperature	Headache	Nausea	Cough	Flu
1	high	yes	no	yes	yes
2	very high	yes	yes	no	yes
3	high	no	no	no	no
4	high	yes	yes	yes	yes
5	normal	yes	no	no	no
6	normal	no	yes	yes	no

Example 3.1 The decision table of Flu diagnosis in Table 3.1 contains four condition attributes of symptoms {*Temperature*, *Headache*, *Nausea*, *Cough*} and one decision attribute {*Flu*}. There are six cases of patient. If the first partitioning is

{*Headache*}, two granular concepts of *g1* and *g2* are obtained as shown in Table 3.2. If the equivalence relation is used to discern patients, there is one common attribute *CA* = {*Headache*} for both *g1* and *g2*. The description language is λ(*g1*) = *Headach*e(yes) and λ(*g2*) = *Headache*(no). The target concept of g1 is having Flu and the semantics conveyed by g1 is the patients who have headache also get flu, with the accuracy of approximation is 3/4. For g2, the target concept is *having no Flu*. The semantics of *having no Flu* is 2/2 of the patients who have no headache. Therefore, the granular concepts are approximated and interpreted to obtain their semantics by the rough accuracy approximation toward the target concept.

Table 3.2 Granulated concepts of Flu diagnosis case base

g1 **:** λ(*g1*) = Headache(yes)					
Case	**Temperature**	**Headache**	**Nausea**	**Cough**	**Flu**
1	high	**yes**	no	yes	yes
2	very high	**yes**	yes	no	yes
4	high	**yes**	yes	yes	yes
5	normal	**yes**	no	no	no
g2 **:** λ(*g2*) = Headache(no)					
3	high	**no**	no	no	no
6	normal	**no**	yes	yes	no

Example 3.2 Student data clustering. Given a test to students in a class, the test is comprised of various concepts in a learning unit or topics and each test item is responsible for testing student knowledge at least one concepts in the domain. If the students' test result are given as shown in Table 3.3, where the first column represent students' ID. Each row represents each student test result and each column represents each test item (Q1 to Q16 in this test). The last column contains student overall performance on the test (e.g. master level=3, intermediate level =2, and poor level=1). We can apply our rough set-based approach to cluster students in this group as follows.

If the first level of clustering applies attribute subset {*Q1, Q2, Q3*} for partitioning, three granular clusters *g1*, *g2* and *g3* are obtained as shown in Table 3.4.

The cluster *g1* and *g3* contains only one student which is the student ID 401 and 405 respectively. If the equivalence relation is used to discern students, the students in this cluster have common attributes *CA* = { *Q1*, *Q2*, *Q3*, *Q6*, *Q8*, *Q10*, *Q14*, *Q16*} for *g2* which means students in this group have common knowledge error in the concepts tested by *Q1*, *Q2*, *Q3*, *Q6*, *Q8*, *Q10*, *Q14*, and *Q16* respectively.

Table 3.3 An example of information system for the student test results

Student ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Performance Level
401	1	1	1	1	0	1	1	0	0	1	0	1	1	0	0	0	3
402	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	2
403	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
404	0	0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	2
405	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1
…	...	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Table 3.4 Granular clusters of students who have similar learning difficulty

Student ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Performance Level
G1 (Q1=1, Q2 =1, Q3=1)																	
401	1	1	1	1	0	1	1	0	0	1	0	1	1	0	0	0	3
G2 (Q1=0, Q2 =0, Q3=0)																	
402	**0**	**0**	**0**	1	0	**0**	1	**0**	1	**0**	1	0	1	**0**	0	**0**	2
403	**0**	**0**	**0**	0	0	**0**	0	**0**	0	**0**	0	1	0	**0**	1	**0**	1
404	**0**	**0**	**0**	1	1	**0**	0	**0**	1	**0**	0	0	1	**0**	1	**0**	2
G3 (Q1=0, Q2 =1, Q3=0)																	
405	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1
…..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..

The description language for each cluster is: $\lambda(g1)=Q1(1)\wedge Q2(1)\wedge Q3(1)$; $\lambda(g2)=Q1(0)\wedge Q2(0)\wedge Q3(0)$; and $\lambda(g3) = \lambda(g2)=Q1(0)\wedge Q2(1)\wedge Q3(0)$. The target concept of *g1* is that the students have performance level=3 and the semantics conveyed by *g1* is the students who answer *Q1*, *Q2*, and *Q3* correctly also at the master performance level, with the accuracy of approximation is 1/1. The target concept of *g2* is that the students have performance level=2 and the semantics conveyed by *g2* is the students who incorrectly answered *Q1*, *Q2*, and *Q3* has master level=2, with the accuracy of approximation is 2/3. The target concept of cluster *g3* is that the student has performance level=1 and the semantics conveyed by *g1* is the student who correctly answered *Q1* and *Q3* but incorrectly answered *Q2* has master level=1, with the accuracy of approximation is 1.

3.2.3 Attribute selection for multilevel partitioning

In order to multilevel partitioning, especially for high-dimensional data, our clustering algorithms focus on subspace clustering. Namely, only some attributes are used at each level partitioning. To select appropriate attribute subset, we concern two aspects of attribute relation which are dependency and correlation between attributes as follows.

3.2.3.1 Applying domain concept hierarchy for attribute selection

When applying our GCH approach to clustering student using an information system as shown in Table 3.3, we found that students in the same cluster is similar to each other by common knowledge errors in the test. However, the primary clustering result shows that in some cluster students who performed common correct or incorrect answers on a set of items are not necessary similar in their performance levels. Therefore, using distance measurement does not convey meaningful clusters for this type of data set.

We considered structure of testing domain which concepts are connected to each other by prerequisite relation. The prerequisite relation is defined between two concepts which absence of understanding one concept affects understanding of other concepts. For example, the basic concept of fraction is a prerequisite of addition of two fractions; therefore, if a student does not hold correct concept about fraction, it will be more difficult to understand the concept of adding two fractions. As a result, order of attribute subsets for partitioning at each level are controlled by domain concept hierarchy such as the prerequisite relations of concepts of a domain.

3.2.3.2 Applying most dominance attribute for attribute selection

When the domain dependency is not available, we design an attribute selection algorithm to order multilevel partitioning. The algorithms compute most dominance degree of attribute subsets at each level of partitioning. The most dominance attribute subset over a considered information granule is selected to compute the equivalence classes for the attribute subset. For more details on most dominance attribute selection, see algorithm 2 in the next section (Granular Concept Hierarchy Construction). A recursive partitioning algorithm is proposed as well as an attribute subset selection algorithm to partition the granularity hierarchically.

3.2.4 Granular concept hierarchy construction

The GCH construction is a recursive granulation by top down partitioning manner. Specifically, the recursive construction is given in Algorithm 1 (Figure 3.3), where the symbols are already defined in Section 3.1. From the algorithm, we obtain a GCH from a recursive tree construction. This algorithm has a recursive function GCHconstruct(g). Input to this function is a decision table and it generates a granular concept hierarchy for the decision table as the output. Before the function is evoked, all variables must be set their initial values as follows:

- Assign the decision table to a granular g ($g \leftarrow D$)
- Assign g_0 as the root of the hierarchy ($g_0 \leftarrow g$)
- Generate a temporary decision table ($TempD \leftarrow D$)
- Set the attribute for partitioning as $\emptyset$ ($B \leftarrow \emptyset$)
- Set of common attributes $CA \leftarrow \emptyset$

Algorithm 1: Granular Concept Hierarchy Construction

Input : a decision table $D = < g, A \cup \{d\}, (V_a)_{a \in A}, V_d, f >$
Output : a granular concept hierarchy, $G = < G, R >$
$g \leftarrow D$
$g_0 \leftarrow g$ // the root of the hierarchy
$TempD \leftarrow D$
$B \leftarrow \emptyset$ // attribute subset for partitioning
$CA \leftarrow \emptyset$ // the set of common attributes
$granulatedStatus(g) = false.$ //mark the granulated concepts

Function GCHconstruct(*g*)
Begin
    if (g is $discernible$) **then**
        1. Find common attribute subset CA of g.
        2. Generate a granule description phrase //See definition 4.
        3. if ($CA \neq \emptyset$) then
            $A \leftarrow A - AC$
            $TempD \leftarrow < g, A \cup \{d\}, (V_a)_{a \in A}, V_d, f >$
        4. Set $B \leftarrow MostDAselect(TempD)$
        //Select the most dominant attribute subset B by Algorithm 2.
        5. Partition the $TempD$ by B, $\{B\} *= \{g_1, g_2, \dots, g_n\}$.
        6. Generate relations $< g, g_1 >, < g, g_2 >, .. < g, g_n > \in R$.
        for all $g_i, < g, g_i > \in R$. **do**
            7. Find $\tau\hat{}(g_i)$.
            //the most dominant target concept of g_i.
            8. Compute semantics of g_i, $\alpha(g_i) = \frac{LOWER\ (g_i)}{UPPER\ (g_i)}$.
        9. $granulatedStatus\ (g_i) \leftarrow false.$
        end
        10. Set $granulatedStatus(g) \leftarrow true$
        11. Set $g \leftarrow g_i$
        12. Set $TempD \leftarrow < g, A \cup \{d\}, (V_a)_{a \in A}, V_d, f >$
    else
        Make a leaf granule.
end
for $all\ g_i$, $(R(g_i) = R(g))$ and $(granulatedStatus(g_i) = false)$ **do**
    GCHconstruct(g_i)
end

Figure 3.3 An algorithm for Granular Concept Hierarchy Construction

Once all variables are defined and imitated, the partitioning process begins with finding common attribute subset. Then, a temporary decision table (*TempD*) is derived from the current decision table by removing the common attributes. The *TempD* is not necessary if there is no common attribute. The attribute sequencing is accomplished through local attributes subset selection in the recursive partitioning. We select the most dominant attribute subset based on the attributes' values available in the decision table. We determine the domination using Algorithm 2. The selected attributes subset is then used to partition *TempD* and assign relationships between the obtained granules (children) and the original granule (parent). If a granule cannot be partitioned by the indiscernibility relation, a leaf node is generated.

Algorithm 2 (Figure 3.4) computes the most dominant attribute subset selection. There are two inputs to this algorithm which are a temporary decision table (*TempD*) for local attribute subset selection and CORE attribute for the universe of the data set. Please note that our algorithm does not compute CORE. The rough set exploration system (RSES version 2.2) (Bazan & Szczuka, 2005) can be used to calculate reducts of the universe. Then CORE can be derived from intersection of all reducts. In addition to the two inputs, there are two parameters required which are N (number of maximum attributes in the attribute subset B) and a threshold ε (co-occurrence proportion between the N attributes' values and a decision value).

Given a *TempD*, we find the N most dominant attributes toward the decision class. CORE is used to preserve the specific attribute(s) of instances in the granule by retaining CORE until the latest granulations. N can be tuned up to the number of condition attributes to compose a good proportion of the concept's extension. In other words, our algorithm allows a flexible number of attributes in a subset for partitioning.

We use co-occurrence proportion of attributes' values and decision classes (correlations) to determine the domination degree. Once the most N dominant attributes are obtained, we determine the co-occurrences within the N attributes to find if any combination of them can be used to approximate a concept by a threshold (ε). A count of co-occurrence amongst condition attributes' values implies the degree of which these attribute values can be used to compose a common concept. We can tune the threshold ε by the number of instances in working granule. For example, if there are 10 instances in the working granule and 5 instances are indiscernible by the attribute subset B, then the domination degree of B is 5/10. The subset of attributes with the greatest domination degree $argmax\,(|[x]_{a \cup d}|)$ and the greatest domination degree is greater than the threshold $argmax(|[x]_{TopB}|) > \varepsilon$ is selected to partition the current granule. If no domination degree of the N combination attributes meets the threshold, the single most dominant attribute is selected.

Algorithm 2: Most Dominant Attribute Subset Selection

Input : a decision table ($TempD$), $CORE$, parameter N and threshold ε.
Output : the most dominant attribute subset B
// this attribute subset will be used to partition the $TempD$ in Algorithm 1.

$MostDA \leftarrow \emptyset$
$TopDA \leftarrow \emptyset$
$B \leftarrow \emptyset$
for $each\ a \in A$ **do**
    for $each\ v_a \in V_a$ **do**
        1. $[x]_{a \cup d} \leftarrow \bigcup\{[a,v] | f(x,a) = v_a, f(x,d) = v_d\}$
    end
    2. $domDegree(a_i) \leftarrow argmax\ (|[x]_{a \cup d}|)$
end
$MostDA \leftarrow argmax\ (domDegree(a_i))$
$TopDA \leftarrow TopN\ argmax\ (domDegree(a_i))$
$B \leftarrow MostDA$
for $each\ TopB\ where\ TopB \subseteq TopDA,\ |TopB| > 1$ **do**
    $[[x]_{TopB} \leftarrow \bigcup\{[a,v]\ | f(x,a) = v_a, a \in TopB\}$
    if $argmax(|[x]_{TopB}|) > \varepsilon$ **then**
        $B \leftarrow topB$
    end
end
if $(B - CORE \neq \emptyset)$ **then**
    $B \leftarrow (B - CORE)$
end
Return B

Figure 3.4 An algorithm for attribute subset selection based on most dominance attribute subset

Example 3.3 illustrates the recursive construction of a GCH using Algorithm 1 and Algorithm 2 in Figure 3.3 and Figure 3.4 respectively.

Example 3.3. A GCH construction for the Flu diagnosis decision table (Table 3.1.) is described step by step. The granulation starts by partitioning the universe. In this example, the equivalence relation is used. The size of attribute subset to partition is one (N = 1) since the number of condition attributes is relatively small. The objects in the universe are discernible by the equivalence relation. Thus, we find reducts for this table which are {*Temperature, Headache, Nausea*}, {*Temperature, Nausea, Cough*}, and {*Headache, Nausea, Cough*}, and core is {*Nausea*}. There is no common attribute value in this granule. We select the first attribute subset by determining the degree of attribute dominations. Headache has the highest domination degree (*domDegree* = 3) compared to the rest of the condition attributes (*domDegree* = 2). Thus, the first attribute subset to partition is {*Headache*} and *g1* = {1; 2; 4; 5} and *g2* = {3; 6} are obtained. Then we continue granulate g1 selecting the most dominant attributes for *g1*. *Temperature, Nausea* and *Cough* attributes have the same degree of domination. *Nausea* is the core; thus, it is retained at this granulation. We select *Temperature* or *Cough* to partition g1. If we apply *Temperature*, we obtain granule *g3* = {1; 4}, *g4* = {5}, *g5* = {2} which are children of *g1*. The granule *g4* and *g5* are indiscernible so they are leaf granule. We then granulate *g3* by finding common attribute subset which is {*Cough*}. The Cough attribute can be now removed. The remaining attribute {*Nausea*} is then used to partition *g3* to obtain *g6* = {2}; g7 = {3}. Since all siblings are now leaf nodes we can return to the higher levels. We continue granulate *g2*. Note that the temporary table can be generated as the common attribute {*Headache*} is neglected. Like partitioning *g1*, Nausea is retained. If we partition *g2* by *Temperature*, the indiscernible granule *g8* = {3} and *g9* = {6} are obtained. Fig. 3.5 shows the GCH for the Flu diagnosis domain.

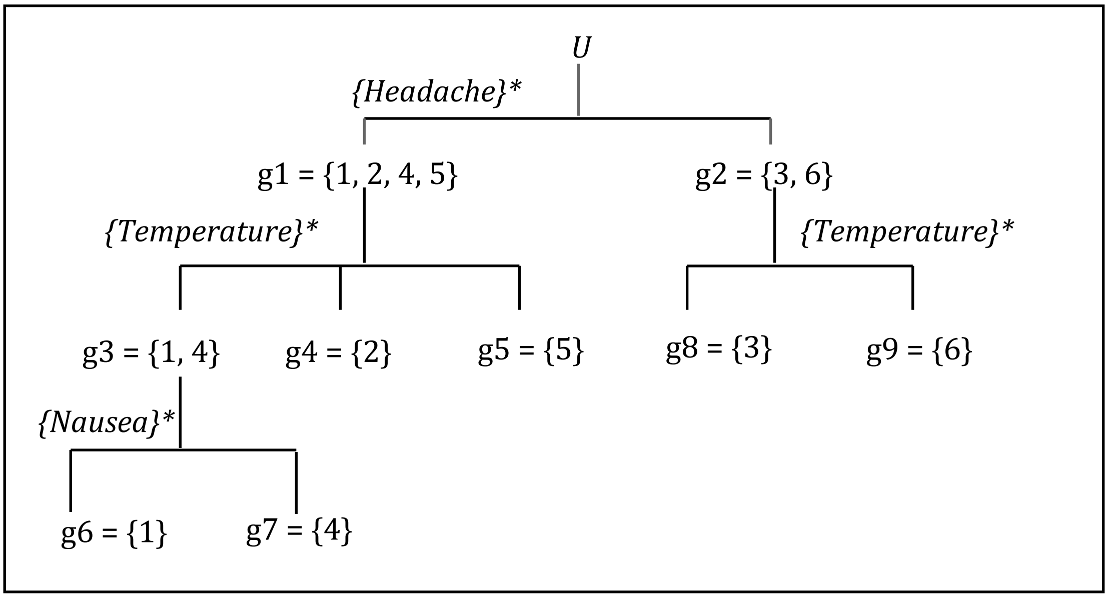


Figure 3.5 A granular concept hierarchy for the flu case base

3.3 Prerequisite concept requirement mapping

One of our goals is to help teachers understand what prior knowledge is lacking and causes students' learning problems. Therefore, in addition to student clustering, we also implement further application for mapping the group prerequisite concept requirement. The prerequisite concept requirement is composed of a list of concepts should be well learned prior to learning a specific concept. In our study, the prerequisite concepts requirement is derived from mapping the student group's characteristics with the prerequisite relations between concepts within a domain. A learning domain can be broadening from a specific learning concept, topic, unit, lesson, and a course. A well known approach for defining the concept dependency relationship is CER model, proposed by (Hwang et. al., 2003). Please refer to Chapter 2, Section 2.2.2, for more details of the CER model. In (Hwang et. al., 2003), the authors defined the prerequisite relations as follows:

> *Consider two concepts to be learned; Ci and Cj, If Ci is a prerequisite to efficiently performing the more complex and higher level concept Cj, then a concept-effect relationship Ci→Cj is said to exist. Notably, a concept may have multiple prerequisite concepts, and a given concept can also be a prerequisite concept of multiple concepts.*

The prerequisite structure of a domain is defined as relationships between two units within the domain where knowledge *Ci* and *Cj* is dependent. The model represents prerequisite relationships among concepts in a course which with the relationships are used to diagnose individual students' learning problems and provide individual learning advice. We borrowed the CER model to indentifying the group-based prior knowledge requirement by mapping the student clusters' characteristic with the prerequisite relationships amongst concepts.

An example of a CER model in the domain of fraction is defined in Figure 3.6. The concept whole number is a prerequisite of the basic concepts in Fraction. Once the student comprehend fractions, then he hold the prerequisite knowledge for comparing two fractions with which one is equal, less than, or greater than the other one and so on.

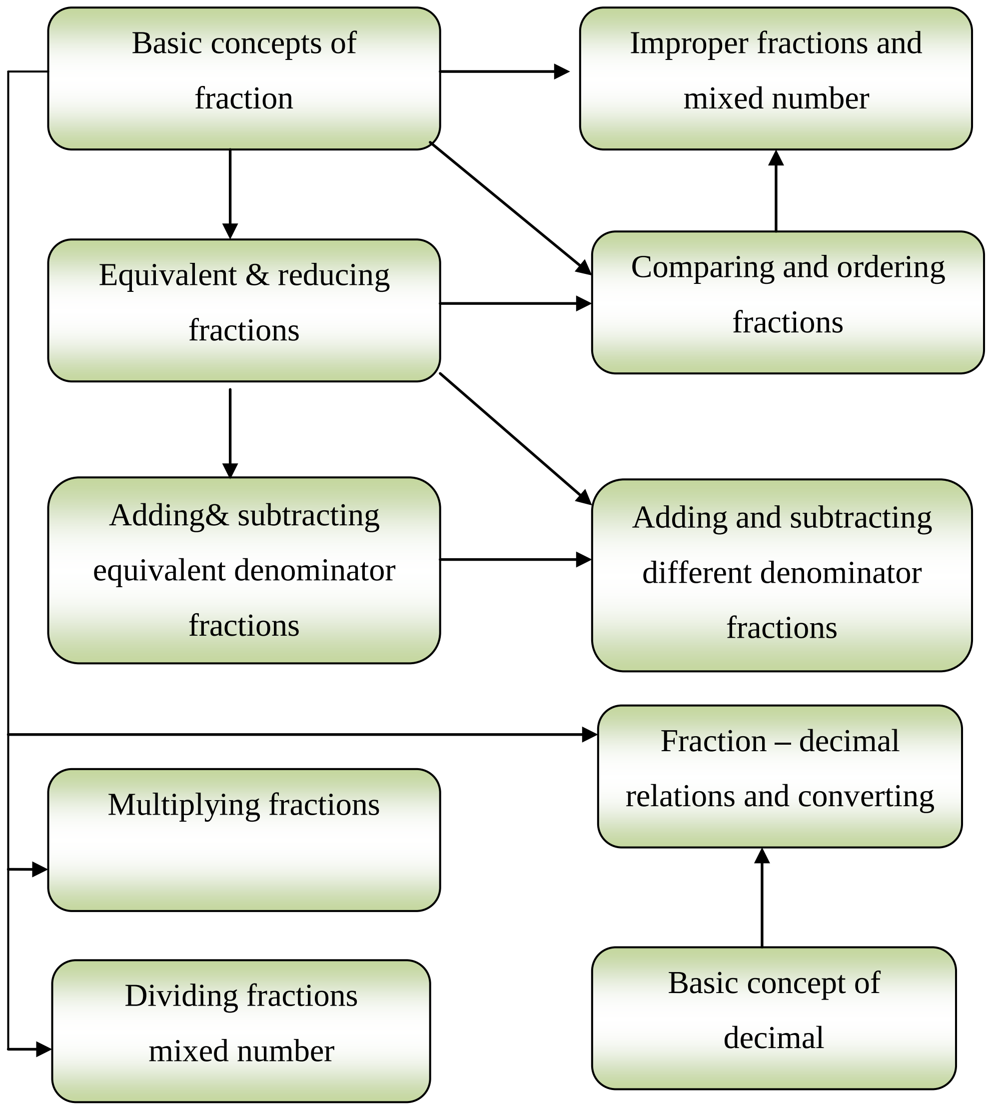


Figure 3.6 A CER model of fraction concepts for primary level

In this chapter, we present background theory of RST and our approach for hierarchical clustering. The decision class is used to guide clustering accuracy and the obtained clusters are defined by rough set-based lower and upper approximation. A recursive algorithm to construct a GCH and algorithm for attribute selection for multi-level partitioning are presented. Moreover, we discussed that some domain should apply the domain dependency (e.g., domain prerequisite relations) to guide multi-level partitioning such as in the student clustering based on an academic test. Next chapter presents an application of the GCH clustering approach to cluster the real world data sets, student prior knowledge testing. Moreover, utilizing from the student clusters is studied through the group-based prerequisite requirement mapping and providing the instruction material for the group’s requirement.

CHAPTER IV
ROUGHCLUST: A STUDENT CLUSTERING AND PREREQUISITE CONCEPT REQUIREMENT MAPPING TOOL

In this chapter, we present an educational application of our GCH clustering approach to group students using the student data from prior knowledge testing and learning performance level. To reach our objectives of student clustering and mapping to prerequisite concept requirement, we design the architecture of our system called *RoughClust* as presented in Section 4.1. Section 4.2 describes input of the system which is collection of student test data sets and data preprocessing and preparation. In Section 4.3, user interface and how to use RoughClust is guided. Interpretation of the output information provided by our system (students' groups, group-based learning difficulty, prerequisite concept requirement, and instruction material) is also suggested. In addition, RoughClust knowledgebase can be extended by the users; therefore, we also present the user interface for extending RoughClust in this section.

4.1 RoughClust architecture

RoughClust is an automatic system for student clustering and prerequisite knowledge requirement mapping. The GCH clustering approach described in Chapter 3 is applied to clustering student data sets in RoughClust. The design of RoughClust (Figure 4.1) offers teachers an alternative approach to group students based on their similar prerequisite knowledge requirement. Using RoughClust, the teachers are provided with the student groups' characteristics, prior knowledge information, and instructional materials related to each group requirement. Moreover, RoughClust is designed to work online in order to facilitate teachers anytime and anywhere.

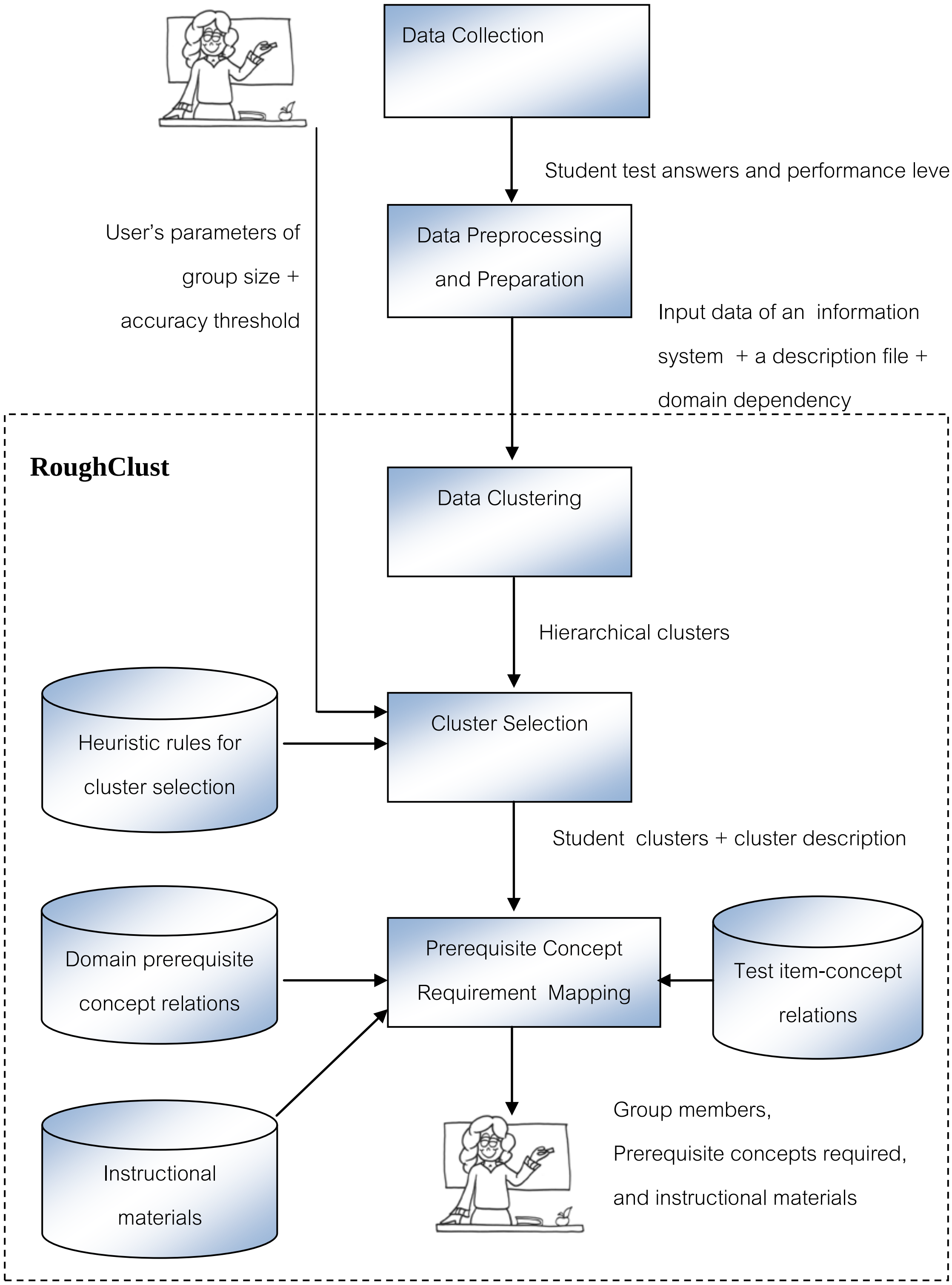


Figure 4.1 Architecture of RoughClust

From the RoughClust architecture, the system requires input data which must be provided by teachers as follows.

1. Data collection. Data is obtained from students' answers of a test to evaluate student background knowledge. Moreover, collection of student grade records of the previous years is valuable to reflect their overall academic performance level in interested subject. Data collection used in our experiment is presented in section 4.2.1.

2. Data preprocessing and preparation. The data collected from students must be preprocessed. The preprocessing is required since the data can be out-of-range values, impossible data combinations, and missing values. Analyzing data that has not been carefully screened for such problems can produce misleading results. Preprocessing data and preparing input to RoughClust are presented in section 4.2.2.

From Figure 4.1, RoughClust performs three main modules: Data Clustering, Cluster Selection, and Prerequisite Concept Requirement Mapping. Moreover, RoughClust requires four knowledge bases which are Heuristic Rules for Cluster Selection, Domain Prerequisite Concept Relations, Test Item-Concept Relations, and an Instructional Materials database. Tasks of each module are as follows.

1. Data Clustering. The student data set is clustered applying the GCH clustering algorithms based on rough sets (Sonamthiang, Cercone, Naruedomkul, 2007; Sonamthiang, Naruedomkul, Cercone, in press) as described in Chapter 3. Using the system to clustering a student data set is thoroughly guided by the user interface illustrated in section 4.3

2. Cluster Selection. The hierarchical clusters provides various levels of granular clusters and clusters at each level represent different degree of specificness/coarseness, accuracy, coverage, and cluster size. Moreover, information provided by the clusters includes boundary region and intra-cluster similarity. Therefore, we use heuristics together with parameters from teachers to select appropriate student clusters. Teachers can adjust parameters which are group size and clustering accuracy threshold as desire.

3. Prerequisite Concept Requirement Mapping. Once student clusters are discovered, RoughClust provides the clusters' description which contains information

about common knowledge error of each group. Therefore, we use the common knowledge errors with the Domain Prerequisite Concept Relations for mapping to granular prerequisite concept requirement of each group or individual student. Moreover, once the group requirement is revealed, the instruction material specific for the group requirement can be queried from the Instructional Materials database.

RoughClust was implemented using the JavaServer Pages (JSP) and Java Servlet technology. The RoughClust clustering module runs at the server side and the RoughClust databases were implemented deploying MySQL database management system.

4.2 Input of RoughClust

4.2.1 Collecting student prior knowledge

Knowledge about student prior knowledge will be collected as we would like to understand the student learning difficulty for treating each group of similar students more suitable. In our study, student prior knowledge is acquired from two sources which are prior knowledge testing and learning performance as described below.

Prior knowledge: A test is used to assess student prior knowledge. The test must be well design to be able to reflect student prior knowledge of the domain. In our study, we experimented on data sets obtained from student prior knowledge testing towards Fractions for primary level. Please see an example of a pretest in Appendix C. A number of 89 students who were attending grade 4, 5, and 6 in academic year of 2010 participated in this study. This student test result is given as an information system in Appendix B.1. The second group is students in grade 6 in the same school but in different academic year. They were tested Fraction prior knowledge using the same test as the first group. There are 30 students in this data set.

Learning performance: The student learning performance can be gathered from the grade record from the previous semester or academic years in the tested subject. If the student grade is not available, we use total score of the test to grading student learning performance on the testing domain and use the grading as the learning performance level in the same manner as using the grade record.

User parameters: An objective of student clustering is to group students who have similar learning problems in similar concepts with which the group characteristics can be used to map for their learning needs. RoughClust produces a hierarchy of student clusters which provides various levels of granular clusters. Clusters at each level represent some degree of specificness/coarseness from space cluster to indiscernible clusters. An important issue to hierarchical clustering analysis is to determine which level of granularity is optimal to solve problems. Finding high-quality clusters, the boundary region is low, may guarantee student clustering granularity level as well as the relative size of clusters. Therefore, we recommended the following heuristics to be used in selecting appropriate student clusters for further mapping application:

1. Relative size of clusters: size of student clusters is depending on users' (teachers and educators) preference. Small size of cluster (5-10 students) is more advisable for arranging within class grouping instruction whereas cluster size = 1 represents individual student prior knowledge.

2. The accuracy threshold is defined by rough accuracy approximation (please refer to definition 7, page 33 in Chapter 3). The value of rough accuracy is in a range [0, 1].

RoughClust requires two parameters from the users to select appropriate granular students clusters. The first parameter is the maximum group size that defines the size of groups according to users' desire in their group-based instruction planning. The value of maximum group size parameter is an integer in the range [1, N] where N is number of students in the data set. This parameter will be used in the Cluster Selection Module to select the clusters that the number of clusters' members is not exceed this parameter and is in corresponding with the second parameter, the accuracy threshold. Please note that the value of accuracy threshold is a decimal in the range [0, 1]. The greater accuracy threshold, the more cohesive clusters are selected.

Once a student test and learning performance level are collected, teachers can prepare input data to RoughClust by checking and scoring the student test and store the test result in a data file for further use. However, it is possible to have missing value or value out of range data in the data set. These characteristics of data can lead to incorrect clustering result. As a result, preprocessing and preparation tasks are essential to provide best input to the system.

4.2.2 Data preprocessing and preparation

The test results and learning performance level are formatted in an information system as an example shown in Table 4.1. The head columns or attributes represent the student ID, followed by the 16 test items and the performance level in the last column. The attribute values for the student ID are nominal. The attribute values for the test items are binary (0 or 1), 0 is incorrect and 1 is correct answer for each test item. The performance level attribute is the decision attribute which is numeric values of student performance level on the test ranging from level 1 to level 4, for instance. Level 1 represents less performance and level 4 represents most performance respectively. In this step, teachers should thoroughly check the data to avoid data missing and data out of range.

Table 4.1 An example of information system for the student test results

Student ID	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15**	**Q16**	**Performance Level**
401	1	1	1	1	0	1	1	0	0	1	0	1	1	0	0	0	3
402	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	2
403	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
404	0	0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	2
405	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1
…	...	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

In practical, an information system can be stored using the comma-separated values (CSV) file format for reuse. A CSV file simply represents values of tabular data by using lines in a text file to represent rows in a table, and commas separate the columns. Formatting a CSV file is an illustrated in Figure 4.2. Then users can save the information system as an input data file: FILENAME.csv.

RoughClust requires the data description to check the data consistency of a data set. Therefore, a description for the data set must be defined. In RoughClust, the description file's extension is FILENAME.fmf where FILENAME is the same as the data file. The description file is formatted as shown in Figure 4.3. In the first row, *X* represent objects, followed by column name, data type, number of objects, and the object identity list. The next rows represent each conditional attributes *C*, followed by column name, data type, number of possible values, and list of all possible value of the conditional attribute. In the last row the decision class is represented by indicating "class" as column name, followed by data type, number of possible classes, and list of all possible classes.

```
401,1,1,1,1,0,1,1,0,0,1,0,1,1,0,0,0,3
402,0,0,0,1,0,0,1,0,1,0,1,0,1,0,0,0,2
403,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1
404,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,2
405,0,1,0,1,0,0,0,0,0,0,0,1,0,0,1,0,1
406,1,0,0,1,0,0,0,0,1,0,0,1,1,0,1,0,2
```

Figure 4.2 An example of the comma separate value file

```
<X 0 ColumnName DataType Number_n value#1 value#2…. value#n>
<C 0 ColumnName DataType Number_i value#1 value#2…. value#i >
<C 0 ColumnName DataType Number_j value#1 value#2…. value#j >
…….
<C 0 ColumnName DataType Number_k value#1 value#2 ….value#k >
<D 0 class DataType Number_d value#1 value#2…value#d>
```

Figure 4.3 Description of a data set stored in *.FMF extension

A concrete example of a data set description is given in Figure 4.4. In the first line, the description shows that this data set contains 27 instances of students and each student label (student ID) is listed after the number (e.g., *601* to *629*). The consequence lines represent each conditional attribute (indicated by *C*), e.g. named *item1*, and the data type of this attribute value is treated as string (*S*). There are two possible values of this condition attribute which are *0* and *1*. The last line indicates the class or decision attribute. In this data set, the decision attribute is the student learning performance level. There are 4 possible attribute values of the decision class which are 1, 2, 3, and 4 respectively.

```
<X 0 Students S 27 601 602 603 604 605 606 607 608 609 610 611 612
613 614 615 616 617 618 619 620 621 622 623 624 625 627 629>
<C 0 item1 S 2 0 1>
<C 0 item2 S 2 0 1>
<C 0 item3 S 2 0 1>
……
<D 0 class S 4 1 2 3 4 >
```

Figure 4.4 An example a data set's description

In addition to the data set information system and the data description, RoughClust also requires the user to define the domain dependency of the attributes, or attribute dependency. The attribute dependency will be used by the Data Clustering Module in attribute selection for multilevel partitioning the student data. Our data set is taken from a pretest which each item of the test is treated as a conditional attribute. Therefore, the attribute dependency can be defined by the attributes orders of the domain prerequisite relationships of concepts. User can use a text editor to define the attribute dependency as the following. In the first line, a list of items' name separated by a comma (,) represent items that are related to the most general concept. Then, use a line feed to separate the more specific concepts in the latter lines. The domain dependency is then saved as FILENAME.coa to be input to RoughClust. An example of the attribute dependency for the pretest in Appendix C is given in Figure 4.5. In the first level partitioning, RoughClust will take *item1* and *item13* to consideration followed by *item2* and *item3* for the second level partitioning and so on.

```
item1,item13
item2,item3
item4,item9
item10,item11
item5,item7
item6,item8
item15,item16
item9
item12
```

Figure 4.5 An example of attribute dependency of a pretest

4.3 User interface design

In designing user interface of RoughClust, we focus on simplicity of use and extendibility of the knowledgebase. RoughClust also allows users to flexibly adjust their input parameters which are cluster size and acceptable accuracy threshold. Graphical user interface for using RoughClust in clustering a student data set is presented in Section 4.3.1. Then, interpretation of the output from RoughClust is clarified in Section 4.3.2. In addition to clustering, users can easily modify, maintain, and extend databases in RoughClust which are the concepts, the concept prerequisite relationships, and instructional material database as described in Section 4.3.3.

4.3.1 User interface for student clustering and prerequisite concept requirement mapping

Once the input data is well prepared, users can use RoughClust online by firstly login to RoughClust as shown in the welcome page in Figure 4.6. This welcome page contains an introduction to RoughClust and its documentation. The user can download RoughClust's documentation without logging in to the system. Once the user logged in, then the user can cluster a student data set by the following steps (Figure 4.7).

1) Select a pretest that is already stored in the Domain Prerequisite Relation database in RoughClust. If the pretest does not exist, the user must define the pretest in Section 4.3.3.

2) Upload the input data (FILENAME.csv), the data description (FILENAME.fmf), and the attribute dependency (FILENAME.coa) to the system.

3) Set user's parameters which are maximum cluster size and accuracy threshold.

4) Process student clustering and prerequisite concept requirement mapping by clicking the "Process" button.

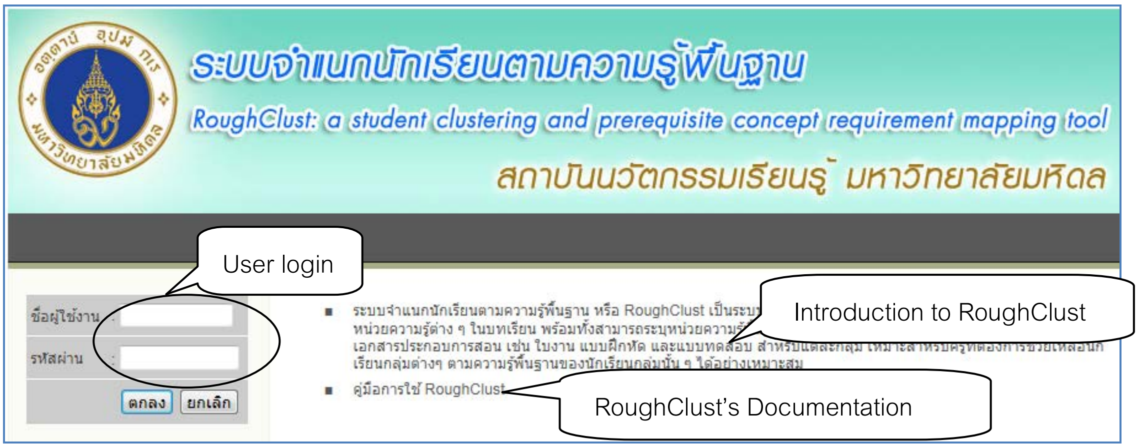


Figure 4.6 RoughClust's welcome page

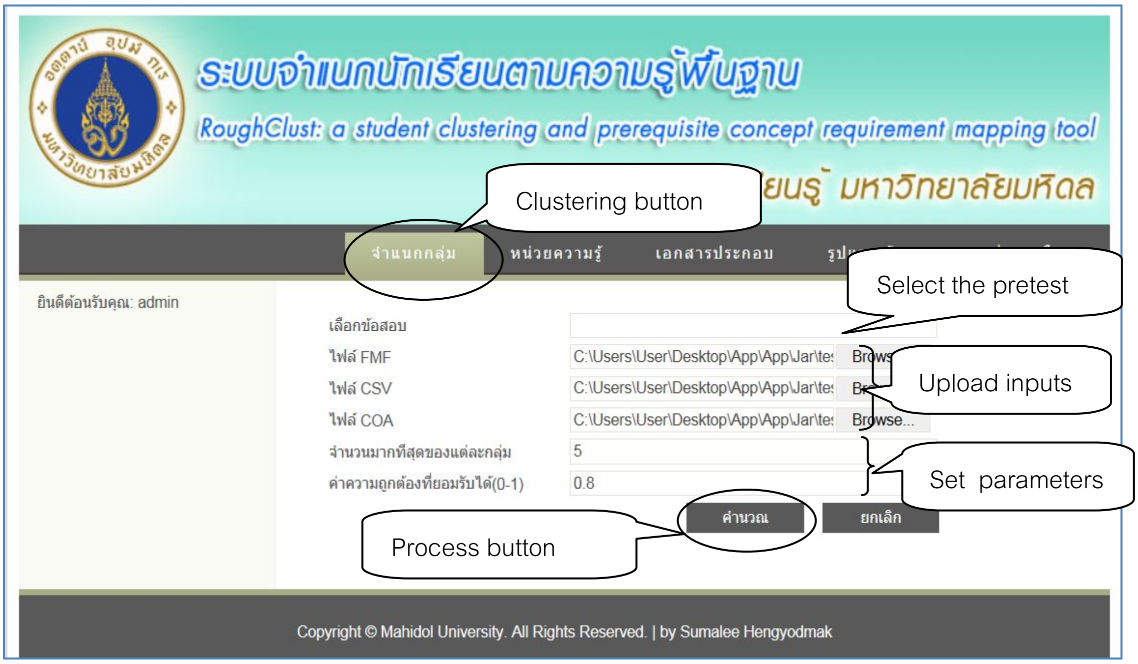


Figure 4.7 The input window to upload input data sets and define parameters

The user is then obtained output from RoughClust as depicted in Figure 4.8. The output contains number of groups obtained and detail of each group characteristics and requirement. The group's characteristics comprise of the following four sets:

1) a list of students' IDs who are assigned to the group,

2) a list of common concepts that are incorrectly answered by the student in this group,

3) a list of the prerequisite concepts required for this group, and

4) a list of instruction materials that is prompt for the teacher to download and print in order to use in planning the group instruction. An example of a instructional material is shown in Figure 4.9.

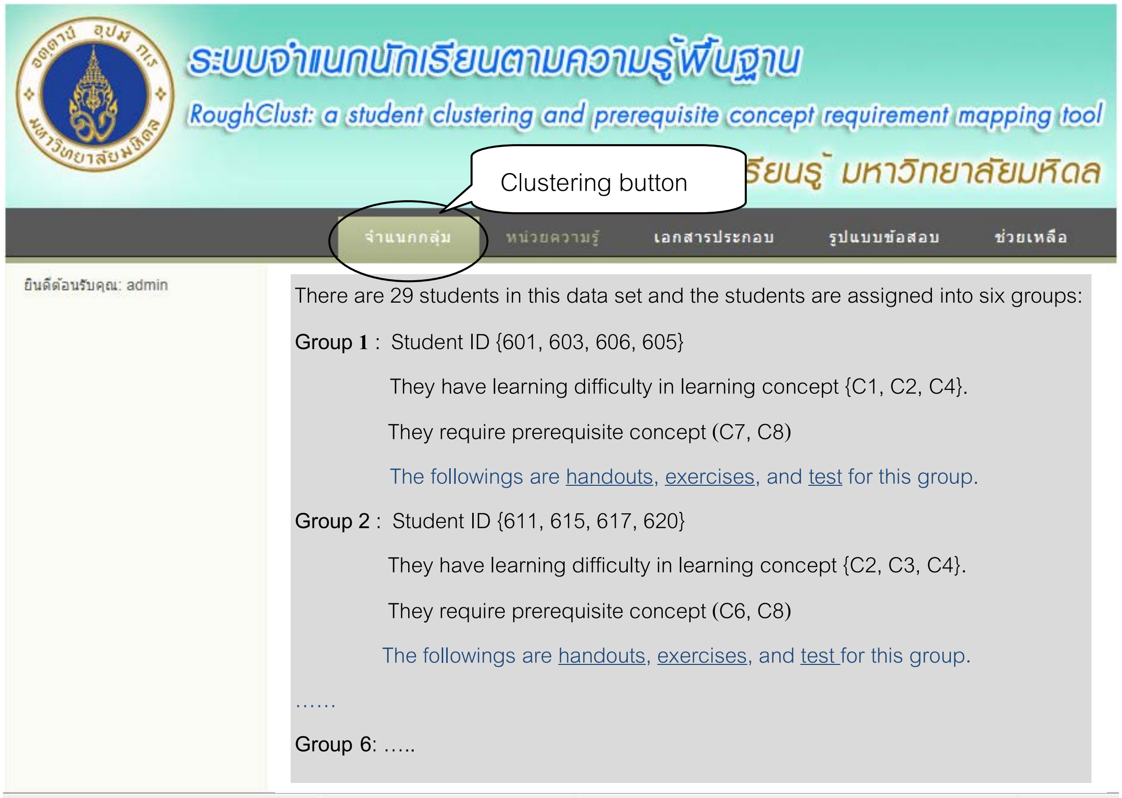


Figure 4.8 An example output of RoughClust (translated from Thai to English)

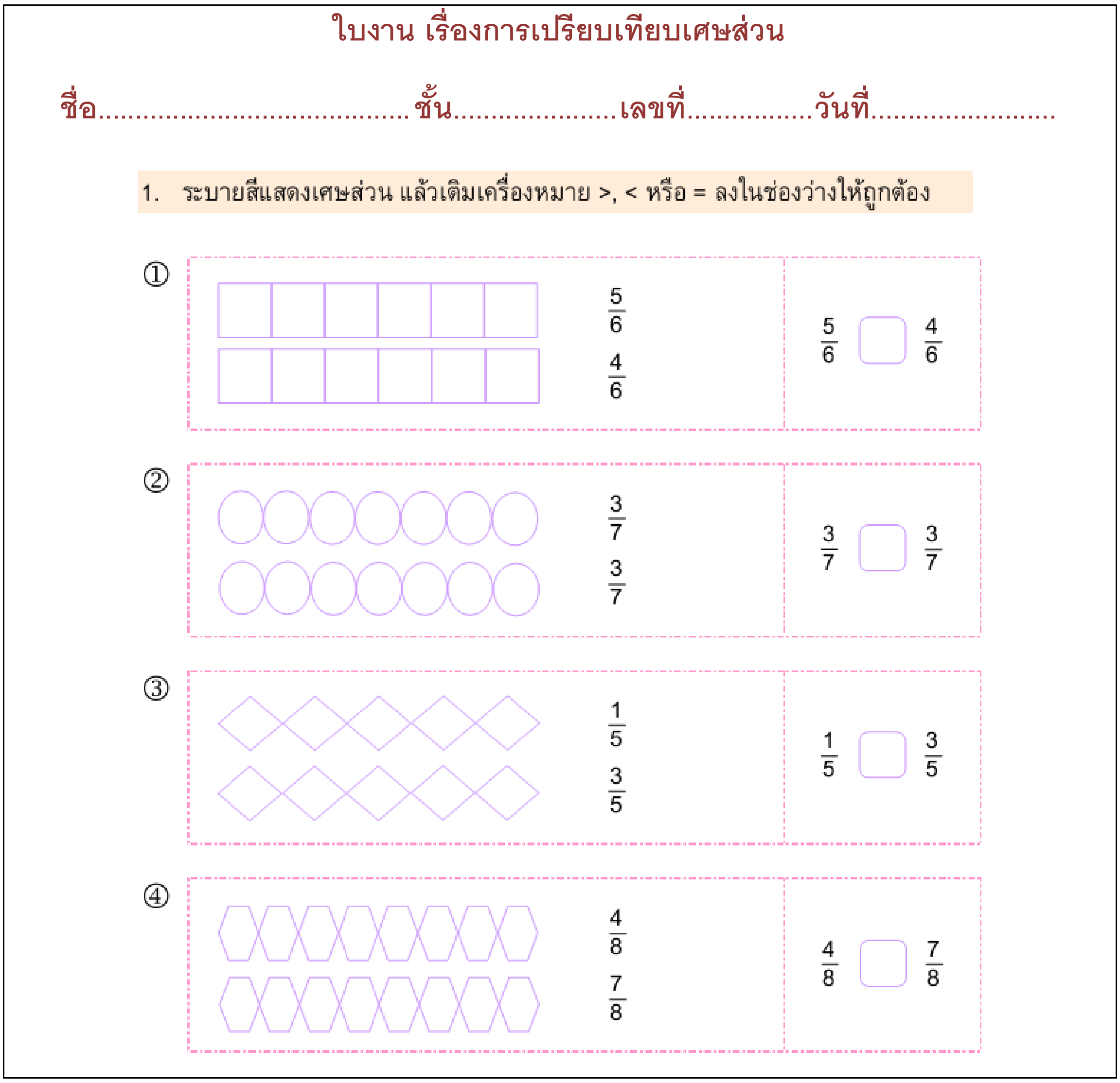


Figure 4.9 An example handout for teaching comparing fractions

4.3.2 Output interpretation

The output from RoughClust contains information in four aspects for the teachers which are: students' groups, group-based learning difficulty, prerequisite concept requirement, and instruction materials. From Figure 4.10, teachers can interpret the output as described using an illustrative example as follows.

1) Students' groups: this information indicates number of groups derived from all students in the input data set. Moreover, the group contains a list of each student as a group member by student ID (or name as indicated in the input data). For

example, there are six groups obtained from clustering 29 students. In Group 1, there are four students in this group which their student IDs are 601, 603, 606, and 605.

2) Group-based learning difficulty: this information shows the common learning difficulty of the concepts that the students in the group perform errors or incorrectly answer the related test items. It comprises of the list of concept names which are related to the incorrect items. For example, students in Group 1 have common learning difficulty in concept *C1*, *C2* and *C4* respectively.

3) Prerequisite concept requirement: in addition to group based common learning difficulty, RoughClust also provides the teachers with a list of prerequisite concepts. It is advisable that the students in the group must review these prerequisite concepts in order to learn the concepts that they have learning difficulty. For students in Group 1, their prerequisite concepts *C7* and *C8* required.

4) Instruction materials: to facilitate teachers, RoughClust suggests various instruction materials suitable for each group learning difficulty and prerequisite concept requirement. Teachers can get the instruction materials in just one click for a handout, exercise, and test to use in their instruction planning.

4.3.3 Extending RoughClust's knowledgebase

RoughClust knowledgebase is extendable. Thus, domain knowledge of RoughClust is not only limited to Fraction for primary student level. Users can add new concepts, modify the existing concepts, and create prerequisite relationships between the concepts. Moreover, the user is also allowed to extend and modify the instruction material database. This section describes step by step of how to extend RoughClust's knowledge base.

4.3.1.1 Adding and modifying a concept and defining a group of concepts

If the user desires to add a more basic concept, for example "comparing two numbers" concept, the user can firstly click on the "Defining concepts" button on the top menu. Secondly, clicking on the "Create a new concept" menu will navigate the user to the screen in Figure 4.10. Three information about the new concept are required which are the concept group' name, the new concept ID, and its description. The new concept must have its own group in order to define its requisite relations. Then, the user clicks on the save button to create the new concept and store in the Domain Concept Relations database.

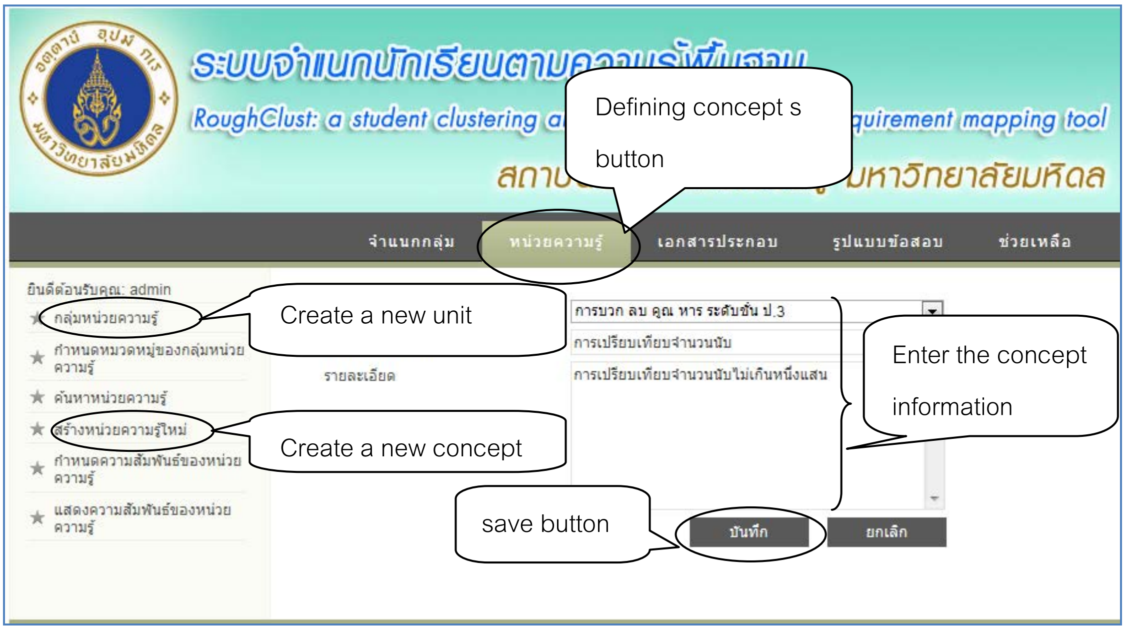


Figure 4.10 The screen to add a new concept to the knowledgebase

If the user wants to create a new group of concepts called "unit", he can select "create a new unit" menu. A new unit cannot be created without having no concept inside. Then, similar to creating a new concept, the user must enter the new unit name, the new unit ID, and its description. Figure 4.11 shows an example of "Fraction for grade 4 unit" after it was created. This unit is composed of six concepts which are meaning of fraction, reading fractions, writing fractions, comparing fractions with the same denominators, sequencing fractions with the same denominators, and adding and subtracting fractions with the same denominators.

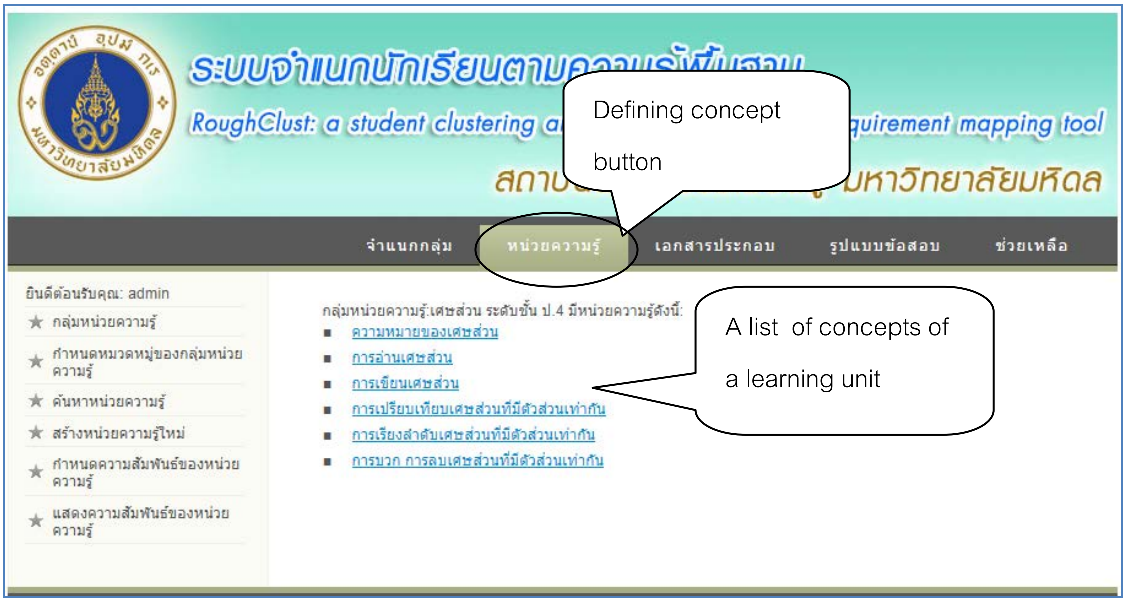


Figure 4.11 A learning unit “fraction for grade 4” and its composition of six concepts

4.3.1.2 Creating the concept relationships

Users can define the prerequisite concept relations for a specific concept by selecting “creating concept relationships” menu, and then choosing the existing concepts to add or modify prerequisite concepts. For example, in Figure 4.12, the concept “comparing fractions with the same denominators” is selected to defining its prerequisite concepts. All the concepts in the same concept’s groups are shown to the user to select what are prerequisite concepts of “comparing fractions with the same denominators.” The user can select a check box to include a particular concept as a prerequisite concept and click “save” button to create the relationships.

Figure 4.12 An example of defining prerequisite concepts for a concept

4.3.1.3 Adding a new pretest and defining item-concept relationships

RoughClust allows the user to store pretests and its domain dependency of item-concept relationships for reuse. Therefore, the user can add and define a new pretest by selecting the "pretest specification" button from the top menu and then select the "create new pretest" button from the left menu. The user will be presented by a screen to define the pretest' category, name, the printout file directory, number of items, and the test description as depicted in Figure 4.13. Clicking on the "process" button, the new pretest is created and the user will be navigated to the next screen in Figure 4.14. This screen shows a table which represents each item of this pretest in the head column and each concept to be tested in each row. User can click on the check boxes to indicate the item-concept relations which define the concepts that affect answering the test items. These relations are used by RoughClust's Prerequisite Concept Mapping module.

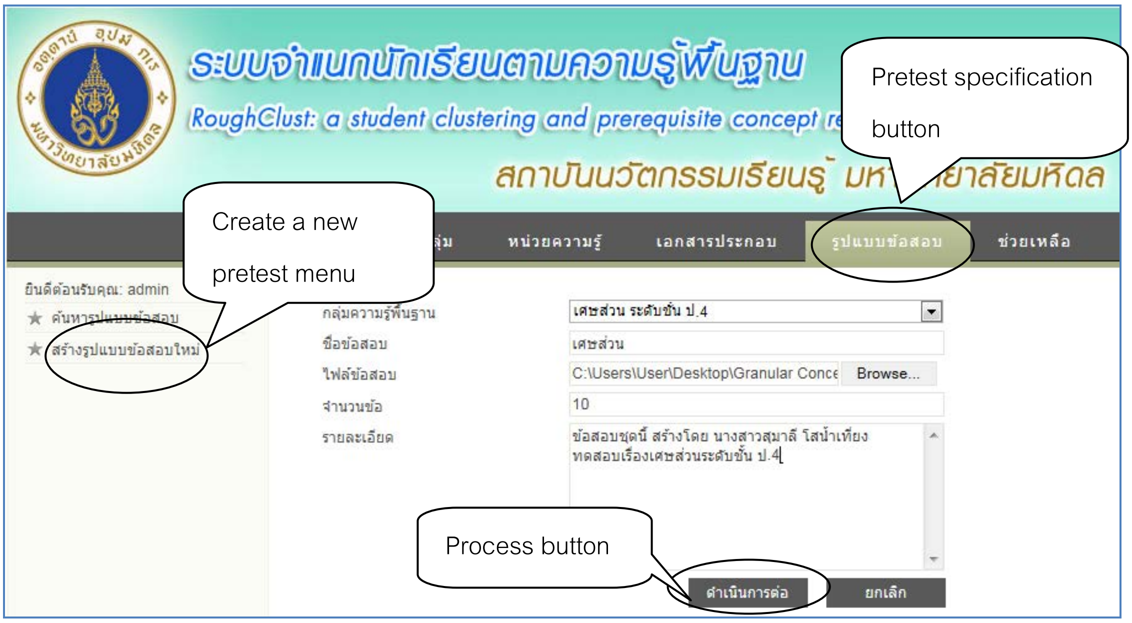


Figure 4.13 The screen to define the pretest.

Figure 4.14 The screen to define relationships amongst test items and concepts

4.3.1.4 Instruction material database management

Users can add and modify RoughClust's instruction material database by selecting the "instruction material" button on the top menu. Then, the user is presented by list of existing instruction materials as shown in Figure 4.15. To add a new instruction material, please select "add new material" on the left menu. The user is then presented by a screen to indicate (Figure 4.16) the new material's name, group of related concepts, instruction material types (handouts, exercises, and tests) and the printout file directory to be uploaded. Click "save" button to store the new material in RoughClust database for further use and share with other users of the system. In modifying an instruction material, please click on "modify" button in Figure 4.15, then the user will be presented by the screen in Figure 4.16 once again to modify its contents as desire.

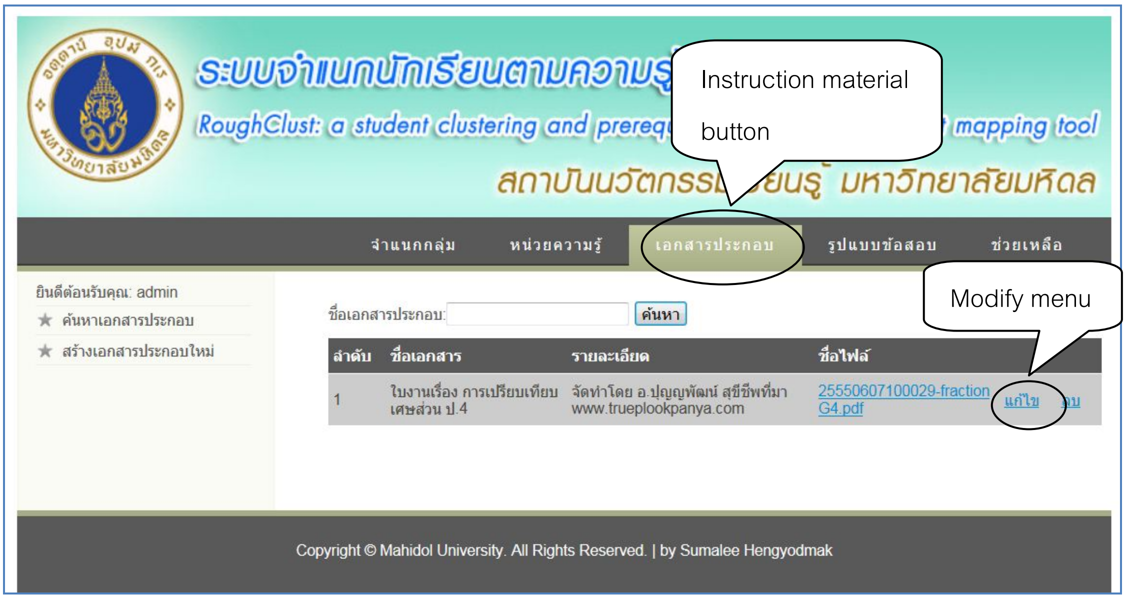


Figure 4.15 The screen to manage the instruction material database

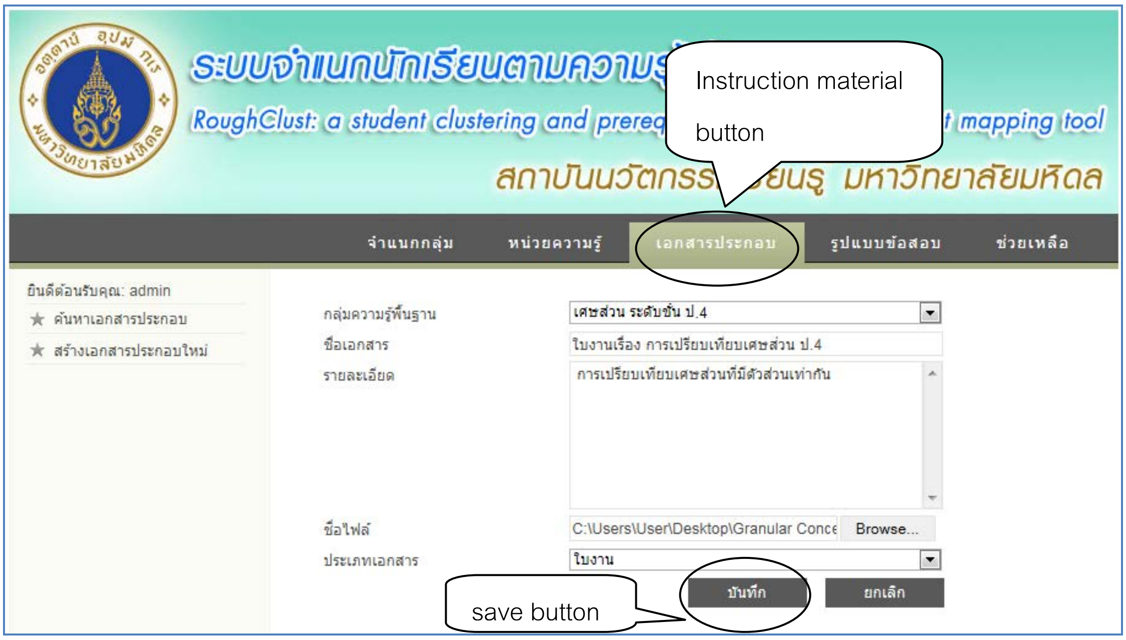


Figure 4.16 The screen to add content to the instruction material database

This chapter summarizes the design and implementation of RoughClust, its user interface, how to interpret the output, and how to extend RoughClust's knowledge base. The next chapter presents the evaluation and experimental results of the GCH clustering approach and RoughClust in respect of student clustering, student learning improvement and the user satisfaction.

CHAPTER V
EVALUATION OF GRANULAR CONCEPT HIERARCHY CLUSTERING AND ROUGHCLUST

This chapter presents evaluation of our GCH clustering approach and its education application, the RoughClust. In Section 5.1, we present evaluations of the cluster obtained from our GCH clustering approach using an artificial data set from the UCI Machine Learning Repository (Frank & Asuncion, 2010) which is hierarchical structured in nature. Next, we evaluate quality of clusters obtained from our algorithm for learning higher order rules. Then in Section 5.2, we discuss the evaluation of RoughClust specifically as a tool for teachers to acquire student groups and their prior knowledge regarding a teaching domain. We also present an experiment on student learning improvement after applying the “within class grouping” from RoughClust results in this section. Finally, an evaluation of user satisfaction is presented in Section 5.3.

5.1 Evaluation of rough set-based granular concept hierarchy clustering

5.1.1 Evaluation of clusters

To evaluate the quality of clusters obtained from a clustering algorithms, two dimension of measurements are commonly used which are an internal evaluation and external evaluation. The internal evaluation is to measure the similarity of objects within clusters. Therefore, internal measurement prefers algorithms that produce high similarity within a cluster and low similarity between clusters. Examples of internal measurement are Davies–Bouldin index (Davies & Bouldin, 1979) and Dunn index (Dunn, 1973). These internal evaluations are used to cluster validity for measuring goodness of a clustering relative to others created by other clustering algorithms, or by the same algorithms using different parameter values. These measurements are the calculations of distance matrices; however, the degree of inclusion based on accuracy of clustering is not considered. As a result, in our study, the accuracy of rough approximation as defined in Chapter 3 (page 34, Definition 7) is used to determine the internal evaluation of a cluster.

In addition to the internal evaluation, it is required for a clustering algorithm to measure how accuracy the clustering is to the predetermined classes. There are various methods to cluster external evaluation such as *F*-measure, the Jaccard index (Jaccard, 1912), Rand measure (Rand, 1971). In rough set paradigm, semantic cohesiveness measurement of a cluster or information granule can be defined as rough inclusion functions. Pawlak & Skowron (2007) presented the standard rough inclusion (SRI) function $v: P(U) \times P(U) \rightarrow [0,1]$ to define the degree of inclusion of X in Y where $X, Y \subseteq U^2$. The SRI function is defined by:

$$V_{SRI} = \begin{cases} \frac{|X \cap Y|}{|X|} & if\ X \neq \emptyset \\ 1 & if\ X = \emptyset. \end{cases} \tag{5.1}$$

Some illustrative example is given in Table 5.1. In case 1, set *X* belongs to set *Y* by the V_{SRI} =0. It indicates that *X* does not belong to *Y*. For case 2, V_{SRI} =0.25 which indicates that there is a probability of *X* belongs to *Y* = 0.25. Therefore, the smaller V_{SRI}, the more degree of distinct clusters is reflected.

Skowron, Stepaniuk & Swiniarski (2011) suggested that more work especially on inclusion of granule with complex structure should be more studied, in particular for granular neighborhoods. In evaluation of GCH clustering, we use the SRI function to measure the semantic cohesiveness of a single cluster at each level of the hierarchy towards the external evaluation.

Table 5.1 Illustration of standard rough inclusion function

Case	*X*	*Y*	V_{SRI}
1	{x1, x3, x7, x8}	{x2, x4, x5, x6, x9}	0
2	{x1, x3, x7, x8}	{x1, x2, x4, x5, x6, x9}	0.25
3	{x1, x3, x7, x8}	{x1, x2, x3, x7, x8}	1

Besides the internal and external evaluation, the structure of the information granule must be also evaluated towards its structure and applicability by the users. For hierarchical clustering, heuristics are required to determine and evaluate the hierarchical clusters (Jonyer, Cook & Holder, 2001).

5.1.2 Heuristic evaluation of hierarchical cluster quality

In 2001, Jonyer, Cook and Holder studied a graph-based hierarchical conceptual clustering and their clustering system SUBDUE was implemented. From their experience in SUBDUE, the authors suggested the "properties of a good clustering" to apply with heuristics evaluation of the cluster hierarchy as follows.

Properties of a good clustering:

1) Small number of clusters
    - Large coverage provides good generality.
2) Big cluster descriptions
    - More features provide more inferential power.
3) Minimal or no overlap between clusters
    - More distinct clusters provide better defined concepts.

We evaluate and discuss the hierarchical structure obtained from the GCH clustering approach based on these heuristics in Section 5.1.3.3.

5.1.3 Experiment and results

We divided the experiment to evaluate our clustering approach into two aspects. Firstly, we validated the hierarchical clustering using the animal data set. Not only the internal and external cluster quality but also the physical properties of the cluster hierarchy were evaluated. Secondly, we measured quality of clusters obtained from our algorithm for learning higher order rules. In addition, the importance of higher order rule is clarified.

5.1.3.1 Cluster internal evaluation by accuracy of rough approximation

The structured data set used in the experiment is the animal data set from the UCI machine learning repository (Frank & Asuncion, 2010) with the label Zoo. See Appendix C for this data set and its description. This data set contains 101 objects, 17 condition attributes and one decision attribute. The condition attributes include 16 Boolean valued attributes and a numerical attribute. The decision attribute contains 7 classes of animal type. There is no missing value in this dataset.

Part of the animal taxonomy obtained from GCH clustering algorithms is shown in Figure 5.1. The indentation represents level of granularity horizontally. A cluster is named beginning by the letter G, followed by an underscore "_" to remarks level of granularity and a number representing the cluster order. After the cluster' name, there is a set of members (animals) that belong to this cluster. For instances, the animal that belongs to the cluster G_3_2_1 are sole, seahorse, and haddock.

In our experiment, the animal data set clustering by the GCH approach gives the accuracy of rough approximation as shown in Figure 5.2. Please note that users can search for the maximum accuracy of rough approximation to obtain the best accuracy in prediction the decision class. If the accuracy of rough approximation reaches 1, the cluster's properties can be used to generate rules for classification task with 100% accuracy as well. For example, the granule "G_3_2 = TARGET(G_3_2) = 4; ACCURACY (G_3_2) = 1.0;" from Figure 5.2 indicates that animal in this clusters are classified into the same target concept 4. Moreover, its accuracy of rough approximation is 1.0. As a result, the GCH clustering approach presents the best accuracy on clustering towards the decision class as the user desire.

G_3 = {bass, catfish, haddock, sole, tuna, slowworm, carp, toad, herring, frog1, seahorse, tuatara, pike, frog2, dogfish, piranha, seasnake, chub, tortoise, newt, pitviper, stingray};
G_3_1 = {seasnake};
G_3_2 = {dogfish, tuna, catfish, piranha, herring, pike, sole, bass, seahorse, haddock, stingray, carp, chub};
G_3_2_1 = {sole, seahorse, haddock};
G_3_2_2 = {carp};
G_3_2_3 = {piranha, tuna, catfish, pike, dogfish, bass, herring, chub};
G_3_2_3_1 = {tuna, pike, dogfish};
G_3_2_3_2 = {bass, herring, piranha, catfish, chub};
G_3_2_4 = {stingray};
G_3_3 = {frog2, toad, frog1};

Figure 5.1 A part of the animal taxonomy

G_3 = TARGET(G_3) = 4; ACCURACY(G_3) = 0.59090906;
G_3_1 = TARGET(G_3_1) = 3; ACCURACY(G_3_1) = 1.0;
G_3_2 = TARGET(G_3_2) = 4; ACCURACY (G_3_2) = 1.0;
G_3_3 = TARGET(G_3_2) = 5; ACCURACY (G_3_2) = 1.0;

Figure 5.2 The rough accuracy approximations of some granular clusters from the animal taxonomy

We compare our approach with three algorithms implemented in the Weka data mining tool (Hall, Frank, Holmes, Pfahringer, Reutemann, & Witten, 2009) which are COBWEB (Fisher, 1987), EM (Dempster, Laird, & Rubin, 1977), and HierarchicalClusterer. Each algorithm clusters the same data set, the animal data set. The *classes to clusters evaluation* option provided in Weka is used to evaluate the accuracy of the clustering algorithm. The classes to clusters evaluation result for each algorithm is presented in Table 5.2. From the table, the EM algorithm produces the highest accuracy clustering results while COBWEB generates the highest errors.

The COBWEB, EM, and HierarchicalClusterer clustering algorithm did not offer fair results in classification task for this data set. One of the causes to this problem is that these algorithms take into account only the distribution of attributes' values and distance between clusters. However, the attribute dependency and attribute selection (using reducts and core attributes) are not considered in their clustering.

Table 5.2 Classes to clusters evaluation of hierarchical clustering algorithms implemented in Weka

Clustering algorithm	Number of incorrectly clustered instances	Incorrect clustering ratio	Accuracy
COBWEB	60	59.41 %	40.59%
EM	34	33.66 %	66.34%
HierarchicalClusterer	59	58.42 %	41.58%

5.1.3.2 Cluster external evaluation by standard rough inclusion function

From the animal data set clustering using the GCH algorithms, the V_{SRI} for clusters in the same level of granularity (or siblings) is 0 since there is no overlap between clusters allowed by the algorithm. However, the sibling clusters share their common features in their parents granular description language. Therefore, the V_{SRI} of a cluster and its parent granule is 1.

From a part of the animal taxonomy in Figure 5.1,

G3_1={seasnake},

G3_2={ dogfish, tuna, catfish, piranha, herring, pike, sole, bass, seahorse, haddock, stingray, carp, chub}, and

G3_2_1 = {sole, seahorse, haddock}.

Therefore, V_{SRI} (G3_1, G3_2) = 0 and V_{SRI} (G3_2_1, G3_2) = 1. The V_{SRI} (G3_1, G3_2) = 0 means that, at this level of granularity, the animal in G3_1 is totally distinct from G3_2. V_{SRI} (G3_2_1, G3_2) = 1 represents the inclusion degree in which G3_2_1 is subset of G3_2.

In addition to distinction cluster evaluation, the GCH clustering algorithm deals with uncertainty instances in clustering by allowing a cluster to have boundary region. For example in Figure 5.3, *newt* is included in the boundary region of cluster G_3_4 which means that it is uncertain to classify *newt* into the target concept = 3.

G_3_4 = {newt, pitviper, tortoise, slowworm, tuatara};
DES(G_3_4) = {eggs(1), breathes(1), domestic(0), tail(1), fins(0)};
TARGET(G_3_4) = 3;
ACCURACY (G_3_4) = 0.8;
BOUNDARY(G_3_4) = {newt};

Figure 5.3 A cluster description obtained from the GCH clustering system

In summary, all granular clusters produced by the GCH clustering algorithm presents good quality of cluster external evaluation by the SRI function since there is no overlap clusters. Moreover, the GCH is capable of defining the uncertain instances of a cluster using the rough set boundary region.

G1: {squirrel, fruitbat, vampire, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, dolphin, porpoise, wallaby, sealion, platypus, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl, calf, goat, pony, reindeer, pussycat}
G11: {fruitbat, vampire}
G12: {platypus}
G13: {squirrel, hare, vole, mole, opossum, cavy, hamster seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer elephant ,giraffe, oryx , boar ,cheetah ,leopard ,lion ,lynx ,mongoose, polecat, puma ,raccoon, wolf, mink, girl, dolphin, porpoise, calf, goat, pony, reindeer}
G131: {dolphin, porpoise}
G132: {squirrel, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl calf, goat, pony, reindeer}
G1321: {squirrel, gorilla, wallaby, girl, hare, vole, cavy, hamster, antelope, buffalo, deer, elephant, giraffe, oryx, calf, goat, pony, reindeer, mole, opossum, aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose polecat, puma, raccoon, wolf, pussycat}
G13211: {hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx, mole, opossum aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
G132111: {mole, opossum, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
G132112: {aardvark, bear}
G132113:{hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx}
G13212: {squirrel, gorilla, wallaby}
G13213: {girl}
G13214: {cavy, hamster, calf, goat, pony, reindeer, pussycat}
G1321411: {cavy}
G1321412: {hamster calf goat pony reindeer}
G1321413: {pussycat}
G1322: {seal, sealion}
G1323: {mink}
G2: {chicken, dove, duck, lark, parakeet, pheasant, sparrow, wren, kiwi, crow, gull, hawk, skimmer, skua, ostrich, flamingo, swan, penguin, rhea, vulture}
G21: {lark, pheasant, sparrow, wren, duck, kiwi, crow, hawk, gull, skimmer, skua}
G211: {kiwi}
G212: {lark, pheasant, sparrow, wren, duck, crow, hawk, gull, skimmer, skua}
G2121: {lark, pheasant, sparrow, wren, crow, hawk}
G21211: {lark, pheasant, sparrow, wren}
G21212: {crow, hawk}
G2122: {duck, gull, skimmer, skua}
G21221: {duck}
G21222: {gull, skimmer, skua}
G22: {chicken, dove, parakeet}
G23: {ostrich, flamingo, swan, rhea, vulture, penguin}
G231: {ostrich, flamingo, rhea, vulture}
G2311: {ostrich, rhea}
G23111: {ostrich}
G23112: {rhea}
G2312: {flamingo, vulture}
G23121: {flamingo}
G23122: {vulture}
G232: {swan penguin}
G2321: {swan}
G2322: {penguin}
G3: {pitviper, seasnake, slowworm, tortoise, tuatara, carp, haddock, seahorse, sole, bass, catfish, chub, dogfish, herring, pike, piranha, stingray, tuna, frog, frog, newt, toad}
G31: {bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna}
G311: {carp, haddock, seahorse, sole}
G312: {bass, catfish, chub, herring, piranha, dogfish, pike, tuna}
G3121: {bass, catfish, chub, herring, piranha}
G3122: {dogfish, pike, tuna}
G313: {stingray}
G32: {pitviper, slowworm, tortoise, tuatara}
G33: {seasnake}
G34: {frog, frog, newt, toad}
G341: {frog, newt}
G3411: {frog}
G3412: {newt}
G342: {frog}
G343: {toad}
G4: {flea, termite, gnat, honeybee, housefly, ladybird, moth, wasp, clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish}
G41: {clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish, flea, termite}
G411: {seawasp, starfish, flea, termite, slug, worm, crab, crayfish, lobster, clam}
G4111: {flea, termite, slug, worm}
G41111: {flea, termite}
G41112: {slug worm}
G4112: {seawasp}
G4113: {starfish, crab, crayfish, lobster, clam}
G41131: {clam}
G41132: {starfish, crab, crayfish, lobster}
G411321: {crab}
G411322: {starfish}
G411321: {crayfish, lobster}
G412: {scorpion}
G413: {octopus}
G42: {gnat, ladybird}
G421: {gnat}
G422: {ladybirdl}
G43: {housefly, moth, wasp}
G431: {housefly, moth}
G432: {wasp}
G44: {honeybee}

Figure 5.4 The animal taxonomy obtained from the rough set-based GCH clustering approach.

5.1.3.3 Hierarchy cluster structure evaluation

We constructed a GCH for the animal data set as shown in Figure 5.4 (Sonamthiang, Cercone & Naruedomkul, 2007). Our hierarchy meets properties of good clustering as suggested by Jonyer, Cook & Holder (2001) as described as follows.

1) Small number of clusters
   - Large coverage → good generality. This is considered as a good property of a clustering approach because less number of clusters is simpler where the clusters remain large coverage and distinct clusters. Number of clusters obtained from our approach may vary at different level of granularity. As a result the approach provides users the number of clusters as the user desires.
2) Big cluster descriptions
   - More features → more inferential power. Our system provides another output file dedicated for cluster description. The cluster description is rich information since it contains information about the cluster’s members (extension), common properties (attributes and the attribute values), the target concept (class), lower approximation, rough accuracy, and the members of the boundary region. An example of a cluster description is illustrated in Figure 5.4.
3) Minimal or no overlap between clusters
   - More distinct clusters → better defined concepts. The cluster obtained from our approach is clearly defined which is no overlap between two clusters is allowed. Moreover, the uncertainty of being included in a cluster is defined by rough set boundary region.

As a result, the hierarchy structure produced by our GCH clustering algorithm offers all good properties as a hierarchical clustering approach should accommodate.

5.1.3.4 Learning higher order rules from the granular concept hierarchy

To prove the usefulness of a GCH clustering, we adopted high order decision rules learning from the GCH structure obtained from the clustering. Definition of higher order rules are introduced by (Yao, Zhou & Chen, 2007). A higher order rule expresses connections of different objects based on their attribute values. An example of a higher order rule is: "If object *x* is related to object *y* with respect to an attribute set *a*, then *x* is related to *y* to another attribute set *b*." The authors recommended mining of higher order rules from a transformed decision table which an entity is a pair of objects from the original table. However, transforming the *n* objects table generates $n!/((n-2)!*2!)$ pairs of objects. We present an alternative approach to extract higher order decision rules from a GCH which no transformation process is required.

From the animal taxonomy, there can be several groups of animals that hold the same attributes' values in a subset of condition attributes. For example, there are 6 groups of animals clustered by attribute set $\{C;E;J;K\}$ which are *Feathers*, *Milk*, *Backbone*, *Breathes* respectively. These attributes draw a concept of *mammal* when $C = 0$; $E = 1$; $J = 1$, and $K = 1$. The concept of bird is drawn when $C = 1$; $E = 0$; $J = 1$, and $K = 1$, the concept of *amphibian* is formed by $C = 0$;$E = 0$; $J = 1$, and $K = 1$. The *arthropod* (bug) concept is formed by $C = 0$; $E = 0$; $J = 0$, and $K = 1$. The concept of fish is formed by $C = 0$; $E = 0$; $J = 1$, and $K = 0$. The concept of being *crustacean* is formed when $C = 0$; $E = 0$; $J = 0$, and $K = 0$. As a results, the common attributes can be used to obtained higher order rules.

Once the hierarchy is obtained, a depth first tree search is performed to find the maximum level of accuracy of each branch. The high order rules are obtained from the conjunctive connection of granular concepts' intensions along the visited branches of the hierarchy, for example, the animal taxonomy. The concept descriptions of the animal groups are used to generate the high order rules.

The extracted higher order decision rules for the animal data set given in Table 5.3 is from our previous research in Granular Concept Maps (Sonamthiang, Naruedomkul & Cercone, 2012). Number of conjunction shows the level of hierarchy, starting from level 0 at the root. The high order rules are applied to the total of 5,050 pairs of animals and there are 1,177 pairs of animals that are belong to the same class. In this experiment, we apply a measurement of rules' *accuracy* and *coverage* to evaluate the higher order rules quality. The rules' *accuracy* and *coverage* are defined as (Tsumoto, 2002):

$$accuracy(premise \rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(premise)|}, \quad (5.2)$$

$$coverage(premise \rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(conclusion)|} \quad (5.3)$$

where $\phi(g)$ is the granule's extension and $|X|$ denotes the cardinality of the set X. The results of the rules' accuracy and coverage are shown in the second and the third column of Table 5.3, respectively.

We discuss the interestingness of the high order decision rules as follows. High order rule is the type of knowledge in more abstract level. This knowledge should be firstly applied to solve a problem. Naturally, given two animals, one can differentiate them by the concepts, not by the detailed of each attribute value if not necessary. The high order rules provide the concepts upon the domain which the rules can be applied for only some groups. The rules obtained from our approach have much higher accuracy degree than the coverage degree. This is because of the tree traversal searches for the maximum accurate level of each branch, where their children do not have smaller boundary regions than the parents. Once the target granules are found, the granules' language can be used to express the connections between objects in the same granule directly. The connections are multi-dimension which reflect the relationships between attributes in the attribute subset (e.g. dependencies) and also the relationships between the condition attribute subset and the decision attribute. On the other hand, if one prefers the rule with higher coverage degree, the bread first search for the coarser granules can be achieved.

Table 5.3 Accuracy and coverage measurement of high order rules generated from the animal hierarchy

Rules	Accuracy	Coverage
Rule 1: IF animals have the same values in condition attribute {*C*; *E*; *J*; *K*} THEN they are in the same class of {*R*}	0.956	0.979
Rule 2: IF animals have the same values in condition attributes {*C*; *E*; *J*; *K*} and animals have the same values in condition attributes {*B*; *D*; *F*; *H*; *I*; *M*; *O*; *P*} THEN they are in the same class of {*R*}	0.972	0.209
Rule 3: IF animals have the same values in condition attributes {*C*; *E*; *J*; } and animals have the same values in condition attributes {*D*; *L*; *M*; *P*} and animals have the same values in condition attributes {*B*; *F*; *G*; *I*; *N*; *O*} THEN they are in the same class of {*R*}	1.000	0.266
Rule 4: IF animals have the same values in condition attributes {*C*; *E*; *J*; *K*} and animals have the same values in condition attributes { *D*; *P*} and animals have the same values in condition attributes {*G*; *I*; *M*; *Q*} THEN they are in the same class of {*R*}	0.961	0.293
Rule 5: IF animals have the same values in condition attributes {*C*; *E*; *J*; *K*} and animals have the same values in condition attributes {*F*} and animals have the same values in condition attributes {*B*; *D*; *M*; *O*} THEN they are in the same class of {*R*}	0.988	0.562
Rule 6: IF animals have the same values in condition attributes {*C*; *E*; *J*; *K*} and animals have the same values in condition attributes {*B*; *D*; *I*; *L*; *M*; *N*; *O*} THEN they are in the same class of {*R*}	0.998	0.534

5.2 Evaluation of RoughClust

This section presents student clustering and prerequisite concept requirement mapping features of RoughClust in Section 5.2.1 . Section 5.2.2 show an experiment and evaluation of student learning improvement after applying group-based instruction suggested by RoughClust. RoughClust was evaluated using two real data sets. The data sets were collected from Fraction prior knowledge test for primary school level.

5.2.1 Student clustering results and analysis

The data set is gathered from a prior knowledge testing using the Fraction test in Appendix C. The test is composed of 16 items and each item is related to fraction concepts. There are 88 students in this data set. The student were attending grade 4, 5, and 6 in academic year of 2010. The data set was preprocessed and prepared as an information system is given in Appendix B.1.

We experiment the optimal parameters for clustering student data set 1 by parameter setting. In the first experiment, the parameter of maximum group is variable as 20, 10, 5, and 2 students, and the accuracy threshold is set to 0.80 as an acceptable accuracy approximation. The experimental result is shown in Table 5.4. For the second experiment, the maximum group size parameter is controlled where the accuracy threshold is variable as 1.00, 0.80, 0.50, 0.20, and 0. The result from the second experiment is given in Table 5.5. Teachers can experiment with these parameters to consider which parameter gives the most suitable to apply to group-based instruction.

Table 5.4 Selected student clusters with variant group sizes.

parameters		number of groups	average group sizes
maximum group size	accuracy threshold		
20	0.8	34	2.59
10	0.8	44	2.00
5	0.8	50	1.76
2	0.8	59	1.49

Table 5.5 Selected student clusters with accuracy threshold.

parameters		number of groups	average group sizes
maximum group size	accuracy threshold		
10	1.00	46	1.91
10	0.80	44	2.00
10	0.50	21	4.19
10	0.20	21	4.19
10	0.00	21	4.19

5.2.2 Student learning improvement

We studied the usefulness and practice of RoughClust by an experiment on Fraction domain for primary level students. There were 26 students from grade 6 participated in this study. The student prior knowledge towards Fraction is tested using a fraction pretest in Appendix C. All students' test answers were used to cluster students into groups. This data set is given in Appendix B.1.

Clustering this group of student using RoughClust with the maximum group size =10 and the accuracy threshold = 0.5, there were 7 groups of students obtained. The sets of common errors and prerequisite concept requirement for each group are listed in Table 5.6. For example, in Group 1, there are six students in this group which are the student ID 628, 627, 610, 626, and 629. The students in this group share incorrect answering towards the concept of:

- reading fraction,
- writing fraction,
- conversion of mixed numbers to improper fractions,
- conversion of improper fractions to mixed number,
- equivalent fractions,
- comparing and arranging sequence of fractions.

Moreover, RoughClust suggests that the students in this group may require the following prerequisite concepts:

- meaning of fractions,
- meaning of improper fractions and mixed numbers,
- comparing fractions with the same, division,
- comparing and ordering whole numbers,
- multiplication,
- arranging sequence of fractions with the same denominator,
- comparing fractions with the same denominator, and
- reduced fraction.

Table 5.6 Prerequisite concepts requirement for each group learning remedy

Group 1	
Student ID	{628, 627, 610, 626, 629}
Common error concepts	reading fraction, writing fraction, conversion of mixed numbers to improper fractions, conversion of improper fractions to mixed number, equivalent fractions, comparing and arranging sequence of fractions
Prerequisite concept required	meaning of fractions, meaning of improper fractions and mixed numbers, Comparing fractions with the same, division, comparing and ordering whole numbers, multiplication, arranging sequence of fractions with the same denominator, comparing fractions with the same denominator, reduced fraction
Group 2	
Student ID	{623, 612, 624, 606}
Common error concepts	reading fraction, writing fraction, equivalent fractions, Comparing and arranging sequence of fractions
Prerequisite concept required	meaning of fractions, comparing fractions with the same denominator, division, comparing and ordering whole numbers, multiplication, arranging sequence of fractions with the same denominator, comparing fractions with the same denominator, reduced fraction
Group 3	
Student ID	{611, 630}
Common error concepts	reading fraction, writing fraction, adding and subtracting fractions with the same denominator, conversion of mixed numbers to improper fractions, conversion of improper fractions to mixed number, equivalent fractions, Comparing and arranging sequence of fractions

Table 5.6 Prerequisite concepts requirement for each group learning remedy (Cont.)

Prerequisite concept required	meaning of fractions, reduced fraction, meaning of improper fractions and mixed numbers, Comparing fractions with the same , division, comparing and ordering whole numbers, multiplication, arranging sequence of fractions with the same denominator, comparing fractions with the same denominator
Group 4	
Student ID	{616, 607}
Common error concepts	reading fraction, writing fraction
Prerequisite concept required	meaning of fractions
Group 5	
Student ID	{619}
Common error concepts	reading fraction, writing fraction, adding and subtracting fractions with the same denominator, conversion of mixed numbers to improper fractions, conversion of improper fractions to mixed number, equivalent fractions, Comparing and arranging sequence of fractions
Prerequisite concept required	meaning of fractions, reduced fraction, meaning of improper fractions and mixed numbers, Comparing fractions with the same , division, comparing and ordering whole numbers, multiplication, arranging sequence of fractions with the same denominator, comparing fractions with the same denominator
Group 6	
Student ID	{608, 614}
Common error concepts	reading fraction, writing fraction, adding and subtracting fractions with the same denominator, conversion of mixed numbers to improper fractions, equivalent fractions, comparing and arranging sequence of fractions
Prerequisite concept required	meaning of fractions, reduced fraction, meaning of improper fractions and mixed numbers, comparing fractions with the same , division, comparing and ordering whole numbers, multiplication, arranging sequence of fractions with the same denominator, comparing fractions with the same denominator
Group 7	
Student ID	{618, 620, 615, 625, 601, 605, 609, 622, 602, 603}
Common error concepts	reading fraction, writing fraction, conversion of mixed numbers to improper fractions, comparing and arranging sequence of fractions
Prerequisite concept required	meaning of fractions, meaning of improper fractions and mixed numbers, Comparing and arranging sequence of fractions with the same denominator, comparing fractions with the same denominator, reduced fraction

To evaluate efficiency of applying results suggested from RoughClust, we asked the two mathematics teachers to plan group-based instruction using the group information and the instructional material provided for each group. The instruction was scoped to the concept "comparing and arranging sequence of fractions." The teachers arranged within-class grouping instruction in two regular mathematics hours. Then, the students were given the posttest in the next day. The results found that 35.15% of students' scores were improved as shown in Table 5.7.

Table 5.7 Student learning improvement

	Number of students	Average pretest score (total of 16 scores)	Average posttest score (total of 16 scores)	Score improvement	Percentage improvement
Group 1	5	7.82	12.60	4.78	61.13
Group 2	4	12.25	13.20	0.95	7.76
Group 3	2	6.67	10.08	3.41	51.12
Group 4	2	4.12	7.00	2.88	69.90
Group 5	1	7.00	10.00	3.00	42.86
Group 6	2	11.5	13.00	1.50	13.04
Group 7	10	8.56	12.40	3.84	44.86
average		8.27	11.18	2.91	35.15

5.3 User satisfaction evaluation

We evaluated teachers' satisfaction using the questionnaire. The questionnaire for RoughClust evaluation is adopted from Devis (1989)'s study, TAM (Technology Acceptance Model) which considers *perceived usefulness* and *perceived ease of use* in computer acceptance behaviors of users. The questionnaires were administered to three mathematics teachers who interacted with RoughClust. Each question has five scales; strongly agree, agree, neither, disagree, and strongly disagree. The questionnaire is presented in Thai version in Appendix C). Some open questions are also attached to this questionnaire. The result of evaluation of the teachers' satisfaction is illustrated in Table 5.8. The average scale given from the three teachers is shown in the last row which is 3.55. This scale shows that teachers have positive perception to RoughClust.

Table 5.8 Questions and results of evaluation of the teachers' satisfaction (translated from Thai to English)

Measurement	**Average Scale**
Usefulness	
1. Using RoughClust would help me in grouping student based on their prior knowledge	3
2. RoughClust provides acceptable accurate and appropriate grouping for the student prior knowledge	3
3. RoughClust suggests useful group's prerequisite concept requirement.	3
4. Instructional materials suggested by RoughClust is useful and suitable for my students' requirement.	5
5. Using RoughClust is helpful in my instruction planning.	4
6. Using RoughClust to instruction planning can help my student learning improvement.	3.5
7. Using RoughClust to Implement group-based instruction can help students attending the class more comfortable.	4
Ease of use	
8. My interaction with RoughClust would be clear and understandable	3
9. I would find RoughClust easy to use.	2.5
10 I will continue using RoughClust.	4
11. I will tell the other teacher to use RoughClust.	4
Average scale	**3.55**

From the open questions, the teachers believe that RoughClust tool is a useful for managerial prerequisite concept relationships and test in order to use the system for other groups of students. They are not only able to reuse of the system but also to extend the domain knowledge of the prerequisite concept relationships and the database of instruction materials. The database of RoughClust also helps them systematically store and share their instruction material with other teachers. The teachers were satisfied to be able to query and reuse the instructional material whenever they want to. They can produce handouts, exercises, or exams appropriate for student prior knowledge requirement much more easily than ever. According to the evaluation results, teachers were satisfied with some noticeable learning improvement of their students in learning fractions when the students were accommodated with the proper prior knowledge instruction.

There are also some valuable comments from the teachers as shown in Figure 5.5. We will have RoughClust improved according to the teacher' suggestions in the near future.

Comment 1: Defining concepts and the concepts relationships in RoughClust is not easy. It takes some times to understand and operate the domain knowledge databases. Moreover, I am not sure the concepts and the relationships added are correct.

Comment 2: Teaching students in groups within my classroom is more fun. Students enjoy working in groups. However, it takes time for me to plan and produce different instructional materials for each group. It would be better if the students could use RoughClust to see their potential and weakness in prior knowledge and get exercises whenever they like to.

Comment 3: I found that it is easy to store and find the instruction material in RoughClust. However, some features of the tool are difficult to use such as defining the concepts. There should be a standard model for concepts with their relationships to avoid incorrect or inconsistent knowledge.

Figure 5.5 Examples of users' comments

Evaluation of the GCH clustering approach and RoughClust are presented in this chapter. GCH generates more informative hierarchical clusters and more accuracy in classification than some hierarchical clustering approaches. We evaluate RoughClust on student prior knowledge testing results. RoughClust offers the teachers with flexible parameters to student cluster selection. The parameter variables were experimented to find the best student clusters. In the next chapter, the thesis is concluded and discussion on further research based on the evaluation in this chapter is presented.

CHAPTER VI
CONCLUSIONS

6.1 Conclusions and discussion

In this thesis, we present not only GCH clustering approach but also its education application called RoughClust, the student clustering and prerequisite concept requirement mapping tool. GCH clustering approach employs indiscernibility relations to multilevel partitioning and rough sets in defining clusters. Moreover, correlation and dependency between attributes are used for attribute selection partitioning. We proposed the designed algorithms to recursively construct a GCH and to select the attribute subset for multilevel partitioning. Core attributes are essential for forming more specific concepts since they contain specific characteristics of an object. In GCH construction, the core attributes are preserved until the last granulation.

GCH clustering approach was validated using the structured animal data set and generating high order rules from the animal taxonomy obtained from the GCH clustering approach. Our approach can discover more understandable granular concept hierarchy for structured data especially in data sets that the correlation and dependency between attributes exist. Since GCH clustering approach takes into account the decision class in clustering and applies reduct and core attributes to preserve specific information in granulation, therefore it gives more accurate clustering results in supervised learning. The comparative results showed that GCH approach provides more understandable abstract conception of a cluster hierarchy and more descriptive cluster information than COBWEB, EM, and HierarchicalClusterer.

Since GCH provides multilevel granular knowledge ranging from the coarsest level at the root and the most specific level at the leaves, GCH structure provides system of granular knowledge mapping through a tree traversal. Searching for a granular concept in GCH can be performed through several techniques including the depth first search and breadth first search. In order to map an appropriate granular knowledge, the problem solver must identify satisfaction criterions. One of satisfaction criterion is that the granular knowledge is evaluated by sufficient knowledge for solving a particular problem. If the problem is to find the decision rules to predict the unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them has smaller boundary regions. If the problem is to predict the missing values of the condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. A satisfaction criterion can be defined by setting the precision tolerance of applying the granular knowledge. This criterion reduces expensive computation of total precision.

To benefits from GCH clustering, we developed RoughClust: a student clustering and prerequisite knowledge requirement mapping tool. RoughClust aims at helping teachers to acquire groups of students who has similar learning difficulty and to understand the groups' prerequisite concept requirement regarding a teaching domain. Teachers can use RoughClust to obtain the granular groups of students, the groups' prerequisite concepts requirement, and the instructional material for their classroom to best serves the students' prior knowledge differences. RoughClust was designed to perform three main modules which are Data Clustering, Cluster Selection, and Prerequisite Concept Requirement Mapping. In Data Clustering module, we employed GCH clustering and domain dependency of a test items in multilevel attribute selection for partitioning. Teachers can define their clusters through the Cluster Selection module by indicating a maximum cluster sizes and rough accuracy threshold. Prerequisite Concept Requirement Mapping is then evoked to search for each group's prerequisite concepts that should be taught to the students before teaching the particular concepts. In addition, RoughClust also offers teachers a collection of instruction materials including handouts, exercises, and tests for each student group. In summary, RoughClust is capable of:

1) Grouping students who hold common knowledge mistakes.
2) Mapping to the groups' granular prerequisite concept requirement by using the group characteristics together with the prerequisite concept relations of the domain.
3) Suggestion of the appropriate and handy instruction materials for the teachers to use in group treatment instruction planning. The

instructional material provided are handouts, exercises, and a set of test items related to each granular learning concept.

4) Articulating mechanism for the granular prerequisite knowledge level as deep as necessary.

We studied the usefulness and practice of RoughClust by an experiment on Fraction domain for primary level students. There were 26 students from grade 6 participated in this study. The students' prior knowledge towards Fraction was tested. All students' test answers were used to cluster students into groups. There were seven groups of students who have similar prior knowledge requirement. Then, we asked the mathematics teacher to arrange within-class grouping instruction and use the instructional material provided for each group instruction. The instruction was carried out for two hours and the students were given a posttest in the next day. The test result showed that the students' scores were significantly improved. Therefore, clustering students and treating each group based on their prior knowledge is worthwhile.

Teachers, students, and parents will benefit greatly from RoughClust. Teachers can quickly determine the students' prior knowledge and prepare the instruction plan since they can use group-based prerequisite concept requirement to construct a group-based curriculum and an instruction plan. In addition, teachers can use the provided materials, e.g., handouts, exercises, and tests in their classroom. Therefore, diversity of student learning ability is easier to accommodate with the group-based curriculum within classroom learning. Moreover, more comfortable classroom learning environment is promoted since students complacently learn with their learning pace, performance, materials, and like-ability peers. Once the students' needs are correctly treated at the school, the supplementary tutoring in out-of-school is reduced. Parents' expense for their children tutoring will be lower or cut off. More importantly, students are able to spend more time for other activities rather than attending the tutoring classes after school or at the weekends.

Contribution of this thesis is concluded as follows:

- What's new? – Granular concept hierarchy clustering approach based on rough sets and granular prerequisite knowledge requirement mapping.

- What’s different? – Granular solution to uncover students’ common errors and groups of student who have similar concept learning difficulty.
- What’s better? – Our approach is better than existing testing diagnostic systems since it provides granular information in two point of views. The first point is granular group of students based on specific common errors. The second point is granular domain concept hierarchy of the prerequisite knowledge requirement.
- What’s significant? – Based on the group’s common errors and the domain’s prerequisite relation, we can provide instruction materials such as handouts, exercises, and tests that are suitable for each group and ready to use by the teachers.

6.2 Further research

GrC provides both perspectives and mechanisms to education research in student learning behaviour and knowledge organization. This thesis presents a good example of applying GrC approach in student clustering and prerequisite mapping. The results of this research will contribute to current attempts to data clustering in education; for example, the national standard test data and university admission test. The informative and accurate clustering methodology should provide insights for educators to manage learning resources for schools and classrooms more effectively and for teachers to perform instruction that serves individual differences. In addition, research in granules in learning behaviour based on (Sonamthiang, Cercone & Naruedomkul, 2007b) will be investigated to study the student learning behaviour patterns when interaction with e-learning systems and ITSs.

In this thesis, RoughClust was implemented and limited for Fraction domain for primary student level. For further research, RoughClust should provide mechanism for teachers to actively corporate with experts to manage and extend the domain prerequisite concept relations in the existing knowledgebase. This capability will allow teachers to more articulate the students’ prior knowledge requirement as desire and the results will be more reliable. Moreover, RoughClust should able to allow collaboration between teachers in expanding its domain knowledge and share instructional materials. Last but not least, intensive study on how to employ group-based instruction based on students' prior knowledge requirement is required.

REFERENCES

Aleven, V., & Koedinger, K. R. (2002). *An effective meta-cognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor*. Cognitive Science, 26(2), 147-179.

Anderson, J., R., Boyle, C. F., & Reiser, B. J. (1985) *Intelligent Tutoring Systems*. Science, 228, 456-462.

Ashlock, R. (2010). *Error Patterns in Computation: Using Error Patterns to Help Each Student Learn* (10th Edition), Allyn & Bacon, 288 pages.

Ausubel, D.P. (1968). *Educational Psychology: A Cognitive View*. New York: Holt, Rinehart & Winston.

Baffes, P. T., & Mooney, R. J. (1996). *A novel application of theory refinement to student modeling.* In Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland, OR, 403-408.

Bazan, J. G. & Szczuka, M. (2005). *The Rough Set Exploration System*. Transactions on Rough Sets III, Springer-Verlag, Berlin Heidelberg New York, 37–56.

Biswas, G.; Weinberg, J. B.; Fisher, Douglas H. (1998). *Iterate: A conceptual clustering algorithm for data mining*. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 28: 100–111.

Brusilovsky, P. (1996) "Methods and techniques of adaptive hypermedia". User Modeling and User Adapted Interaction, Volume 6, Issue 2-3, July 1996, 87-129.

Capuano, N., Marsella, M., & Salerno, S. (2000) *ABITS: an agent-based intelligent tutoring system for distance learning*. Proceedings of ITS'2000, Montreal.

Crowley, R.S., Medvedeva, O., Jukic, D. (2003) *SlideTutor – A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis.* Proceedings of the 11th International Conference on Artificial Intelligence in Education. Sydney, Australia, 2003. 157-164.

Carbonell J.R. (1970). *AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction*, IEEE Transactions on Man-Machine Systems, Vol.11, No.4, 190-202.

Christen, W. L., & Murphy, T. J. (1991*). Increasing comprehension by activating prior knowledge*. ERIC Digest # 61. Bloomington.

Clark, R. (2003). *Building expertise: Cognitive methods for training and performance improvement*. Washington, DC: International Society for Performance Improvement.

Clarke, T., Ayres, P. & Sweller, J. (2005) *The Impact of Sequencing and Prior Knowledge on Learning Mathematics Through Spreadsheet Applications,* ETR&D, Vol. 53, No.3, 15-24.

Conati, C., Gertner, A. S., VanLehn, K., & Druzdzel, M. J. (1997). *On-line student modeling for coached problem solving using Bayesian network*s. In Proceedings of UM-97, Sixth International Conference on User Modeling, Sardinia, Italy: Springer. 231–242.

Corbett, A. T., Anderson, J. R., & Patterson, E. G., (1990). *Student modeling and tutoring flexibility in the LISP Intelligent Tutoring System*. In C. Frasson and G. Gauthier (Eds.) Intelligent tutoring systems: At the crossroads of artificial intelligence and education. Norwood, NJ: Ablex. 83-106.

Corter, J. E. & Gluck, M. A. (1992), *Explaining basic categories: Feature predictability and information*, Psychological Bulletin 111 (2): 291–303.

Davies, D. L. & Bouldin, D. W. (1979). *A Cluster Separation Measure*. IEEE Transactions on Pattern Analysis and Machine Intelligence (2): 224.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), 39 (1), 1–38.

Dochy, F., Segers, M., & Buehl, M.M. (1999). *The relation between assessment practices and outcomes of studies: the case of research on prior knowledge*. Review of Educational Research, 69(2), 145-186.

Dunn, J. C. (1973). *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters*. Journal of Cybernetics 3 (3): 32–57.

Ericson, K. A., & Polson, P.G. (1988). *An experimental analysis of the mechanisms of a memory skill*. Journal of Experimental Psychology: Learning, Memory and Cognition, 14, 305-316.

Fisher, D. H. (1987). *Knowledge acquisition via incremental conceptual clustering*. Machine Learning 2 (2): 139–172.

Frank, A. & Asuncion, A. (2010). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Greer, J. E. & McCalla, G. I. (1994). *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, NATO ASI, Series F. Springer-Verlag.

Gluck, M. A. & Corter, J. E. (1985), *Information, uncertainty, and the utility of categories*, Program of the Seventh Annual Conference of the Cognitive Science Society, 283–287.

Grzymala-Busse, J. (2004). *Rough Set Approaches to Rule Induction from Incomplete Data*. Proceedings of the IPMU 2004, 923–930.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009). *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1, 10-18.

Harrison, A. G., Grayson, D. J., & Treagust, D. F. (1999). *Investigating a grade 11 student's evolving conceptions of heat and temperature*. Journal of Research in Science Teaching Volume 36, Issue 1, 55–87.

Hartigan, J. A. & Wong, M. A. (1979*). Algorithm AS 136: A K-Means Clustering Algorithm*, Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 28, No. 1, 100-108.

Hewson, M. G. & Hewson P W. (1983). *Effect of instruction using students' prior knowledge and conceptual change strategies on science learning*. Journal of Research in Science Teaching Vol.20, Issue 8, 731-743.

Hwang, G. J (2003). *A concept map model for developing intelligent tutoring systems*. Computers & Education, 40(3), 217-235.

Jaccard, P. (1912). *The distribution of flora in the alpine zone.* New Phytologist. 11, 37–50.

Jonyer, I., Cook, D.J. & Holder, L.B. (2001) *Graph-Based Hierarchical Conceptual Clustering*. Journal of Machine Learning Research 2 (2001) 19-43.

Kalyuga, S., Ayres P., Chandler ,P., & Sweller, J. (2003). *Expertise reversal effect*. Educational Psychologist, 38, 23-31.

Kavčič, A. (2004). *Fuzzy User Modeling for Adaptation in Educational Hypermedia*. IEEE Transactions on Systems, Man, and Cybernectics - Part C: Applications and Reviews 34(4), 439-449.

Kirn, N., Evens, M., Michael, J. A. & Rovick, A. A. (1989) *CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology*. In Maurer, H. (ed.), Computer-Assisted Learning: 2nd International Conference, ICCAL '89, pp. 254-266. Berlin: Springer-Verlag, 1989. Published as v. 360 of Lecture Notes in Computer Science.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. (1997). *Intelligent tutoring goes to school in the big city*. International Journal of Artificial Intelligence in Education, 8, 30-43.

Kulik, C.L. C., & Kulik, J. A. (1982). *Effects of ability grouping on secondary school students: A meta-analysis of evaluation findings*. American Educational Research Journal, 19(3), 414-428.

Kulik, J. A., & Kulik, C.L. C. (1989). *Effects of Ability Grouping on Student Achievement*, Equity and Excellence 23, 1-2: 22-30.

Lebowitz, M. (1987). *Experiments with incremental concept formation*. Machine Learning 2 (2): 103–138.

Lin, T.Y. (1997). *Granular Computing*, Announcement of the BISC Special Interest Group on Granular Computing.

Lin, Y.C., Lin, Y.T. & Huang, Y.M. (2011) *Development of a diagnostic system using a testing-based approach for strengthening student prior knowledge.* Computers & Education, Volume 57, Issue 2, September 2011, 1557–1570.

Lloyd, S. P. (1982). *Least squares quantization in PCM*. IEEE Transactions on Information Theory 28 (2): 129–137

Mayer, H. (2004). *Novice and Expert Teachers Conceptions of Learner Prior Knowledge*, Journal of Science Education, 88(6), 970-983.

Mitrovic A. (2003). *An Intelligent SQL Tutor on the Web.* International Journal of Artificial Intelligence in Education .Volume 13, Numbers 2-4 / 2003, 173 – 197.

Panjaburee, P., Hwang, G.J., Triampo, W. & Shih, B.Y. (2010) *A multi-expert approach for developing testing and diagnostic systems based on the concept-effect model,* Computers & Education, Volume 55, Issue 2, September 2010, 527-540.

Pawlak, Z. (1982). *Rough Sets.* International Journal of Computer and Information Sciences, Vol.11, No. 5.

Pawlak, Z. (1991). *Rough Sets. In Theoretical Aspects of Reasoning about Data*. Kluwer, Netherlands.

Pawlak, Z. (1998). *Granularity of knowledge, indiscernibility and rough sets*, Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 106-110.

Pawlak, Z. & Skowron, A. (2007). *Rough sets: Some extensions. Information Sciences,* Vol. 177, No. 1, 28–40.

Pedrycz, W., Skowron, A. & Kreinovich, V. (Eds.) (2008), *Handbook of Granular Computing*, New York: Wiley.

Pillay, H., Wilss, L., & Boulton-Lewis G. (1998). *Sequential Development of Algebra Knowledge: A Cognitive Analysis*, Mathematics Educational Research Journal, Vol. 10, No.2, 87-102.

Quinlan, J. R. (1986). *Induction of Decision Trees*. Machine Learning 1, 1, 81-106.

Rogers, K. (1998). *Using current research to make "good" decisions about grouping*. National Association of Secondary Schools Principals Bulletin, Feb 1998.

Rich, E. (1989), *Stereotypes and User Modeling*. In: A. Kobsa and W. Wahlster: 1989, User Models in Dialog Systems, Heidelberg: Springer Verlag.

Skowron, A. & Peters, J. F. (2008). *Rough-granular computing*, in: W. Pedrycz, A.Skowron, V. Kreinovich (eds.), Granular Computing, John Wiley & Sons, Ltd., Chichester, UK, 2008, 225-328.

Slavin, R. E. (1987). *Ability grouping: A best-evidence synthesis*. Review of Educational Research, 57, 293-336.

Johnson, W. L. & Soloway E. (1985). *PROUST: knowledge-based program understanding*. IEEE Transactions on Software Engineering Volume 11, Issue 3 (March 1985). 267 – 275.

Skowron, A., Stepaniuk, J. & Swiniarski, R. (2011). *Calculi of Approximation Spaces in Intelligent Systems*, in Granular Computing and Intelligent Systems, Intelligent Systems Reference Library, 2011, Volume 13, 35-55

Sleeman, D. & Brown, J. S., (1982) *Intelligent Tutoring Systems,* Science, Volume: 228, Issue: 4698, Publisher: Academic Press, 456-462.

Sonamthiang, S., Naruedomkul, K. & Cercone, N. (2006). *Initializing student models using Dempster-Shafer theory*. In Proceeding of Computers and Advanced Technology in Education 2006, Lima , Peru , October 4-6, 2006. 76-81.

Sonamthiang, S., Cercone, N. & Naruedomkul, K. (2007a). *A rough set based Map Granule*. Rough Sets and Intelligent Systems Paradigms. Lecture Notes in Computer Science, Springer Berlin /Heidelberg, 4585/2007, 290-299.

Sonamthiang, S., Cercone, N. & Naruedomkul, K. (2007b). *Discovering Hierarchical Patterns of Students' Learning Behaviour in Intelligent Tutoring Systems*. IEEE GrC 2007: 485-489.

Span, G. (1993). *LITES, an intelligent tutoring system for legal problem solving in the domain of Dutch Civil law*. International Conference on Artificial Intelligence and Law. Amsterdam, Netherlands. 76 – 81.

Tsumoto, S. (2002). *Accuracy and Coverage in Rough Set Rule Induction*. Rough Sets and Current Trends in Computing Lecture Notes in Computer Science, 2002, Volume 2475/2002

Virvou, M. & Tsiriga, V. (2001). *Web passive voice tutor: an intelligent computer assisted language learning system over the WWW*. Proceedings of the IEEE International Conference on Advanced Learning Technologies: Issues, Achievements and Challenges. IEEE Computer Society Press, Los Alamitos CA, 131-134.

Virvou, M. & Tsiriga, V. (2003). *Initializing Student Models in Web-based ITSs: a Generic Appproach*, proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies(ICALT’03), Athens, Greece.

Weber G. & Specht M. (1997). *User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems*. Eds.), User Modeling: Proceedings of the Sixth International Conference,UM97. Vienna, New York: Springer Wien New York.

Weber, G., & Brusilovsky, P. (2001). *ELM-ART: an adaptive versatile system for web-based instruction*. Journal of Artificial Intelligence in Education, 12, 351-384.

Yao, Y.Y. (2000). *Granular computing: basic issues and possible solutions*, Proceedings of the 5th Joint Conference on Information Sciences, Volume I, 186-189.

Yao, Y.Y. (2007), *Structured writing with granular computing strategies*, 2007 IEEE International Conference on Granular Computing, 72-77.

Yao, Y.Y., Zhou, B., Chen, Y. (2007). *Interpreting Low and High Order Rules: A Granular Computing Approach*. RSEISP 2007: 371-380.

Yao, Y.Y. (2009) *Integrative levels of granularity*, in: Bargiela, A. and Pedrycz, W. (Eds.) Human-Centric Information Processing Through Granular Modelling, Springer-Verlag, Berlin, pp. 31-47.

Yao, Y.Y. (2010). *Human-inspired granular computing*, J.T. Yao (Ed.), Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, Information Science Reference, Herskey, PA, 1-15.

Yara, P. O. (2009) *Students Attitude Towards Mathematics and Academic Achievement in Some Selected Secondary Schools in Southwestern Nigeria*. European Journal of Scientific Research Vol.36 No.3 (2009), pp.336-341.

Zadeh, L.A. (1979). *Fuzzy sets and information granularity*. In Advances in Fuzzy Set Theory and Applications. M. Gupta, R.K. Ragade, R.R. Yager (eds). North-Holland Publishing Company. 3-18.

Zadeh, L.A. (1997a). *Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic*, Fuzzy Sets and Systems, 19, 111-127.

Zadeh, L.A. (1997b). *Announcement of GrC*.

Basic Education Core Curriculum B.E. 2551 (A.D. 2008). Ministry of Education Thailand. (http://academic.obec.go.th/web/doc/d/147). Accessed June, 2012.

SimBionic (http://www.simbionic.com/indexjs.htm). Accessed June, 2012.

APPENDICES

APPENDIX A
PREREQUSITE CONCEPT RELATION OF FRACTION DOMAIN

This appendix illustrates the relations of concepts and its granular prerequisite in learning fraction based on the Basic Education Core Curriculum B.E. 2551 (A.D. 2008) Mathematics. Primary level students in Thailand start to learn fraction at grade 4, and the more complex concept and operations on fraction at grade 5 and grade 6 respectively. From the curriculum, we defined the prerequisite concept relation in the domain of fraction for primary level students as the following.

Fraction concepts to be learnt in grade 4:

1. Understand, write, and read fractions (C4_1)
   - C4_1_1 meaning of proper fractions
   - C4_1_2 reading fractions
   - C4_1_3 writing fractions
2. Compare and arrange sequence of fractions with the same denominator (C4_2)
   - C4_2_1Compare and arrange sequence of fractions with the same denominator
   - C4_2_2Arrange sequence of fractions with the same denominator
3. Operations on fractions with the same denominator (C4_3)
   - C4_3_1 Add fractions with the same denominator
   - C4_3_2 Subtract fractions with the same denominator

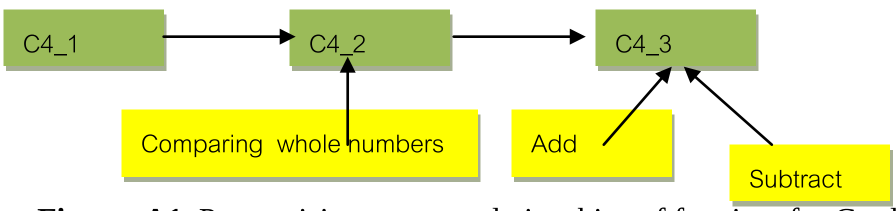


Figure A1. Prerequisite concept relationships of fraction for Grade 4

Fraction concepts to be learnt in grade 5:

1. Proper fraction, improper fraction, and mixed number (C5_1)
    - Converting mixed numbers to fractions (C5_1_1)
    - Converting improper fractions to mixed number (C5_1_2)
2. Fractions that equal whole numbers (C5_2)
3. Equivalent fractions (C5_4)
4. Reduced fractions or simplifying fractions (C5_5)
5. Compare and arrange sequence of fractions that have denominators that are multiples of each other (C5_6).
6. Conversion between fractions with dominators 10 and 100 to decimals and percents (C5_7)
7. Operations on Fractions (C5_8)
    - Addition and subtraction of fractions that have denominators that are multiples of each other (C5_8_1)
    - Multiplying fractions by whole numbers (C5_8_2)
    - Multiplying fractions by fractions (C5_8_3)
    - Dividing fractions by whole numbers (C5_8_4)
    - Dividing whole numbers by fraction (C5_8_5)
    - Dividing fractions by fractions (C5_8_6)

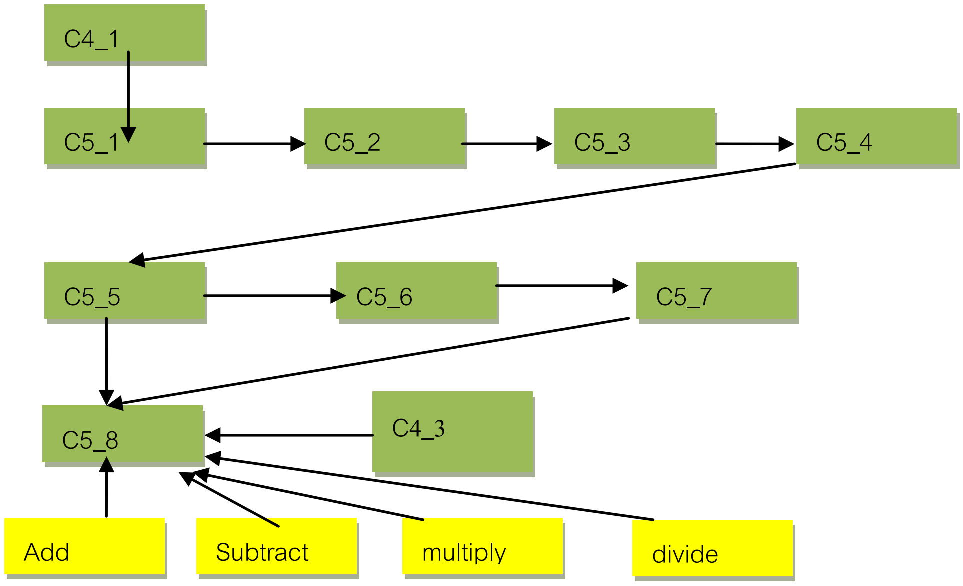


Figure A2. Prerequisite concept relationships of fraction for Grade 5

Fraction concepts to be learnt in grade 6:

1. Comparing and arranging sequence of fractions (C6_1)
2. Conversion between decimals and frations (C6_2)
   - Conversion 3 places decimal to fractions (C6_2_1)
   - Conversion between fractions with dominators 10, 100, and 1000 to decimals (C6_2_2)
3. Operation and fractions (C6_3)
   - Addition, subtraction, multiplication, and division of fractions (C6_3_1)
   - Addition, subtraction, multiplication, and division of mixed number (C6_3_2)
   - Mix operations on mixed number (C6_3_3)
   - Word problems of operation on fractions (C6_3_4)
   - Word problems of mix operation on fractions (C6_3_5)

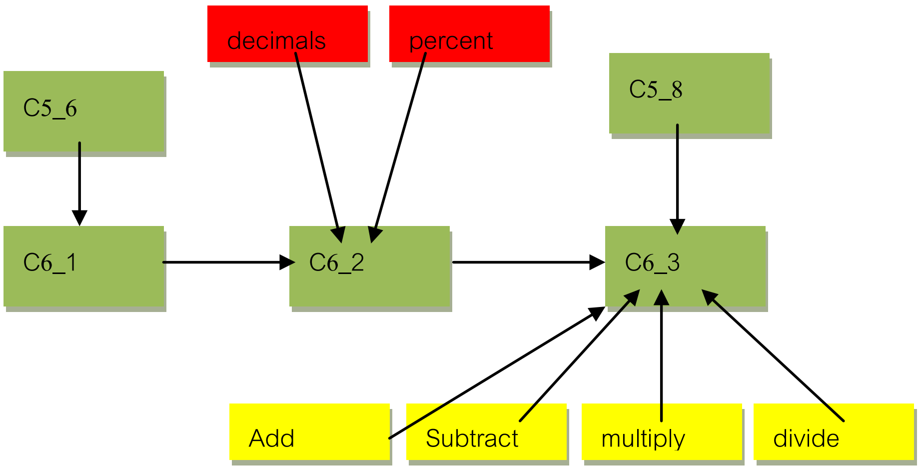


Figure A2. Prerequisite concept relationships of fraction for Grade 5

APPENDIX B
DATA SETS

This appendix presents datasets used in this thesis which are three student pretest data sets in B.1 and an animal data set in B.2.

B.1 Student pretest data set

This dataset was collected from 89 primary school students taking pretest on Fraction Friend Tutor as in Appendix A. The dataset contains 89 students from grade 4, 5, and 6 in academic year of 2010. The students were from Sansook School, Buriram Province, Thailand. In this information system, the first column represents student ID, followed by 16 test items and performance level in the last column.

	Test item number																
Student ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Performance level
401	1	1	1	1	0	1	1	0	0	1	0	1	1	0	0	0	3
402	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	2
403	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
404	0	0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	2
405	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1
406	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	2
407	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1
409	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	1
410	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1
411	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	2
412	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
413	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
414	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1
415	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
416	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
417	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
418	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1
419	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
420	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

421	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
423	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1
424	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	1
425	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
426	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
427	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
428	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1
429	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
430	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
431	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
432	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
500	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
501	1	1	0	1	0	0	1	0	1	0	0	1	1	0	1	0	2
502	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1
503	1	1	1	1	1	0	1	0	1	1	0	0	1	0	1	0	3
504	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	2
505	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	2
507	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	2
508	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1
509	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
511	1	1	0	1	1	1	1	0	1	0	0	1	1	0	1	0	3
512	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	2
513	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1
515	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1
516	1	1	0	1	1	1	1	0	1	0	0	1	1	0	1	0	3
516	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1
517	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
518	1	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	2
519	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1
520	1	1	0	1	0	0	0	0	1	0	0	0	1	0	1	0	2
521	0	1	0	1	0	0	1	0	1	0	0	0	0	0	1	0	2
522	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
523	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1
524	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
525	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1
526	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1
527	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1
528	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1
530	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1
531	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
532	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1
534	1	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	2

534	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
535	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
601	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	2
602	1	1	1	1	0	0	1	0	0	1	0	1	0	1	1	1	3
603	1	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	2
605	1	1	1	0	0	0	1	0	0	1	0	1	0	1	0	0	2
606	0	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	3
607	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	4
608	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	1	2
609	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	2
610	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1
611	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	2
612	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1
614	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1	3
615	1	1	1	1	0	0	1	0	1	0	0	0	0	1	0	0	2
615	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1
616	1	1	1	1	1	0	1	0	1	0	0	0	1	1	1	0	3
618	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	0	2
619	1	1	0	1	0	0	0	0	1	0	0	0	1	0	1	0	2
620	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1	1	3
622	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0	2
623	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1
624	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1
625	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1
626	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1
627	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	2
628	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1
629	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1

B.2 Animal data set

Animal dataset (Frank & Asuncion, 2010) can be obtained from the UC Irvine Machine Learning Repository (this data set is available online at [http://archive.ics.uci.edu/ml] with labeled Zoo data. The data set is described as follows.

Data Set Characteristics:	Multivariate	**Number of Instances:**	101	**Area:**	Life
Attribute	Categorical,	**Number of**	17	**Date**	1990-

Characteristics:	Integer	**Attributes:**		**Donated**	05-15
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	55754

Creator: Richard Forsyth

Donor:

Richard S. Forsyth

8 Grosvenor Avenue

Mapperley Park

Nottingham NG3 5DX

0602-621676

Data Set Information:

A simple database containing 17 Boolean-valued attributes. The "type" attribute appears to be the class attribute. Here is a breakdown of which animals are in which type: (I find it unusual that there are 2 instances of "frog" and one of "girl"!)

Class# -- Set of animals:

====== ==

1 -- (41) aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby,wolf

2 -- (20) chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren

3 -- (5) pitviper, seasnake, slowworm, tortoise, tuatara

4 -- (13) bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna

5 -- (4) frog, frog, newt, toad

6 -- (8) flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp

7 -- (10) clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

Attribute Information:

1. animal name: Unique for each instance
2. hair: Boolean
3. feathers: Boolean
4. eggs: Boolean
5. milk: Boolean
6. airborne: Boolean
7. aquatic: Boolean
8. predator: Boolean
9. toothed: Boolean
10. backbone: Boolean
11. breathes: Boolean
12. venomous: Boolean
13. fins: Boolean
14. legs: Numeric (set of values: {0,2,4,5,6,8})
15. tail: Boolean
16. domestic: Boolean
17. catsize: Boolean
18. type: Numeric (integer values in range [1,7])

Table B.2 The animal data set.

Animal	**hair**	**feathers**	**eggs**	**milk**	**airborne**	**aquatic**	**predator**	**toothed**	**backbone**	**breathes**	**venomous**	**fins**	**legs**	**tail**	**domestic**	**catsize**	**Class**
aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
buffalo	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
calf	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	1
carp	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	4
catfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
cavy	1	0	0	1	0	0	0	1	1	1	0	0	4	0	1	0	1

Animal	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	Class
cheetah	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
chicken	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2
chub	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
clam	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	7
crab	0	0	1	0	0	1	1	0	0	0	0	0	4	0	0	0	7
crayfish	0	0	1	0	0	1	1	0	0	0	0	0	6	0	0	0	7
crow	0	1	1	0	1	0	1	0	1	1	0	0	2	1	0	0	2
deer	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
dogfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	1	4
dolphin	0	0	0	1	0	1	1	1	1	1	0	1	0	1	0	1	1
dove	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2
duck	0	1	1	0	1	1	0	0	1	1	0	0	2	1	0	0	2
elephant	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
flamingo	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	1	2
flea	0	0	1	0	0	0	0	0	0	1	0	0	6	0	0	0	6
frog1	0	0	1	0	0	1	1	1	1	1	0	0	4	0	0	0	5
frog2	0	0	1	0	0	1	1	1	1	1	1	0	4	0	0	0	5
fruitbat	1	0	0	1	1	0	0	1	1	1	0	0	2	1	0	0	1
giraffe	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
girl	1	0	0	1	0	0	1	1	1	1	0	0	2	0	1	1	1
gnat	0	0	1	0	1	0	0	0	0	1	0	0	6	0	0	0	6
goat	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	1
gorilla	1	0	0	1	0	0	0	1	1	1	0	0	2	0	0	1	1
gull	0	1	1	0	1	1	1	0	1	1	0	0	2	1	0	0	2
haddock	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	0	4
hamster	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	0	1
hare	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	0	1
hawk	0	1	1	0	1	0	1	0	1	1	0	0	2	1	0	0	2
herring	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
honeybee	1	0	1	0	1	0	0	0	0	1	1	0	6	0	1	0	6
housefly	1	0	1	0	1	0	0	0	0	1	0	0	6	0	0	0	6
kiwi	0	1	1	0	0	0	1	0	1	1	0	0	2	1	0	0	2
ladybird	0	0	1	0	1	0	1	0	0	1	0	0	6	0	0	0	6
lark	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	2
leopard	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
lion	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
lobster	0	0	1	0	0	1	1	0	0	0	0	0	6	0	0	0	7
lynx	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1

Animal	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	Class
mink	1	0	0	1	0	1	1	1	1	1	0	0	4	1	0	1	1
mole	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	0	1
mongoose	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
moth	1	0	1	0	1	0	0	0	0	1	0	0	6	0	0	0	6
newt	0	0	1	0	0	1	1	1	1	1	0	0	4	1	0	0	5
octopus	0	0	1	0	0	1	1	0	0	0	0	0	8	0	0	1	7
opossum	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	0	1
oryx	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
ostrich	0	1	1	0	0	0	0	0	1	1	0	0	2	1	0	1	2
parakeet	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2
penguin	0	1	1	0	0	1	1	0	1	1	0	0	2	1	0	1	2
pheasant	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	2
pike	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	1	4
piranha	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
pitviper	0	0	1	0	0	0	1	1	1	1	1	0	0	1	0	0	3
platypus	1	0	1	1	0	1	1	0	1	1	0	0	4	1	0	1	1
polecat	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
pony	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	1
porpoise	0	0	0	1	0	1	1	1	1	1	0	1	0	1	0	1	1
puma	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
pussycat	1	0	0	1	0	0	1	1	1	1	0	0	4	1	1	1	1
raccoon	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
reindeer	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	1
rhea	0	1	1	0	0	0	1	0	1	1	0	0	2	1	0	1	2
scorpion	0	0	0	0	0	0	1	0	0	1	1	0	8	1	0	0	7
seahorse	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	0	4
seal	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	1
sealion	1	0	0	1	0	1	1	1	1	1	0	1	2	1	0	1	1
seasnake	0	0	0	0	0	1	1	1	1	0	1	0	0	1	0	0	3
seawasp	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	7
skimmer	0	1	1	0	1	1	1	0	1	1	0	0	2	1	0	0	2
skua	0	1	1	0	1	1	1	0	1	1	0	0	2	1	0	0	2
slowworm	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	3
slug	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	7
sole	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	0	4
sparrow	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	2
squirrel	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	0	1
starfish	0	0	1	0	0	1	1	0	0	0	0	0	5	0	0	0	7

Animal	**hair**	**feathers**	**eggs**	**milk**	**airborne**	**aquatic**	**predator**	**toothed**	**backbone**	**breathes**	**venomous**	**fins**	**legs**	**tail**	**domestic**	**catsize**	**Class**
stingray	0	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	4
swan	0	1	1	0	1	1	0	0	1	1	0	0	2	1	0	1	2
termite	0	0	1	0	0	0	0	0	0	1	0	0	6	0	0	0	6
toad	0	0	1	0	0	1	0	1	1	1	0	0	4	0	0	0	5
tortoise	0	0	1	0	0	0	0	0	1	1	0	0	4	1	0	1	3
tuatara	0	0	1	0	0	0	1	1	1	1	0	0	4	1	0	0	3
tuna	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	1	4
vampire	1	0	0	1	1	0	0	1	1	1	0	0	2	1	0	0	1
vole	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	0	1
vulture	0	1	1	0	1	0	1	0	1	1	0	0	2	1	0	1	2
wallaby	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	1	1
wasp	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	0	6
wolf	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
worm	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	7
wren	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	2

APPENDIX C
FRACTION PRETEST

This appendix presents the fraction pretest to evaluate students' prior knowledge towards fraction for primary level. The test comprises of 16 items. The fraction pretest is as follows.

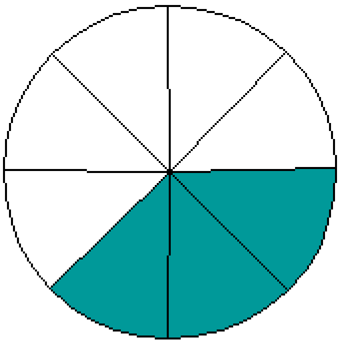
แบบทดสอบความรู้พื้นฐาน เรื่องเศษส่วน

ระดับประถมศึกษา

คำชี้แจง ข้อสอบชุดนี้มีทั้งหมด 16 ข้อ คะแนนเต็ม 16 คะแนน ข้อสอบมีทั้งแบบตัวเลือกและแบบเติมคำตอบให้นักเรียนเลือกคำตอบที่ถูกที่สุด 1 คำตอบแล้ววงกลมตัวเลือกที่ถูกที่สุด หรือเขียนคำตอบที่ถูกต้องลงในช่องว่าง

เวลาในการสอบ 1 ชั่วโมง

ข้อ 1) จากรูปข้อใดเขียนเศษส่วน แสดงส่วนที่แรเงาได้ถูกต้อง



ก. $\frac{3}{4}$ ข. $\frac{3}{5}$ ค. $\frac{8}{3}$ ง. $\frac{3}{8}$

ข้อ 2) เศษส่วนต่อไปนี้ข้อใดมีค่าเท่ากับ $\frac{1}{2}$?

ข. $\frac{3}{4}$ ข. $\frac{3}{6}$ ค. $\frac{2}{8}$ ง. $\frac{9}{15}$

ข้อ 3) ข้อใดคือเศษส่วนอย่างต่ำของ $\frac{9}{12}$

ก. $\frac{1}{3}$ ข. $\frac{3}{4}$ ค. $\frac{12}{9}$ ง. $\frac{7}{10}$

ข้อ 4) $\frac{3}{8} + \frac{2}{8} = \boxed{}$

ข้อ 5) 0.5 เขียนเป็นเศษส่วน คือ

ก. $\frac{1}{4}$ ข. $\frac{1}{2}$ ค. $\frac{1}{5}$ ง. $\frac{1}{3}$

ข้อ 6) เศษส่วนใดต่อไปนี้มีค่ามากที่สุด ?

ก. $\frac{1}{2}$ ข. $\frac{1}{3}$ ค. $\frac{1}{4}$ ง. $\frac{1}{5}$

ข้อ 7) $\frac{5}{5} =$?

ก. 5 ข. $\frac{1}{2}$ ค. 0 ง. 1

ข้อ 8) จงเรียงลำดับเศษส่วนต่อไปนี้จากน้อยไปมาก.

$\frac{1}{3}, \frac{2}{8}, \frac{1}{2}, \frac{3}{3}, \frac{6}{8}$ ....

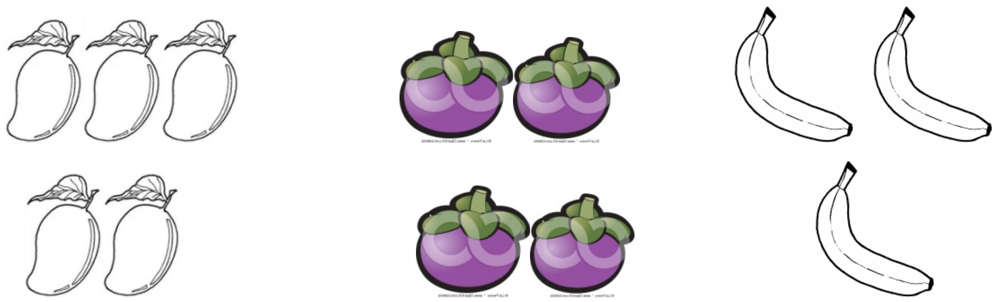
ข้อ 9) $\frac{9}{10} - \frac{5}{10} = \boxed{}$

ข้อ 10) จำนวนคละของ $\frac{18}{10}$ คือ $\boxed{}$

ข้อ 11) เศษเกินของ $2\frac{1}{4}$ คือ $\boxed{}$

ข้อ 12) จำนวนทศนิยมของ $\frac{7}{10}$ คือ $\boxed{}$

ข้อ 13) จากรูป มีผลไม้ทั้งหมด 12 ผล แบ่งเป็นมะม่วง 5 ผล มังคุด 4 ผล และกล้วย 3 ผล แล้วจำนวนมะม่วงต่อจำนวนผลไม้ทั้งหมด เขียนเป็นเศษส่วนได้ดังข้อใด



ก. $\frac{4}{12}$ ข. $\frac{5}{12}$ ค. $\frac{3}{12}$ ง. $\frac{3}{4}$

ข้อ 14) $\frac{5}{12} - \frac{1}{6} = \boxed{}$

ข้อ 15) $\frac{2}{3} \times \frac{1}{2} = \boxed{}$

ข้อ 16) $\frac{2}{3} \div \frac{1}{2} = \boxed{}$

APPENIX D
QUESTIONAIRE FOR USER SATISFACTION EVALUATION

This appendix contains the evaluation of RoughClust with respect to user satisfaction as presented in chapter 5. A questionnaire for teachers in Thai and in English is presented in D.1 and D.2 respectively.

D.1 Questionnaire for teacher satisfaction evaluation (Thai)

แบบสอบถามวัดความคิดเห็นของครูคณิตศาสตร์ต่อ

โปรแกรมช่วยจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่ต้องซ่อมเสริม (RoughClust)

คำชี้แจง

1. แบบสอบถามนี้จัดทำขึ้นเพื่อสอบถามความคิดเห็นของครูที่มีต่อการใช้โปรแกรมโปรแกรมช่วยจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่ต้องซ่อมเสริม แบ่งออกเป็น 2 ตอนประกอบด้วย

ตอนที่ 1 ข้อมูลผู้ตอบแบบสอบถาม

ตอนที่ 2 โปรแกรมช่วยจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่ต้องซ่อมเสริม

2. ในแต่ละข้อจะมีข้อความกำหนดให้ ให้นักเรียนอ่านข้อความให้เข้าใจ แล้วแสดงความคิดเห็นที่มีต่อข้อความนั้นด้วยการทำเครื่องหมาย ✓ ลงในช่องระดับความคิดเห็น ที่ตรงกับความรู้สึกของนักเรียนมากที่สุด ดังนี้

5	หมายความว่า	มีความเห็นด้วยอยู่ในระดับมากที่สุด
4	หมายความว่า	มีความเห็นด้วยอยู่ในระดับมาก
3	หมายความว่า	มีความเห็นด้วยอยู่ในระดับปานกลาง
2	หมายความว่า	มีความเห็นด้วยอยู่ในระดับน้อย
1	หมายความว่า	มีความเห็นด้วยอยู่ในระดับน้อยที่สุด

ตอนที่ 1 ข้อมูลผู้ตอบแบบสอบถาม

(1) เพศ ☐ ชาย ☐ หญิง

(2) อายุ .... ปี

(3) สอนวิชาคณิตศาสตร์ระดับชั้น ป. ....

(4) ความสามารถในการใช้คอมพิวเตอร์ ☐ ดีมาก ☐ ดี ☐ พอใช้ ☐ น้อย

<u>ตอนที่ 2</u> ความคิดเห็นต่อโปรแกรม ช่วยจัดกลุ่มนักเรียนตามความรู้พื้นฐ านและค้นหาหน่วยความรู้ที่ต้องซ่อมเสริม

รายการประเมิน	**ระดับความคิดเห็น**				
	5	**4**	**3**	**2**	**1**
ประสิทธิภาพและประโยชน์					
1. ROUGHCLUST มีประโยชน์ในการจัดกลุ่มนักเรียนตามความรู้พื้นฐาน					
2. ROUGHCLUST สามารถจัดกลุ่มนักเรียนตามความรู้พื้นฐานได้ถูกต้อง เหมาะสม					
3. ROUGHCLUST สามารถระบุหน่วยการเรียนรู้พื้นฐานที่นักเรียนต้องได้รับการซ่อมเสริมได้อย่างถูกต้อง เหมาะสม					
4. ROUGHCLUST แนะนำสื่อการสอนได้ตรงความรู้พื้นฐานของนักเรียน					
4. สื่อประกอบการสอนที่ ROUGHCLUST แนะนำมีประโยชน์					
5. การจัดกลุ่มนักเรียนด้วย ROUGHCLUST มีส่วนช่วยให้นักเรียน เรียนเรื่องเศษส่วนได้ดียิ่งขึ้น					
6. การจัดกลุ่มนักเรียนด้วย ROUGHCLUST มีส่วนช่วยให้นักเรียน เรียนอย่างมีความสุข และชอบเรียนวิชาคณิตศาสตร์มากขึ้น					
ความง่ายต่อการใช้งาน					
7. การใช้งาน ROUGHCLUST มีคำชี้แจงใช้งานชัดเจนและเข้าใจง่าย					
8. ROUGHCLUST ง่ายต่อการใช้งาน					
9. ท่านจะใช้ ROUGHCLUST ในการจัดกลุ่มนักเรียนและค้นหาความความรู้พื้นฐานที่นักเรียนควรได้รับการซ่อมเสริมต่อไป					
10. ท่านจะแนะนำครูท่านอื่นให้ใช้ ROUGHCLUST					

ROUGHCLUST มีประโยชน์ในการจัดกลุ่มนักเรียนหรือไม่ อย่างไร?

……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………

โปรแกรม ROUGHCLUST มีข้อเสียอะไรบ้าง?

……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………

ท่านใช้เวลาเรียนรู้การใช้งานโปรแกรม ROUGHCLUST นานเพียงใด

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

โปรดแสดงความคิดเห็นหรือข้อเสนอแนะต่อการปรับปรุงและพัฒนาโปรแกรม ROUGHCLUST

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

D.2 Questionnaire for teacher satisfaction evaluation (English)

Measurement	Scale				
	5	4	3	2	1
Usefulness					
1. Using RoughClust would help me in grouping student based on their prior knowledge					
2. RoughClust provides acceptable accurate and appropriate grouping for the student prior knowledge					
3. RoughClust suggests useful group's prerequisite concept requirement.					
4. Instructional materials suggested by RoughClust is useful and suitable for my students' requirement.					
5. Using RoughClust is helpful in my instruction planning.					
6. Using RoughClust to instruction planning can help my student learning improvement.					
7. Using RoughClust to Implement group-based instruction can help students attending the class more comfortable.					
Ease of use					
8. My interaction with RoughClust would be clear and understandable					
9. I would find RoughClust easy to use.					
10 I will continue using RoughClust.					
11. I will tell the other teacher to use RoughClust.					

How useful is the tool to group students and manage the teaching domain knowledge?..
………………………………………………………………………………………
………………………………………………………………………………………
What are the disadvantages of this tool? …………………………………………...
………………………………………………………………………………………
What does the tool need to improve?..
……………………………………………………………………………………..
How long until you can operate this tool by yourself?
………………………...
Do you have any suggestion about these tools?...
………………………………………………………………………………………
………………………………………………………………………………………

APPENDIX E
ROUGHCLUST DOCUMENTATION

This appendix illustrates the documentation of RoughClust in Thai.

คู่มือการใช้งาน RoughClust

1. RoughClust คืออะไร

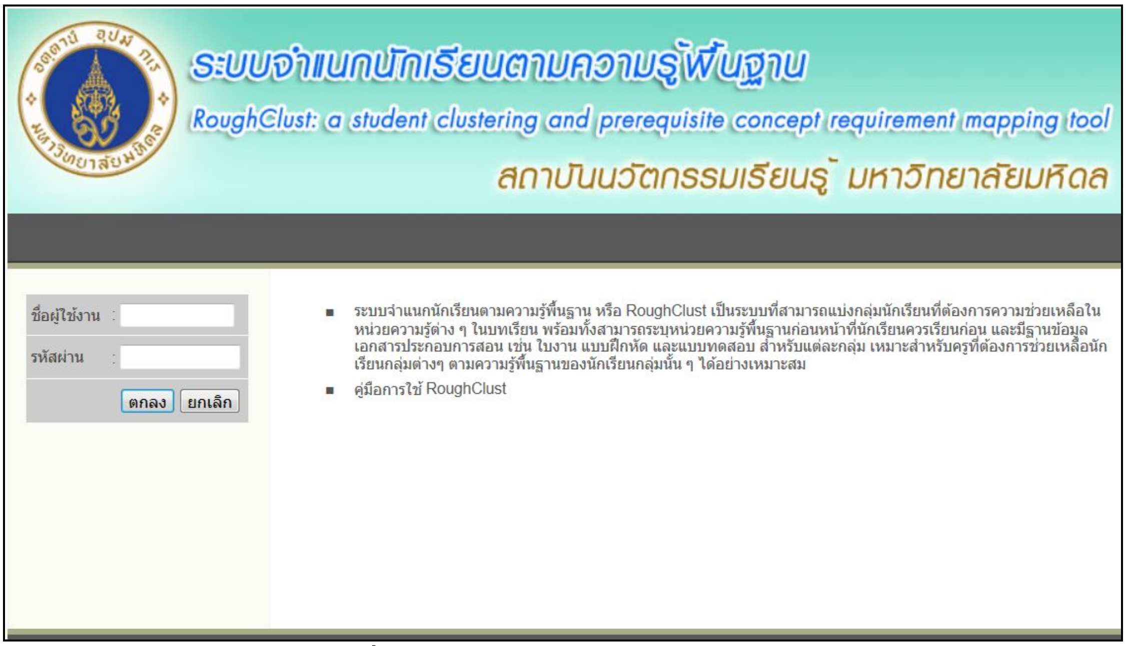
การวางแผนและการจัดการเรียนการสอนในห้องเรียน จำเป็นอย่างยิ่งที่ครูต้องคำนึงถึงความรู้พื้นฐานของนักเรียนที่มีมาก่อน ซึ่งจะส่งผลกระทบต่อการเรียนรู้หน่วยการเรียนรู้ได้หากนักเรียนยังขาดความรู้พื้นฐานที่จำเป็น การที่ครูจะทราบความรู้พื้นฐานที่จำเป็นที่นักเรียนขา ด ไปเป็นรายบุคคลนั้นใช้เวลามาก นอกจากนี้การจัดการเรียนการสอนให้เหมาะกับนักเรียนแต่ละคนก็เป็นไปได้ยาก ดังนั้นRoughClust จึงได้ถูกพัฒนาขึ้นเพื่อแก้ปัญหาดังกล่าว

RoughClust เป็นระบบการจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่จำเป็นต้องซ่อมเสริม เพื่อเป็นแนวทางให้ครูใช้วางแผนการจัดการเรียนการสอนให้เหมาะสมกับความรู้พื้นฐานของนักเรียนทั้งแบบกลุ่มและแบบรายบุคคลได้อย่างสะดวกและลดเวลา ในการทำงาน โดยใช้ระเบียบวิธีการจัดกลุ่มข้อมูลอย่างเป็นลำดับชั้นด้วยราฟเซต และข้อมูลคำตอบของนักเรียนจากแบบทดสอบความรู้พื้นฐาน ในการจัดกลุ่มนักเรียนตามความรู้พื้นฐานใช้ความสัมพันธ์แบบขึ้นต่อกันของหน่วยการเรียนรู้ในการค้นหาหน่วยการเรียนรู้ที่จำเป็นต้องได้รับการซ่อมเสริม และเตรียมสื่อการเรียนรู้ ไว้สำหรับครูให้ประกอบการทำแผนการสอนได้อย่างสะดวก

ผู้ใช้งานระบบสามารถใช้ RoughClust ออนไลน์โดยต้องมีโปรแกรม web browser เช่น Internet Explore, Firefox, Chrome เป็นต้น

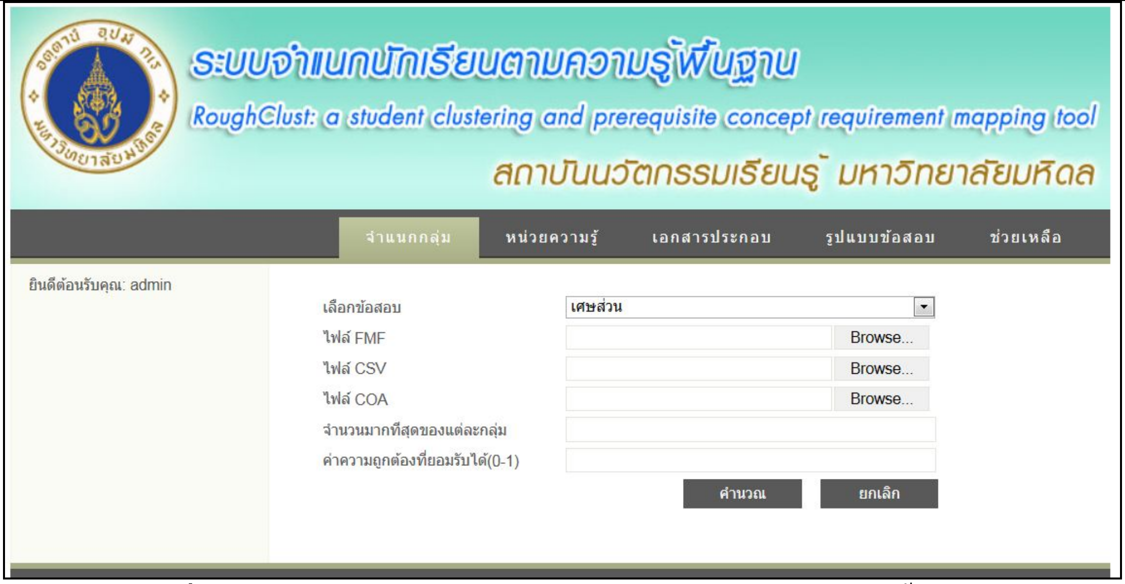
2. Graphical User Interface

2.1 หน้าจอลงทะเบียนเข้าใช้งานระบบ ก่อนใช้งาน RoughClut ผู้ใช้ระบบจะต้องทำการลงทะเบียนและยืนยันตัวตนผู้ใช้ระบบ



รูปที่ 1 หน้าจอลงทะเบียนเข้าใช้งานระบบ

2.2 หน้าจอการจัดกลุ่มนักเรียน เมื่อยืนยันตัวตนผู้ใช้งานแล้ว จะสามารถทำการจัดกลุ่มนักเรียนตามความรู้พื้นฐานและค้นหาหน่วยความรู้ที่จำเป็นต้องซ่อมเสริม โดยเลือกข้อสอบและ upload ไฟล์ผลสอบของนักเรียน รวมทั้งไฟล์นิยามข้อมูล และไฟล์กำหนดลำดับการใช้กลุ่มของข้อสอบเพื่อใช้ในการจัดกลุ่มข้อมูล ดังรูปที่ 2



รูปที่ 2 หน้าจอ upload input สำหรับการจัดกลุ่มนักเรียนตามความรู้พื้นฐาน

2.2.1 การเตรียมข้อมูล input

2.2.2.1 กำหนดนิยามข้อมูล เป็นการกำหนดว่า ข้อมูลผลการทดสอบความรู้พื้นฐานของนักเรียนนั้น มีนักเรียนกี่คน ชื่อหรือรหัสอะไรบ้าง และมีข้อสอบกี่ข้อ แต่ละข้อมีผลการทดสอบเป็นแบบใดได้บ้าง (เช่น ถูก = 1, ผิด =0) สุดท้ายผลสอบโดยรวมว่านักเรียนได้คะแนนในระดับใด เช่น 4= ดีมาก, 3 = ดี, 2= พอใช้, 1= ต้องปรับปรุง รูปแบบการนิยามข้อมูล แสดงไว้ในรูปที่ 3 ผู้ใช้สามารถใช้โปรแกรม editor เช่น notepad ในการสร้างไฟล์กำหนดนิยามข้อมูล โดยแถวแรกให้ระบุข้อมูลเกี่ยวกับนักเรียน แถวต่อไปเป็นข้อสอบทีละข้อ และแถวสุดท้ายเป็นระดับความสามารถของนักเรียน ดังตัวอย่างในรูปที่ 4 จากนั้นให้บันทึกนิยามข้อมูลนี้ไว้ โดยกำหนดนามสกุลของไฟล์เป็น .FMF

<X 0 ColumnName DataType Number_n value#1 value#2.... value#n>

<C 0 ColumnName DataType Number_i value#1 value#2.... value#i >

<C 0 ColumnName DataType Number_k value#1 value#2 ....value#k >

....

<C 0 ColumnName DataType Number_k value#1 value#2 ....value#k >

<D 0 class DataType Number_d value#1 value#2...value#d>

รูปที่ 3 รูปแบบการนิยามข้อมูล

```
<X 0 Students S 27 601 602 603 604 605 606 607 608
609 610 611 612 613 614 615 616 617 618 619 620
621 622 623 624 625 627 629>
<C 0 item1 S 2 0 1>
<C 0 item2 S 2 0 1>
<C 0 item3 S 2 0 1>
......
<D 0 class S 3 1 2 3 >
```

รูปที่ 4 ตัวอย่างการนิยามข้อมูล

2.2.2.2 ผลการทดสอบความรู้พื้นฐานของนักเรียน หลังจากได้ตรวจข้อสอบของนักเรียนแล้วครูสามารถนำผลการตรวจมาบันทึก อาจใช้โปรแกรม MS Excel ช่วยในการกรอกข้อมูล โดยให้แต่ละแถว แทนข้อมูลผลการสอบของนักเรียนหนึ่งคน ในคอลัมน์แรกเป็นชื่อหรือเลขที่ของนักเรียน คอลัมน์ต่อไปเป็นข้อมูลผลสอบ (ถูก = 1, ผิด =0) แต่ละข้อเรียงไปจนครบทุกข้อ ส่วนคอลัมน์สุดท้าย เป็นระดับของผลรวมคะแนนในข้อสอบชุดนี้ โดยอาจแบ่งเป็น 3 ระดับคือ (4= ดีมาก, 3 = ดี, 2= พอใช้, 1= ต้องปรับปรุง) เมื่อข้อมูลผลสอบเรียบร้อยแล้ว ให้บันทึกเป็นไฟล์ .CSV แล้วตรวจสอบความถูกต้องของข้อมูอีกครั้งให้แน่ใจว่า ไม่มีข้อมูลขาดหายไป หรือค่าของข้อมูลไม่อยู่ในช่วงค่าที่กำหนดไว้ (ดูการกำหนดนิยามข้อมูล)

```
401,1,1,1,1,0,1,1,0,0,1,0,1,1,0,0,0,3
402,0,0,0,1,0,0,1,0,1,0,1,0,1,0,0,0,2
403,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1
404,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,2
405,0,1,0,1,0,0,0,0,0,0,0,1,0,0,1,0,1
406,1,0,0,1,0,0,0,0,1,0,0,1,1,0,1,0,2
```

รูปที่ 5 ข้อมูลผลสอบของนักเรียน

2.2.2.3 กำหนดลำดับของ attribute ที่ใช้ในการจัดกลุ่มข้อมูล เป็นการกำหนดการขึ้นต่อกันของหน่วยความรู้ในข้อสอบที่ใช้ในการสอบ โดยข้อสอบที่เกี่ยวข้องกับหน่วยความรู้พื้นฐานจะถูกใช้ในการจัดกลุ่มนักเรียนก่อน ลำดับต่อไปตามด้วยข้อสอบที่เกี่ยวข้องกับหน่วยความรู้ที่ซับซ้อนขึ้น ตามลำดับ เช่นในรูปที่ 6 แสดงลำดับของข้อสอบที่ใช้ในการจัดกลุ่มนักเรียน ผู้ใช้สามารถกำหนดลำดับเหล่านี้โดยใช้โปรแกรม notepad ระบุชื่อของข้อสอบแต่

ละข้อ ใช้เครื่องหมาย “,” ในการแยกชื่อข้อสอบแต่ละข้อ และใช้การขึ้นบรรทัดใหม่ (enter) ในการแยกกลุ่มของข้อสอบ

```
item1,item13
item2,item3
item4,item9
item10,item11
item5,item7
item6,item8
item15,item16
item9
item12
```

รูปที่ 6 ลำดับของข้อสอบที่ใช้ในการจัดกลุ่มข้อมูลนักเรียน

3. การใช้งานโปรแกรม RoughClust ในการขยายขอบเขตหน่วยความรู้พื้นฐานของระบบ การจัดฐานข้อมูลเอกสารประกอบ และสร้างรูปแบบข้อสอบ

3.1 หน้าจอการจัดการหน่วยความรู้และความสัมพันธ์ของงหน่วยความรู้ มีเมนูย่อย 6 เมนู คือ

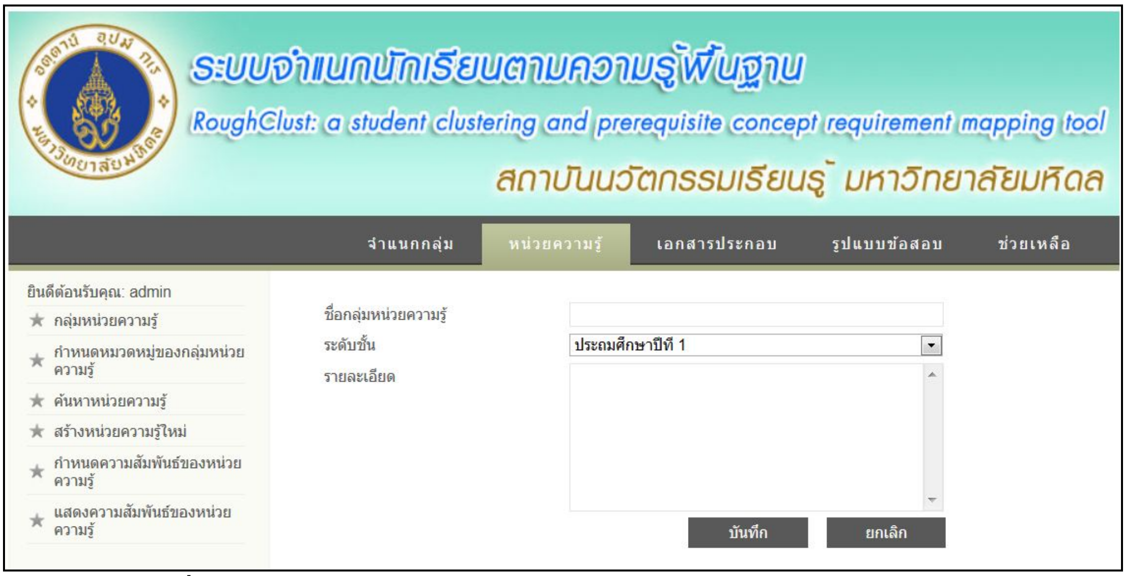
3.1.1 หมวดหมู่ของหน่วยความรู้ ใช้สร้างวิชา หรือ ระดับชั้น

3.1.2 กลุ่มของหน่วยความรู้ ใช้สร้างหน่วยความรู้หลัก ของวิชา หรือระดับชั้น

3.1.3 หน่วยความรู้ ใช้สร้าง ค้นหา และแก้ไขหน่วยความรู้

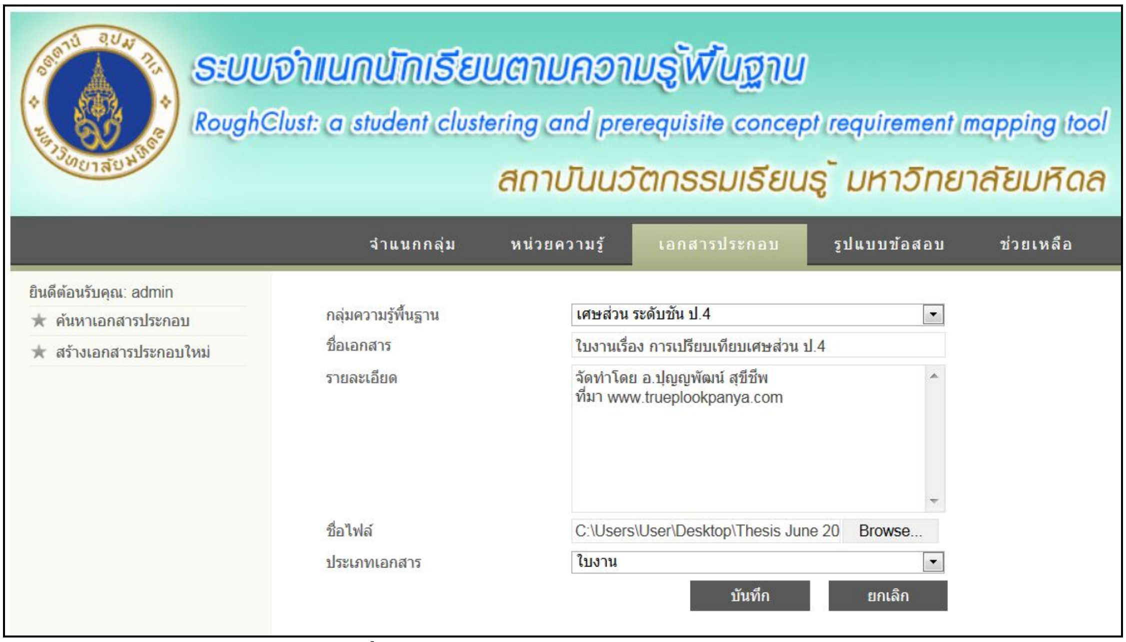
3.1.4 กำหนดความสัมพันธ์ของหน่วยความรู้ ใช้กำหนดความรู้พื้นฐานที่ต้องมีก่อนการเรียนหน่วยความรู้นั้น ๆ

3.1.5 แสดงความสัมพันธ์ของหน่วยความรู้ ใช้แสดงความสัมพันธ์ของหน่วยความรู้ต่างๆ กับความรู้พื้นฐาน



รูปที่ 7 หน้าจอการจัดการหน่วยความรู้และความสัมพันธ์ของงหน่วยความรู้

3.2 หน้าจอการจัดการเอกสารประกอบ ใช้ในการจัดการฐานข้อมูลใบงาน แบบฝึกหัด และข้อสอบที่สัมพันธ์กับหน่วยความรู้ต่าง ๆ



รูปที่ 8 หน้าจอการจัดการเอกสารประกอบ

3.3 หน้าจอการจัดการข้อสอบ ใช้ในการกำหนดความสัมพันธ์ระหว่างข้อสอบแต่ละข้อในชุดข้อสอบกับหน่วยความรู้ที่ใช้ในการทำข้อสอบข้อนั้น ๆ

รูปที่ 9 หน้าจอการจัดการข้อสอบ

3.4 หน้าจอช่วยเหลือ มีคู่มือการใช้งาน RoughClust รวมทั้งข้อมูลการติดต่อผู้วิจัย

รูปที่ 10 หน้าจอช่วยเหลือ

BIOGRAPHY

NAME	Sumalee Sonamthiang
DATE OF BIRTH	17 December 1979
PLACE OF BIRTH	Roi Et, Thailand
INSTITUTIONS ATTENDED	Mahasarakham University, 1998-2002 Bachelor of Science (Computer Science) Mahasarakham University, 2002-2003 Grad.Dip. Teaching (Computer Science) Mahidol University, 2003-2011 Doctor of Philosophy (Science and Technology Education) Dalhousie University, Nava Scotia, Canada 2005-2006 Visiting Student (VSGS) York University, Ontario, Canada, 2006-2007 Visiting Researcher
RESEARCH GRANTS	-
HOME ADDRESS	99/163 Parichard Village, Khumklao Road, Lampakchee Sub District, Ladkrabang Bangkok, 10520 Tel. (+66) 02-3635435 E-mail : tew197@gmail.com
EMPLOYMENT ADDRESS	Department of Mathematics Statistics and Computer, Faculty of Science, Ubon Ratchathani University

PUBLICATION / PRESENTATION

Sonamthiang, S., Naruedomkul, K. & Cercone, N. (In Press) *Granular Concept Mapping and Applications*. Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam Vol.1, Series: Intelligent Systems Reference Library, Vol. 42, Skowron, A.; Suraj, Z. (Eds.) Springer Berlin Heidelberg, New York, 2013, 581-596.

Sonamthiang, S., Cercone, N., & Naruedomkul, K. (2007). *Discovering hierarchical patterns of students' learning behavior in Intelligent Tutoring Systems.* In Proceedings of the 2007 IEEE International Conference on Granular Computing. Silicon Valley, USA , November 2-4, 2007. 485-489.

Sonamthiang, S., Cercone, N., & Naruedomkul, K. (2007). *A Rough Set Based Map Granule*. Rough Sets and Intelligent Systems Paradigms. Lecture Notes in Computer Science, Springer Berlin /Heidelberg, Vol. 4585/2007, 290-299.

Alshammari, R., Sonamthiang. S., Teimouri, M., & Riordan D. (2007). *Using Neuro-Fuzzy approach to reduce false positive alerts*. In Proceeding of the Fifth Annual Conference on Communication Networks and Services Research, Fredericton , New Brunswick , Canada , May 14 - 17, 2007. 345-349.

Sonamthiang, S., Naruedomkul, K., & Cercone, N. (2006). Initializing student models using Dempster-Shafer theory. *In Proceeding of Computers and Advanced Technology in Education 2006,* Lima, Peru , October 4-6, 2006. 76-81.

Rattanawongchaiya, N., Sonamthiang, S. & Naruedomkul, K. (2005) Thai intelligent tutor with information retrieval. In *2005 Proceedings of IEEE Integration of Knowledge Intensive Multi-Agent Systems* (pp. 549 – 554). Boston, MA., April 2005, 18-21.