A MAJORITY DENSITY APPROACH WITH THE COOPERATION OF MULTIPLE EXPERTS FOR DEVELOPING TESTING AND DIAGNOSTIC LEARNING SYSTEMS BASED ON A CONCEPT-EFFECT RELATIONSHIP MODEL

DECHAWUT WANICHSAN

**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

A MAJORITY DENSITY APPROACH WITH THE COOPERATION OF MULTIPLE EXPERTS FOR DEVELOPING TESTING AND DIAGNOSTIC LEARNING SYSTEMS BASED ON A CONCEPT-EFFECT RELATIONSHIP MODEL

……………….………..…..………
Mr. Dechawut Wanichsan
Candidate

……………………….…..…..………
Miss Patcharin Panjaburee, Ph.D.
Major advisor

……………………….…..…..………
Mr. Parames Laosinchai, Ph.D.
Co-advisor

……………………….…..…..………
Assoc. Prof. Wannapong Triampo, Ph.D.
Co-advisor

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

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

Thesis
entitled

A MAJORITY DENSITY APPROACH WITH THE COOPERATION OF MULTIPLE EXPERTS FOR DEVELOPING TESTING AND DIAGNOSTIC LEARNING SYSTEMS BASED ON A CONCEPT-EFFECT RELATIONSHIP MODEL

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Doctor of Philosophy (Science and Technology Education)
on
September 27, 2012

……………….…………..…..………
Mr. Dechawut Wanichsan
Candidate

……………………..…...………………
Prof. Gwo-Jeng Hwang, Ph.D.
Chair

……………………..…...………….……
Miss Patcharin Panjaburee, Ph.D.
Member

……………………..…...……………...…
Assoc. Prof. Wannapong Triampo, Ph.D.
Member

……………………..…...………………..
Mr. Parames Laosinchai, 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

I would like to sincerely thank my great advisor, Dr. Patcharin Panjaburee, for her assistance, guidance, understanding, and most importantly, her friendship during my graduate studies. She also stands by my side when I feel exhausted or need some helps during conducting the research. My graduate life is easier thanks to her.

I would like to gratefully thank Dr. Parames Laosinchai who is a brilliant and distinguished thinker. The majority-density algorithm would not be completely developed without his idea and suggestions; moreover, he is also my inspiration to improve my English skill since he nearly got full marks in TOEFL.

I greatly appreciate Assoc. Prof. Dr. Wannapong Triampo, my grand advisor, who has given me several useful comments and suggestions to reveal beneficial aspects of this dissertation. I really think he is a generous teacher although he often tries to make me feel that he is cruel.

I also thank Miss Sasithorn Chookaew, my lovely friend, for everything she has done for me. She often provides me for some humor and entertainment to reduce stressful environments. Additionally, I am very grateful for the friendship of all friends that are not mentioned.

I thank all my current and past teachers at Mahidol University, Chulalongkorn University, Phranakorn Si Ayutthaya Rajabhat University, Thamai Phulsawat Ratnukul School, Streemandapitak School and Joseph Vittaya School. They have educated me not only for academic matter but also for my life. Without them, I would not be here.

Finally, and most importantly, I am grateful to my beloved father and mother for their faith in me and allowing me to be as ambitious as I wanted. Their unconditional love and supports are very invaluable in my life. I love them so much.

Dechawut Wanichsan

A MAJORITY DENSITY APPROACH WITH THE COOPERATION OF MULTIPLE EXPERTS FOR DEVELOPING TESTING AND DIAGNOSTIC LEARNING SYSTEMS BASED ON A CONCEPT-EFFECT RELATIONSHIP MODEL

DECHAWUT WANICHSAN 5237247 ILSE/D

Ph.D.(SCIENCE AND TECHNOLOGY EDUCATION)

THESIS ADVISORY COMMITTEE : PATCHARIN PANJABUREE, Ph.D., PARAMES LAOSINCHAI, Ph.D., WANNAPONG TRIAMPO, Ph.D.

ABSTRACT

In the past decade, testing and diagnostic learning systems have been considered as a useful tool for analyzing students' conceptual learning problems and providing helpful learning suggestions for them to improve their conceptual learning outcome. Among the existing methods for developing testing and diagnostic learning systems, a multi-expert approach based on a CER (Concept-Effect Relationship) model was proposed using a set of rules to integrate test item–concept relationship opinions from multiple experts. However, there were some drawbacks when integrating the opinions from multiple experts that might affect the quality of learning suggestions for individual students. Furthermore, it was time consuming to reconsider their opinions when conflicting opinions existed. Therefore, in this study, a new method was proposed to overcome the drawbacks of the previous work. In addition, a practical testing and diagnostic system on a “Computer Programming” course for undergraduate students was implemented to demonstrate the effectiveness of this innovative approach. To evaluate students' knowledge, they took an achievement test after receiving the suggestions and learning materials from the system. By analyzing the results of students after receiving learning suggestions from two different systems, it was found that the conceptual learning outcome of the students who received learning suggestions from the proposed system was significantly better than that of those who received guidance based on the previous system implying that the current approach is more effective than the previous one.

KEY WORDS: CONCEPT-EFFECT RELATIONSHIP MODEL / COMPUTER-BASED TESTING / TESTING AND DIAGNOSTIC SYSTEM / EXPERT SYSTEM / MAJORITY DENSITY APPROACH

123 pages

การใช้รูปแบบความหนาแน่นและเสียงส่วนมากจากความร่วมมือของผู้เชี่ยวชาญหลายคนเพื่อพัฒนาระบบแบบทดสอบและวินิจฉัยผลการเรียนโดยใช้แบบจำลองความสัมพันธ์ระหว่างมโนมติกับผลลัพธ์

A MAJORITY DENSITY APPROACH WITH THE COOPERATION OF MULTIPLE EXPERTS FOR DEVELOPING TESTING AND DIAGNOSTIC LEARNING SYSTEMS BASED ON A CONCEPT-EFFECT RELATIONSHIP MODEL

เดชาวุฒิ วานิชสรรพ์ 5237247 ILSE / D

ปร.ด. (วิทยาศาสตร์และเทคโนโลยีศึกษา)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : พัชรินทร์ ปัญจบุรี Ph.D., ปรเมศวร์ เหล่าสินชัย Ph.D., วรรณพงษ์ เตรียมโพธิ์ Ph.D.

บทคัดย่อ

ในช่วงทศวรรษที่ผ่านมา ระบบแบบทดสอบและวินิจฉัยปัญหาการเรียนได้รับการพิจารณาว่าเป็นเครื่องมือที่มีประโยชน์สำหรับการวิเคราะห์ปัญหาการเรียนเชิงมโนมติของนักเรียน ตลอดจนยังสามารถให้คำแนะนำที่มีประโยชน์ต่อการพัฒนาความรู้ของนักเรียนอีกด้วย ในบรรดาระบบแบบทดสอบและวินิจฉัยปัญหาการเรียนจำนวนมากที่ได้รับการพัฒนาขึ้นมานั้น วิธีการใช้กฎเพื่อรวมค่าความเห็นเกี่ยวกับความสัมพันธ์ระหว่างข้อสอบแต่ละข้อและมโนมติของผู้เชี่ยวชาญหลายคนบนแบบจำลองความสัมพันธ์ระหว่างมโนมติกับผลลัพธ์ได้รับการเสนอขึ้น อย่างไรก็ตาม วิธีการนี้ยังมีข้อบกพร่องซึ่งอาจส่งผลกระทบต่อคุณภาพของคำแนะนำที่ให้กับนักเรียนเป็นรายบุคคลได้ นอกจากนี้เมื่อความเห็นของผู้เชี่ยวชาญทั้งหลายไม่เป็นไปในทิศทางเดียวกัน ทำให้ผู้เชี่ยวชาญต้องเสียเวลาในการพิจารณาค่าน้ำหนักเหล่านั้นอีกครั้ง ดังนั้นในงานวิจัยนี้ วิธีการที่สามารถเอาชนะจุดอ่อนงานก่อนหน้าจึงได้รับการเสนอขึ้น อนึ่งระบบแบบทดสอบและวินิจฉัยปัญหาการเรียนเชิงมโนมติยังได้รับการพัฒนาขึ้นจริงโดยใช้กับวิชาการเขียนโปรแกรมคอมพิวเตอร์สำหรับนักศึกษาระดับปริญญาตรี เพื่อแสดงให้เห็นประสิทธิผลของวิธีการที่นำเสนอ สำหรับการประเมินความรู้เชิงมโนมติในการเขียนโปรแกรมคอมพิวเตอร์ของนักศึกษานั้น แบบวัดความรู้ได้ถูกนำไปใช้หลังจากที่นักเรียนได้รับคำแนะนำและได้รับใบความรู้จากระบบแบบทดสอบที่สร้างขึ้น จากการวิเคราะห์ผลการเรียนของนักเรียนพบว่าผลการเรียนของนักเรียนที่ได้รับคำแนะนำจากระบบจากงานวิจัยนี้มีคะแนนสูงกว่านักเรียนที่รับคำแนะนำจากระบบของงานวิจัยก่อนหน้าอย่างมีนัยสำคัญ จะเห็นได้ว่าวิธีการใหม่ที่ได้รับการพัฒนาขึ้นจากงานวิจัยนี้มีประสิทธิภาพกว่าวิธีเดิม

123 หน้า

CONTENTS

CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CER	Concept-Effect Relationship
KEISC	Knowledge Elicitation and Integration System for Determining the Weights of Concepts
TDLP	Testing and Diagnostic Learning Problem

CHAPTER I
INTRODUCTION

The first chapter aims to provide the background, motivation and significance of this study in the development of a testing and diagnostic learning system based on a majority-density approach. The system was developed emphasizing for a computer programming course because it has been considered as one of the difficult subjects for undergraduate students. Then, the research questions and research objectives are explained to frame this study. After that, the importance of this study shows the usefulness of the system in education and other related domains. The research instruments for collecting data are also shown in this chapter. Finally, an organization of this thesis is shown as an outline of the remaining chapters.

1.1 Background and motivation

In normal situation, after finishing a course, the student conceptual learning outcome would be evaluated by testing in order to obtain a total score or grade. However, these test results were inadequate for students to improve their conceptual learning when they failed to do the test or got low scores. Learning guidance was so necessary and so helpful for students in difficult situations (Gerber, Grund, & Grote, 2008). Because computer, internet, and communication technologies have been ubiquitous, touching human life, online testing and diagnostic systems were developed and also played important roles in diagnosing students' conceptual learning problems, especially based on concept mapping (Bai & Chen, 2008a, 2008c; Casamayor, Amandi, & Campo, 2009; C.-M. Chen, 2008; G.-J. Hwang, 2003, 2007; Panjaburee, Hwang, & Shih, 2010).

A concept map was a diagram representing relationships among many concepts that students had to learn (Kao, Lin, & Sun, 2008; Liu, Don, & Tsai, 2005). It was a useful tool that could promote meaningful conceptual learning. Therefore, several testing and diagnostic systems employed it as an important part of them. There were two environments of the systems using concept maps i.e., a system that automatically generated a concept map from a student's testing (Bai & Chen, 2008a), and a system that designed a concept map in a well-defined order before it was applied for testing and diagnosing students (Bai & Chen, 2008c; C.-M. Chen, 2008; S.-M. Chen & Bai, 2009; Günel & Aşlıyan, 2010; G.-J. Hwang, 2003; Panjaburee et al., 2010). This work focused on developing a testing and diagnostic system in the latter environment. For instance, Chen (C.-M. Chen, 2008) used a genetic algorithm as a core of the system to detect students' learning paths based on their incorrect answers.

Hwang (G.-J. Hwang, 2003) proposed a concept-effect relationship (CER) model presenting relationships of several concepts to develop an intelligent tutoring system; however, this system supported only one expert that could make some mistakes while determining relationships. Therefore there was a research (Panjaburee et al., 2010) introducing a multi-expert testing and diagnostic system using a set of rules to integrate the opinions of multiple experts to one value before serving as an input for the testing and diagnostic system. Nevertheless, the previous work might have some drawbacks for the integration of weighting values in a testing and diagnostic learning system because the integrating method considered only opinions of some experts and omitted the rest. Due to these limitations, unreliable and low-quality learning suggestions were provided to students; moreover, when several cases of conflicts existed, it was time-consuming for experts to recheck their values.

To handle these drawbacks, a majority density approach to integrating the relationship values of each concept and test item from multiple experts was proposed. This method considered majority opinions from several experts, and was a pragmatic method to generate reasonable integrated weighting values for developing a testing and diagnostic learning system. The resultant system could accurately detect conceptual learning problems and provide useful conceptual leaning suggestions to students.

1.2 Difficulty of learning computer programming

Nowadays, the rapid growth of information technology makes computer very important to modern human life. Consequently, in various universities, several departments such as computer engineering, computer science, software engineering, information technology, computer business, and computer education have been created. One of the fundamental subjects that students in these programs have to learn is "Computer Programming". Moreover, not only does computer programming play an important role for student in computer areas, but a lot of students in related fields also have to learn this subject as one of the core subjects. This course could help students to develop ideas and logic of programming through four standard tasks of the system development life cycle (SDLC), i.e., analysis, design, implementation, and maintenance. The first step, analysis, involves studying a given problem statement. In the design step, the programmer writes a plan to solve the problem using pseudocode or a flow chart. Implementation means translating ideas from the previous step into the programming code of a certain programming language. The last step, maintenance, consists of testing and improving the program until it meets the requirements.

This subject mainly focuses on improving students' skill in the implementation step. Students have to write a programming code by using a computer language. It has been considered a difficult task because they have to remember the syntax of the computer language and they have to apply them while writing a code to solve a problem. In reality, students often lack the ability to combine single statements from different concepts in order to construct a whole program (Eckerdel, 2009; Soloway & Spohrer, 1989; Winslow, 1996). Most importantly, students are also rarely aware of the problems that can be solved by a computer (Lemone & Ching, 1996). Some of the problems from each concept in a computer programming course have some relationships to each other; for example, variables and data types are fundamental concepts of array.

Due to these reasons, many students who cannot grasp the most fundamental concepts of programming are unable to produce basic programs and also unable to learn and understand more complicated concepts in the future (Eckerdel, 2009). Therefore, it might be better if we could find an appropriate way to improve their conceptual learning ability to produce basic programs and learn advanced concepts in the computer science area.

1.3 Significance of this study

Due to the difficulty of learning computer programming mentioned in the previous section, it is very important to know what are the real causes of conceptual learning problems of students or what topics or concepts that they do not have enough knowledge. Therefore, the main objective of this research is to propose a majority density algorithm for developing a testing and diagnostic learning system with the cooperation of multiple experts that can generate accurate learning suggestions based on learning paths in a concept-effect relationship model for individual students so as to help them to realize their drawbacks, and to improve their conceptual learning outcomes. This system is used for diagnosing conceptual learning problems of students; moreover, it can serve as one of the important parts in an adaptive learning system.

Not only do the results of this work directly impact on the educational domain, but they also are useful in other applications such as medicine and group decision-making. In addition, the success of this study plays an important role enhancing the effectiveness of the entire computer-based adaptive learning systems which are developed based on the concept-effect relationship model.

1.4 Research questions and research objectives

This study was conducted to investigate the following research questions:

(1) To what extent is the testing and diagnostic learning system using the majority density approach helpful to the students in improving their conceptual learning outcome?

(2) Is the testing and diagnostic learning system using a majority density approach more effective in improving students' achievement compared to the system using Panjaburee et al. (2010)'s approach?

(3) To what extent can the testing and diagnostic learning system using the majority density approach reduce conflicting cases and the number of reconsidering cases compared with the system using Panjaburee et al. (2010)'s approach?

(4) How do students feel after receiving learning guidances from the testing and diagnostic learning system?

To optimize the testing and diagnostic learning system scheme, four aims of this study were followed:

(1) To develop a majority density algorithm in order to integrate the weights of associations between test items and concepts for a testing and diagnostic learning system.

(2) To develop a testing and diagnostic learning system to diagnose students' conceptual learning problems and provide conceptual learning guidance for them.

(3) To examine the students' conceptual learning outcome after receiving learning guidance from the enhanced learning diagnosis model based on the proposed approach and the original model.

(4) To investigate the students' satisfaction with the learning suggestions provided by the testing and diagnostic learning system.

1.5 Research tools

There are several tools and instruments used in this research as follows:

(1) Testing and diagnostic learning system: it is developed using the PHP programming language and MySQL as the database management system.

(2) Diagnostic test (pre test): it is embedded as one part of the system. It is constructed to measure prior conceptual knowledge of students in a computer programming course, at the same time, it is used for diagnosing their conceptual learning problems.

(3) Conceptual test (post test): it is used for evaluating the conceptual learning outcome of students after receiving treatments from this research.

(4) Questionnaire: it is used for measuring the learning satisfaction of students with the testing and diagnostic learning system.

1.6 Organization of the dissertation

There are five main steps to conduct the research, i.e., (1) investigate related works about testing and diagnostic learning systems, especially based on the concept-effect relationship diagram, (2) detect some drawbacks of previous works, (3) develop a system to prove the existing drawbacks, (4) develop some method to solve the problems and develop a practical system, (5) evaluate the performance and usefulness of the developed system by measuring the learning outcome and learning satisfaction of students compared with those using the original system proposed by previous researchers. All of the steps are organized through following chapters:

The first chapter introduces and outlines this research work. It introduces the background and some related works about the development of testing and diagnostic learning systems; in addition, it shows how significance this research is. Research purposes and research questions are also introduced . Moreover, tools for conducting the research are briefly introduced here. Finally, it provides an organization of the dissertation.

The literature review will be shown in chapter 2. It presents related works in deep detail about developing testing and diagnostic learning systems, especially the systems based on the concept-effect relationship (CER) model. Moreover, it focuses on exploring some drawbacks of the previous works, and provides explicit examples to reveal existing problems.

In chapter 3, methodology for conducting the research including a majority density algorithm for integrating the weighting values of multiple experts in order to solve the problems of previous works and to develop the testing and diagnostic learning system based on the concept effect relationship (CER) model is proposed. Furthermore, a practical developed system for a computer programming course that can diagnose conceptual learning problems of students and can provide learning suggestions for them to improve their knowledge is presented. Moreover, the experimental design and research instruments (pre-test, post-test, and questionnaire) are provided in this chapter.

The experimental results are shown in chapter 4. The analysis of the pre-test is used for claiming that there is no difference in prior knowledge of the students in experimental and control groups. The analysis of the post-test is used to compare the students' conceptual learning outcomes between both groups. The satisfaction of students after receiving the suggestions from the testing and diagnostic learning system is revealed by using the analysis of the questionnaire. The saving in experts' time of the proposed method is also shown in the analysis of number of reconsidering cases. The contribution of the proposed work and guidelines for applying the proposed approach into the classroom are discussed in this chapter.

The last chapter, chapter 5, presents the conclusions of this study. Limitations of the study and suggestions for developing future works are also included in this chapter.

CHAPTER II
LIRERATURE REVIEW

2.1 Testing and diagnostic system

After students had studied some subjects in any course, their conceptual learning outcomes should be evaluated via testing in order that they would know their successes and failure in the course through their scores. These results would not be helpful for them to improve themselves when they wanted to receive feedbacks or suggestions from a teacher. Nevertheless, this situation was difficult to cope with because the number of teachers compared to the number of students in a classroom was very inappropriate, and a teacher needed to spend a lot of time to analyze learning problems of each student. According to these drawbacks and because of the rapid growth of information and communication technology including internet and computer technology, there was various research interest in computer-assisted testing and diagnostic learning systems (Bai & Chen, 2008a; G.-J. Hwang, 2003; Y.-C. Lin, Lin, & Huang, 2011; Martin, 2001; Panjaburee et al., 2010; Stankov, Rosić, Žitko, & Grubišić, 2008; J. C. R. Tseng, Chu, Hwang, & Tsai, 2008). A computer-based and internet-based testing system was used for analyzing and detecting students' conceptual learning problems, and providing useful conceptual learning suggestions and learning materials for students to improve their knowledge. It also helped teachers to save the time for diagnosing individual students.

Since 1960, computers have been used in the education area (Martin, 2001). They were used by both teachers and students in several aspects, for examples, assisting teaching, preparing a handout for students, collecting records of students, conducting research, and searching data for doing a homework. With their popularization, computers play an important role for teachers and students. In 1965, an individually guided education (IGE) program was developed by researchers of the University of Wisconsin (Klausmeier, 1976). The aims of the program were to organize and deliver educational experiences from research teams that studied how people learn and how to provide instruction processes for individual students. In IGE, a student's profile could be used to determine appropriate guidance and learning materials. LISP Tutor was proposed by (Anderson & Reiser, 1985) in order to teach the programming language LISP. The system had an interesting function used for comparing a student's performance to that of an expert using the "model tracing". It responded to an individual student based on the expert problem-solving path; it provided advice for guiding the student back onto the (expert's) path. Adaptive hypermedia (Brusilovsky, 1996) was the idea that the course content in the system should be adjusted for a student based on the profile or personal record. An adaptive hypermedia system should be composed of three components:

1. It should have hypermedia
2. It should have a user model from the user's profile
3. The hypermedia could be adjusted depending on the user model.

Paolucci (Paolucci, 1998) ensured the importance of the hypermedia system that could adapt to an individual student. The adaptive hypermedia systems should be able to diagnose and identify learning problems of each student. Diagnosing conceptual learning problems of students was crucial in the research area of adaptive learning systems. In recent years, approaches/models/algorithms that have been used for developing such testing and diagnostic systems have included Bayesian cybernetics, fuzzy rules, genetic algorithms, clustering techniques, and concept-effect relationship models (Bai & Chen, 2008a; C.-M. Chen, 2008; S.-M. Chen & Bai, 2009; Cheng, Lin, Chen, & Heh, 2005; Kaburlasos, Marinagi, & Tsoukalas, 2008) and other techniques (L.-H. Chen, 2011; B. Jong, Lin, Wu, & Chan, 2004; Y.-C. Lin et al., 2011; Manning & Dix, 2008; Stankov et al., 2008) For example, Cheng et al. (2005) proposed using a hierarchical clustering algorithm to categorize students into several groups depending on their testing results and each group consisted of students who shared similar learning problems. Bai & Chen (2008a) proposed a method based on fuzzy rules that used testing results of students to construct a concept map representing a relationship among nodes. Kaburlasos et al. (2008) proposed a Bayesian and statistical model to choose test items for each student during testing; each of them faced a different set of test items based on his/her answer to each test item. Chen (2008) developed a web-based learning system based on a genetic algorithm that provided learning path guidance for students depending on their incorrect responses. A hypermedia-based English Learning system for Prepositions (HELP) was developed (Lo, Wang, & Yeh, 2004); it was the system for non-native speakers of English that could diagnose conceptual learning problems of students and provide remedial lessons according to their assessment results. Bai and Chen (2008b) proposed a method for automatically constructing grade membership functions of fuzzy rules for student evaluation. Bai and Chen (2008c) developed a method for learning barrier diagnosis of students in adaptive learning systems. Lee et al. (2009) showed a method for automatically constructing concept maps and conceptual diagnosis of e-learning. Agrawal & Srikant (1994) proposed using an a-priori algorithm to automatically construct the concept maps to diagnose the learning problem of students. Casamayor et al. (2009) developed adaptive learning systems based on the personal features or learning behaviors of students.

A concept map was a diagram representing a relationship among concepts that was used in several contexts. Many researchers used it as a tool to reflect students' knowledge at the conceptual level by letting them draw and design by themselves (G.-J. Hwang, Wu, & Ke, 2011; Kao et al., 2008). Moreover, another type of concept map comprising each concept linked in hierarchical order was called a concept-effect relationship (CER) model (G.-J. Hwang, 2003). It was a useful tool to determine the cause(s) of learning failure, provide helpful personalized feedback, and guide the teachers in which concept a student needed to improve. This work was proved to be a practical system which helped students in science courses to improve their knowledge by providing learning suggestions to individual students. For this reason, several researchers have been researching into the concept-effect relationship model to develop testing and diagnostic systems (Bai & Chen, 2008a, 2008c; C.-M. Chen, 2008; S.-M. Chen & Bai, 2009; Günel & Aşlıyan, 2010; Panjaburee et al., 2010). Not only did the successful cases of the concept-effect relationship model–based testing and diagnostic systems demonstrate the benefits of applying the concept-effect relationship model to cope with learning diagnosis problems, but they also depicted the difficulty of applying it. For example, Hui Chun Chu, Hwang, Tseng, & Hwang (2008) presented a learning diagnosis approach based on the concept-effect relationship model to provide personalized learning suggestions to students by analyzing their test results. Jong, Chan, and Wu (2007) developed a learning-behavior diagnosis system applied to a computer course at a university and the experimental results showed improvements in both learning status and learning achievement. Tseng et al. (2007) employed this model to provide useful learning guidance for individual students in the physics course at a junior high school. However, these systems supported only one expert for knowledge acquisition, and human errors could easily appear. Therefore, based on the idea of the collaborative teachers, a multiple expert approach was proposed by Panjaburee et al. (2010) to determine the weighting values for each test item to the specified concepts by integrating the opinions of multiple experts. Obviously, this model has been successfully applied in order to detect the learning problems of students and to give personalized suggestions in several domains, including natural sciences, mathematics, physics, engineering, and life science courses.

2.2 The concept-effect relationship (CER) model and knowledge elicitation method to develop testing and diagnostic system

The concept-effect relationship (CER) diagram was first introduced by Hwang in 2003. It was a diagram which represented the prerequisite relationships among concepts that students needed to learn in a well-defined order. All steps used for developing a testing and diagnostic learning system based the CER model were depicted in Figure 2.1.

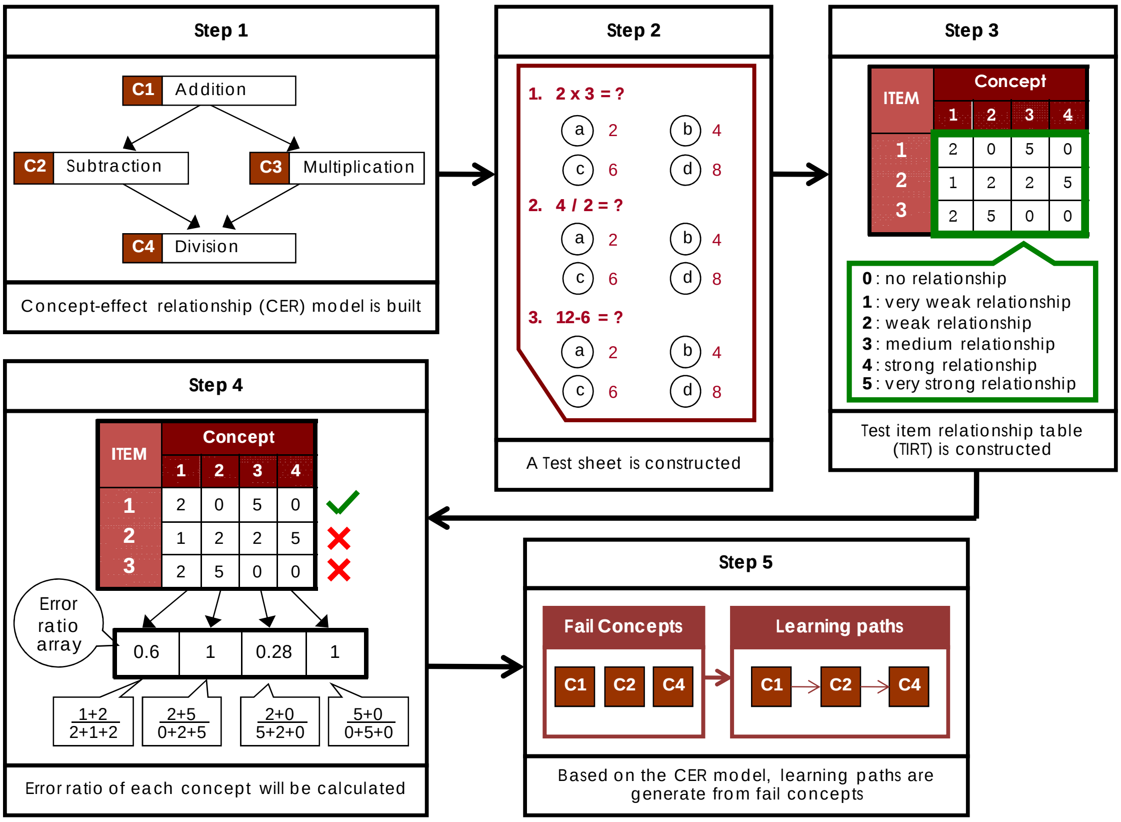


Figure 2.1 An example of using the CER model to develop a testing and diagnostic learning system

According to Figure 2.1, an example of a testing and diagnostic learning system based on a CER model was developed for a mathematics course at an elementary school. To construct the CER model was the first task to do. In the first step, there were four different concepts that students needed to learn, i.e., Addition, Subtraction, Multiplication, and Division. All of them were linked in a well-defined order. C_1 (Addition) was the concept that students should first learn before learning other concepts, i.e., C_2 (Subtraction) and C_3 (Multiplication); in other words, C_1 was more fundamental than any other concept. In the same way, a student who learned C_4(Division) required first learning both C_2 and C_3. From these concepts in the CER model, there were two possible learning paths:

PATH1: C_1:Addition $\rightarrow$ C_2 Subtraction $\rightarrow$ C_4:Division and

PATH2: C_1:Addition $\rightarrow$ C_3:Multiplication $\rightarrow$ C_4:Division.

This model considered the relationship between a prior and posterior knowledge while planning personalized learning paths by determining the conceptual learning problems of each student and provided conceptual learning guidance for individual students by tracing the concept-effect relationship diagram. For example, a student failed to learn C_2. One of the possible reasons might be learning problems of the student for the concept “Subtraction”. It might also be that s/he did not have enough understanding in the prerequisite concept C_1. In this case, the student should be suggested that s/he should study the concept “Addition” more thoroughly before learning the more complex concept “Subtraction”.

In step 2, an expert or a teacher needed to design and construct a multiple-choice test in order to be used as a diagnostic test. Each test item should have some relationship to at least one of the concepts of the predefined CER model, shown in step 1 of Figure 2.1. An illustrative example of a test sheet was depicted in step 2 of Figure 2.1 which contained three test items, and each test item comprised four choices.

The following step, step 3, was the most important step because it elicited the knowledge of an expert and transferred into the knowledge base of a testing and diagnostic system. The knowledge would be obtained from an expert (in this case a teacher was an expert). The main role of an expert was to determine weighting values representing relationships among concepts and each test item represented by a test-item relationship table (TIRT). The value “0” indicated “no relationship”, “1” indicated “very weak relationship” , “2” indicated “weak relationship” , “3” indicated “medium relationship” , “4” indicated “strong relationship” , and “5” indicated “very strong relationship”. Step 3 in Figure 2.1 represented an example of TIRT; the rows of the TIRT represented the three test items (related to step 2) and the columns of the TIRT represented the four concepts (related to step 1). The example of determining weighting values showed that the weighting values of item 1 for C_1 to C_4 were “2”, “0”, “5”, and “0” respectively because item 1 was a question about C_3, which also needed to use some knowledge about C_1 as a basic knowledge.

When knowledge was completely acquired from an expert for all test items, the test would be used as a computer-based diagnostic test. In the next step, after the test sheet was completed by a student, it was important to know what were the concepts that the student had difficulty learning. Therefore the error ratio (ER) of each concept would be calculated by dividing the summation of the weighting values of the test items that the student failed to answer by that of all the weighting values of all the test items. If ER of a concept was greater than a predefined threshold θ, it would be considered a failed concept of the student. For example, in step 4 of Figure 2.1, the student failed to answer items 2 and 3 and the calculated ERs were 0.6, 1, 0.28, and 1 for the concepts C_1 to C_4; in this case, θ was assumed to be 0.4, and failed concepts were C_1, C_2, and C_4, which would be used as nodes for detecting learning paths based on the CER model. Therefore, conceptual learning problems of the student were misunderstanding of C_1, C_2, and C_4; C_1 should be learned before C_2 and C_4; in addition, a student should learn C_2 before C_4.

However, this approach supported only a single expert, and human error due to tiredness, inadequate knowledge and ignorance would easily appeared while determining weight values; therefore, it would be a good idea if there were multiple experts that collaboratively determine the weighting values. They could help each other to verify their knowledge; moreover, all of them could discuss or interchange their opinions when conflict happened. Panjaburee et al. (2010) proposed a set of rules used for integrating weighting values from multiple experts. In the next section, Panjaburee et al.'s work would be briefly introduced and some drawbacks would also be shown.

2.3 The multi-expert testing and diagnostic system based on the CER model

Although the developed system based on a single expert was easy to implement, it was not effective enough to use because only one expert could make some mistakes due to insufficient knowledge, ignorance, or subjective opinion when providing knowledge; therefore, using a single expert may not always be appropriate. Several researchers (Chu & Hwang, 2008; Huang & Shimizu, 2006; G.-H. Hwang, Chen, Hwang, & Chu, 2006; Léger & Naud, 2009; Medsker, 1995; Mittal & Dym, 1985) suggested that several experts in the same domain with diverse experience could have different expertise or understanding of each portion of the knowledge. It is a more effective way to recognize and reject incorrect solutions and suggestions compared to a single-expert approach.

The first testing and diagnostic system for education that could work with multiple experts was proposed by Panjaburee et al. (2010). Each individual expert needed to give a weighting value to show the relationships between each test item and concept. The weighting value ranging from “1” to “5” was used to show the degree of relationship, corresponding to “very weak”, “weak”, “medium”, “strong”, and “very strong” relationships respectively. In addition, “X” represented “no relationship”. Moreover, another value that an expert had to give together with a weight relationship was degree of confidence, where “S” and “N” represented “high confidence” and “low confidence” respectively, which was used to help an expert to easily make a decision on giving weighting value. Furthermore, the weighting values that were more than 3, that is 4 and 5, were called “strong-side weighting values”, the weighting values that were less than 3, that is 1 and 2 not including X, were called “weak-side weighting values”, and the value “3” could be called either a “strong-side” or “weak-side” weighting value. The core of the system was a set of rules used for integrating weighting values given by multiple experts. All of the rules depicted in Table 2.1 were categorized into four different groups as follows: (1) Integration rules for the same value with different confidence degree, (2) Integration rules for the values on the same side with different confidence degree, (3) Integration rules for the values with “X”, and (4) Integration rules for the value on different sides.

Table 2.1 The fourteen rules and their output

Rule#	Condition	Integrated weight	Certainty level
1	All experts assign the same weight; moreover, the number of experts who assign the value with *high confidence* is more than that of experts who assign the value with *low confidence*	The assigned value	“S”

Rule#	Condition	Integrated weight	Certainty level
2	All experts assign the same weight; moreover, the number of experts who assign the value with *high confidence* is less than that of experts who assign the value with *low confidence*	The assigned value	“N”
3	All experts assign *weak side* values; moreover, the number of experts who assign the value with *high confidence* is more than that of experts who assign the value with *low confidence*	The minimum of *weak side* weights	“S”
4	All experts assign *weak side* values; moreover, the number of experts who assign the value with *high confidence* is less than that of experts who assign the value with *low confidence*	The minimum of *weak side* weights	“N”
5	All experts assign *strong side* values; moreover, the number of experts who assign the value with *high confidence* is more than that of experts who assign the value with *low confidence*	The minimum of *strong side* weights	“S”
6	All experts assign *strong side* values; moreover, the number of experts who assign the value with *high confidence* is less than that of experts who assign the value with *low confidence*	The minimum of *strong side* weights	“N”
7	Some experts assign the weight “X” with *high confidence*; moreover, some experts assign *weak side* values or *strong side* values with *high confidence.*	Reconsidering weights	-
8A	Some experts assign the weight “X” with *low confidence*; moreover, some experts assign *weak side* values with *high confidence.*	The minimum of *weak side* weights	“S”
8B	Some experts assign the weight “X” with *low confidence*; moreover, some experts assign *strong side* values with *high confidence.*	The minimum of *strong side* weights	“S”
9A	Some experts assign the weight “X” with *high confidence*; moreover, some experts assign *weak side* values with *low confidence.*	“X”	“S”

Rule#	Condition	Integrated weight	Certainty level
9B	Some experts assign the weight “X” with *high confidence*; moreover, some experts assign *strong side* values with *low confidence.*	“X”	“N”
10A	No expert assigns the weight with *high confidence*, and some experts assign the weight “X” with *low confidence*; moreover, some experts assign *weak side* values with *low confidence*	The minimum of *weak side* weights	“N”
10B	No expert assigns the weight with *high confidence*, and some experts assign the weight “X” with *low confidence*; moreover, some experts assign *strong side* values with *low confidence*	The minimum of *strong side* weights	“N”
11	Some experts assign *weak side* values with *high confidence*; moreover, some experts assign *strong side* values with *high confidence*.	Reconsidering weights	-
12	Some experts assign *weak side* values with *low confidence*; moreover, some experts assign *strong side* values with *low confidence*.	Reconsidering weights	-
13	No expert assigns “X”, and some experts assign *weak side* values with *low confidence*; moreover, some experts assign *strong side* values with *high confidence*.	The minimum of *strong side* weights	“S”
14	No expert assigns “X”, and some experts assign *weak side* values with *high confidence*; moreover, some experts assign *strong side* values with *low confidence*.	The minimum of *weak side* weights	“S”

The fourteen rules used mathematical symbols including ∀ (for all) and ∃ (for some) as an important part to integrate weighting values from multiple experts; however, there was a vague point due to using the symbol ∃. Consequently, different integrated results could be generated from different rules; moreover, rules using ∃ would not consider the majority opinion from multiple experts. An example of this drawback in using rule#7 to integrate the opinions from five experts for concept#2 and item 2 was shown in Table 2.2.

Table 2.2 Examples of using rule 7 and its integrated weight

Rule 7					
E_i	E_1	E_2	E_3	E_4	E_5
Weighting (E_i, Q_2, C_3)	X	X	5	X	4
Certainty (E_i, Q_2, C_3)	S	N	S	S	N
Integrated result	Ask the experts to check and reconsider their weighting values				

The example in Table 2.2 revealed some interesting points, i.e. more than one rule could be chosen to generate different results, and all opinions of multiple experts were not considered. The first rule that could be chosen to integrate the weighting values from multiple experts was rule 7; however, this rule used only values of E_1 and E_3 as input values while others were omitted to generate the result "reconsidering", where all experts had to check and reconsider their opinions. Another rule that could be used to integrate values was rule 8B to provide "5" with "S" as an integrated result; however, this rule used only values of E_2 and E_3 as input values while others were omitted to generate the output. Another applicable rule was rule 9B that used the opinions of E_1 and E_5 to generate "0" with "N" as a result. The details of rule#7, rule#8B and rule#9B and their outputs were depicted in Table 2.1.

According to this example, it was clear that using Panjaburee et al.'s approach revealed some limitations and could be unreliable. First, more than one rule could be chosen for integrating weighting values and different rules generated different results. Furthermore, because of the lack of majority consideration, a large number of reconsideration would appear and experts had to spend a lot of time reconsidering their values to solve all conflicting cases. In addition, only the integrated weighting value would be used in a testing and diagnostic learning system even though the set of rules could provide both the integrated weighting value and degree of confidence. Moreover, the most important factor affecting the performance of a testing and diagnostic learning system was the integrated weighting values, which were used as an input so as to analyze students' learning problems and provide learning advice based on learning paths. However, low-quality suggestions could be generated from low-quality integrated weighting values. An illustrated example was depicted in Figure 2.2.

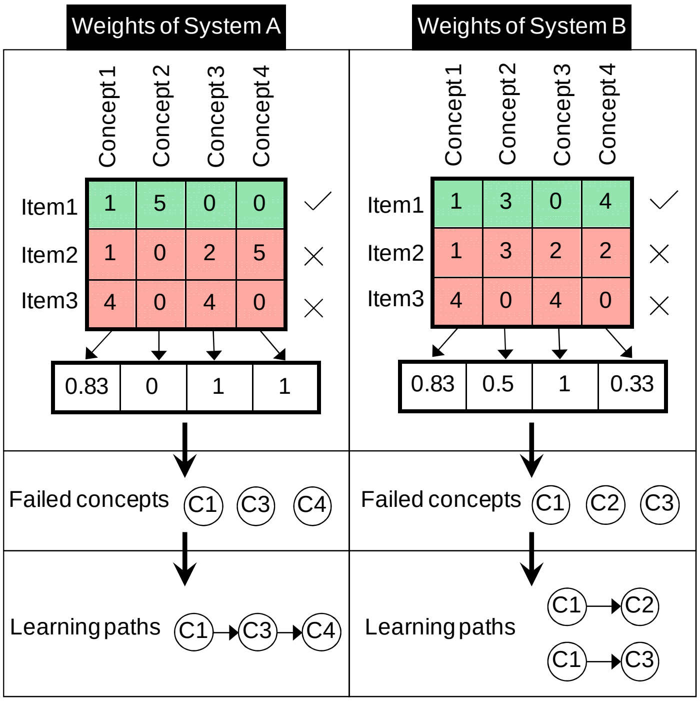


Figure 2.2 Different integrated weighting values resulting in different learning paths: ✓ represents correct answer and × represents incorrect answer

According to Figure 2.2., there were two testing and diagnostic learning systems, System A and System B, composed of different integrated weighting values from domain experts in the same test sheet. The differences among those values were in test items#1 and #2, and concepts #2 and #4. In this situation, a student answered test items 2 and 3 incorrectly. It was clear that the different error ratios among concepts resulted in different learning paths given to the student.

Due to the drawbacks of the approach, the problem might result from one of the experts making mistakes while providing knowledge via weighting values. The mistakes might come from insufficient knowledge or ignorance. Later, the weighting values were chosen by some rules to integrate weighting values. Consequently, the system might provide several unreliable results and a large number of reconsidering cases. Therefore, it was a good idea to remove the abnormal opinions of some experts before integrating all weighting values to develop a multi-expert testing and diagnostic learning system.

2.4 Anomaly Detection

Anomaly detection is a method or an algorithm which is used for detecting the data that are not in the same patterns or do not have the same behavior as most data. The detected data are called anomalies, changes, outliers, exceptions, deviations, or intrusions depending on the applications. Anomaly detection may involve various disciplines such as data mining, machine learning, artificial intelligence, information theory, and statistics; it applies these concepts to solve specific problems. Anomaly detection can be presented in several applications such as network-intrusion detection (Atallah, Gwadera, & Szpankowski, 2004; Gwadera, Atallah, & Szpankowski, 2005), fraud detection (Aggarwal, 2005; Ghosh & Reilly, 1994), medical anomaly detection (J. Lin, Keogh, Fu, & Herle, 2005; Wong, Moore, Cooper, & Wagner, 2003), industrial damage detection (Basu & Meckesheimer, 2007; Keogh, Lonardi, & Chiu, 2002), image processing (Pokrajac, Lazarevic, & Latecki, 2007; Singh & Markou, 2004), textual anomaly detection (Manevitz & Yousef, 2002; Srivastava, 2006), and sensor networks (Branch, Szymanski, Giannella, Wolff, & Kargupta, 2006; Du, Fang, & Ning, 2006).

Output of anomaly detection applications can be reported in one of two ways: *scores* or *labels*. For *scores*, the higher the degree that an instance is considered an anomaly, the higher the possibility that that instance is a true anomaly; there is a threshold value used to separate "*abnormal*" data from "*normal*" data. For *Labels*, each instance is clearly classified to be either "normal" or "abnormal". Anomaly detection can be categorized into three main modes based on data labeling. They are:

(1) Supervised anomaly detection: this mode requires a data set containing instances that both labeled as "*normal*" and "*abnormal*". These data are called training data; they are used to construct a classifier or predictive model. An unseen instance is compared to the constructed model and is classified as "*normal*" or "*abnormal*".

(2) Semi-supervised anomaly detection: sometimes labeled data are difficult and expensive to obtain, especially for "*abnormal*" data, because data are labeled by human experts. This mode requires only "*normal*" data and anomalies are not required. This mode will often be used when the training data set does not cover or has only a small amount of "*abnormal*" data.

(3) Unsupervised anomaly detection: training data are not necessary for this mode. All data are unlabeled and "*normal*" data will be defined based on the frequency and the majority of all data, while anomalies are instances which appear with low frequency or have different characteristics compared to the majority.

Some of the popular techniques for anomaly detection are classification-based anomaly detection techniques (based on the assumption that a classifier, constructed from training data, is able to classify between normal data and anomalies), clustering-based anomaly detection techniques (based on the assumption that normal data have to belong to at least one cluster while anomalies do not belong to any cluster), and nearest neighbor–based anomaly detection techniques (based on the assumption that normal data reside in the same area while anomalies are distant from other normal data).

According to the introduction of related works, it becomes an interesting and challenging issue to develop a novel method, a majority density approach, to integrate the opinions of multiple experts (teachers). This method can remove abnormal weighting values before integrating them; it makes a good quality knowledge base for the system. Therefore, in the next chapter, the new procedure for eliciting and integrating test item–concept relationships with the cooperation of multiple experts will be introduced to overcome the limitations of the rule-based approach to develop a multi-expert testing and diagnostic learning system based on the concept-effect relationship (CER) model.

CHAPTER III
METHODOLOGY

Due to the difficulties and problems of Panjaburee et al's model (2010) mentioned in the previous chapter, this chapter presents an exploratory research conducted in November 2010. This algorithm was developed three important characteristics: (1) majority, (2) density, and (3) taking out outliers. These characteristics might be useful for solving problems of Panjaburee et al's work. It was a novel approach for integrating the weighting values of the association between each test item and each concept from multiple experts. Moreover, the proposed method, a majority-density algorithm, was developed to be a full-scaled algorithm that could remove more than one outlier, and was used as a crucial part of a testing and diagnostic conceptual learning system for a computer programming course in the main study in Thailand. The experimental design and research instruments (pre-test, post-test, and questionnaires) were also provided.

3.1 Exploratory research

The overview of the exploratory research that was conducted in November 2010 is described in this section. It was used for examining the possibility of using a majority-density algorithm in the context of a multi-expert system. The promising results were used as the starting point for developing a full-scaled testing and diagnostic learning system.

3.1.1 The necessity of exploratory research

Although the multi-expert approach for developing a testing and diagnostic learning system that was introduced in the previous chapter yields a good result in improving conceptual learning outcome of individual students, it still had some drawbacks of applying the algorithm and was unreliable due to the lack of majority consideration, and the omission of the degree of confidence which could result in some inaccurate conceptual learning suggestions for students. Therefore, the main objective of the exploratory research was to investigate the possibility of using a proposed majority-density algorithm for managing weighting values given by multiple experts.

3.1.2 A majority density algorithm (exploratory version)

An algorithm used to integrate weights from several experts was developed to possess three important characteristics, and its name, majority density algorithm, also came from these properties:

(1) Majority: the weight values which are similar to each other should affect the integrated opinion more than those of the minority. An easy example of a majority is depicted in Figure 3.1.

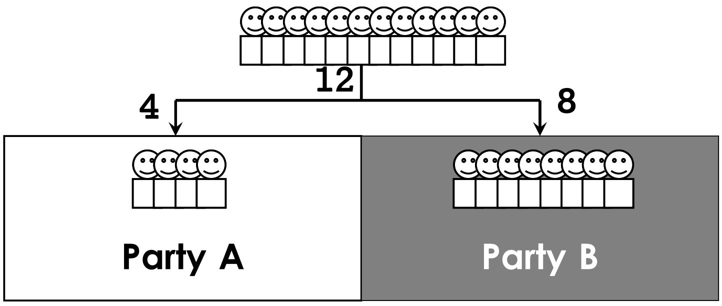


Figure 3.1 An example of majority

From Figure 3.1, there are twelve people who need to choose one of two parties. Party A is chosen by four people; on the contrary, Party B is chosen by eight people. Consequently, the winner is Party B because the majority of people choose it.

(2) Density: since the scale of the weights is broad, the density at a given weight is the degree to which the area in the data space around that weight is filled with other weights. Figure 3.2 shows an easy example of a dense area.

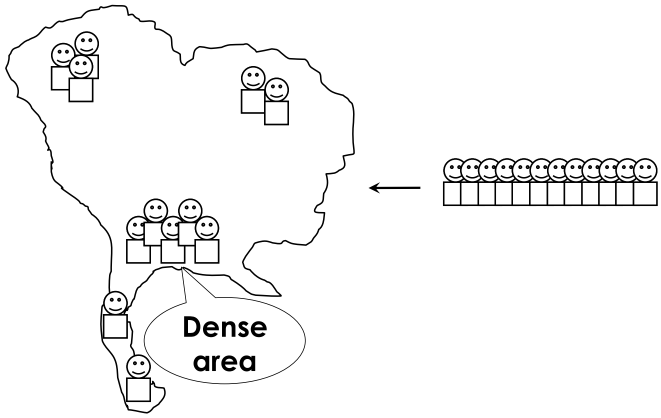


Figure 3.2 An example of a dense area

From Figure 3.2, there are twelve people on a map of Thailand. The most dense area is in the central region of the map where there are five people although they are not situated in the same position. the other areas have at most three people.

(3) Taking out outliers: some weight values that are different from most can be defined as a minority and they are removed before calculating the integrated weight. An easy example of data that can be defined as outliers is depicted in Figure 3.3.

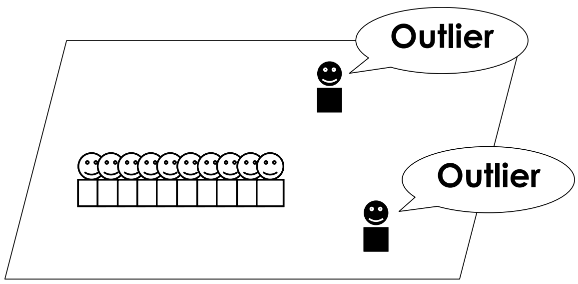


Figure 3.3 An example of outliers

From Figure 3.3, there are twelve people in a field. There are ten people who are white, and two people who are black. In this case, two black people are considered outliers since they have a different characteristic or location compared with the majority of people (ten white people).

In developing a testing and diagnostic system, an expert is faced with a difficult situation to assign a weighting value that reflects the relationship between a test item and a concept. Therefore, an expert should be able to indicate how confident he/she is about the assigned weight. The value of the degree of confidence could be either “S” or “N”, where “S” represents “high confidence” and “N” represents “low confidence”.

The first step of this proposed work was to adjust weighting values to be quantitative. Because a weighting value with “high confidence” should not be equal to a weighting value with “low confidence”. The value with “low confidence” should be adjusted. The most basic idea of adjusting a weighting value is to use multiply the old value by a constant; however it is not good enough because some strong-sided weighting values can be switched to the opposite side after being multiplied; for instance, “4” (a strong-sided weighting value) times “0.5” (a constant) is “2” (the adjusted value which is a weak-sided value). Therefore, an idea of adjusting all weighting values with “low confidence” to a suitable one should comprise two basic ideas shown below:

(1) A weighting value should not become less or more than its adjacent value; for examples, “5” should not be decreased to be less than “4” after it is adjusted.

(2) All weighting values should be adjusted toward the center the average of all possible weights. In this case, the value of the center is “2.5”.

A simple method that conforms is to adjust to both ideas. All weighting values toward the center by 0.5 which is the half of the distance between adjacent values. Six possible weighting values and their adjusted values are depicted in Figure 3.4.

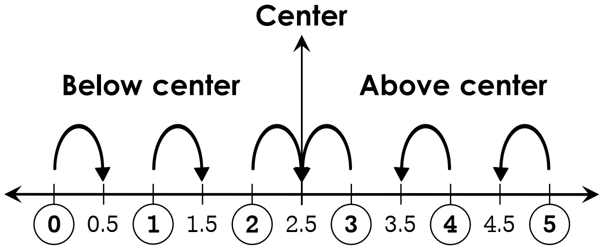


Figure 3.4 A method for adjusting weights (only values with “low confidence”)

There are two ways to change a weighting value, that is, increasing or decreasing it. If an expert gives a strong-sided weighting value with "N", s/he has low confidence in the strong value. Therefore such value should be decreased. For instance, if "5", which is a strong-sided weighting value, is given with "N", the value "5" is decreased and the adjusted value is calculated from the average of the value and the adjacent weight value, yielding $\frac{5+4}{2}=4.5$. Clearly, an expert who thinks that the relationship between a test item and a concept is less than strong "5" would not give "5" as a weighting value even with low confidence, Thus the value "5" should be decreased to a value not even less than "4". On the other hand, if an expert gives a weak-sided weighting value with low confidence, it is more likely that the true value is greater; for examples, if "0" is given as a weight with "N", the expert thinks that there may be some relationship between the concept and the test item. As a result, the value is increased and calculated in a similar way, yielding $\frac{0+1}{2}=0.5$, which is still less than "1".

To summarize this idea using mathematical notations, the input data for this approach are weighting values and confident degrees of n experts that show a relationship between a test item Q_j and a concept C_k; they are defined as $W_{Q_jC_k} = \{W_{Q_jC_k}(E_i) \mid i = 1 \text{ to } n\}$ and $CD_{Q_jC_k} = \{CD_{Q_jC_k}(E_i) \mid i = 1 \text{ to } n\}$ respectively, where $W_{Q_jC_k}(E_i) \in \{0, 1, 2, 3, 4, 5\}$ and $CD_{Q_jC_k}(E_i) \in \{\text{S, N}\}$. The weighting values ranging from 0 to 5 represent "no relationship", "very weak relationship", "weak relationship", "medium relationship", "strong relationship", and "very strong relationship" between a test item and a concept. The confident degree "S" means that an expert has high confidence in determining the association between a test item and a concept, whereas the confident degree "N" means that an expert has low confidence in determining such an association.

As mentioned before, all weighting values with low confidence are adjusted before being integrated by a majority-density algorithm in the next step. The adjusted weighting value for test item Q_j and concept C_k of expert E_i is denoted by $adjW_{Q_jC_k}(E_i)$. While adjusting the weighting values, we shall call the values that are equal to or less than 2 the "weak side" and those greater than or equal to 3 the "strong side". Two conditions for adjusting the weighting values are as follows:

Condition 1:

IF $W_{Q_jC_k}(E_i) \leq 2$ **AND** $CD_{Q_jC_k}(E_i)$ = "N"

THEN $adjW_{Q_jC_k}(E_i) = W_{Q_jC_k}(E_i) + 0.5$

Condition 1 is used for handling the case that an expert has assigned a weak-sided weighting value with low confidence. In this case, the value given by this expert is increased by 0.5. For example, if expert E_1 has assigned the association between test item Q_1 and concept C_2 as 1 with low confidence, this weighting value will be adjusted to 1.5.

Condition 2:

IF $W_{Q_jC_k}(E_i) \geq 3$ **AND** $CD_{Q_jC_k}(E_i)$ = "N"

THEN $adjW_{Q_jC_k}(E_i) = W_{Q_jC_k}(E_i) - 0.5$

Condition 2 is used for handling the case that an expert has assigned a strong-sided weighting value with low confidence. In this case, the value given by this expert is decreased by 0.5. For example, if expert E_2 has assigned the association between test item Q_2 and concept C_4 as 5 with low confidence, this weighting value will be adjusted to 4.5.

For a weighting value given by an expert with high confidence, that weighting value $W_{Q_jC_k}(E_i)$ is assigned to the value of $adjW_{Q_jC_k}(E_i)$. After adjusting all weighting values for test item Q_j and concept C_k, the set of those values of n experts is defined as $adjW_{Q_jC_k} = \{ adjW_{Q_jC_k}(E_i) \mid i = 1 \text{ to } n\}$.

In the next step, the basic idea of a majority-density algorithm to calculate an integrated value using all weighting values is presented. First we need to calculate the arithmetic mean, known as the central location of the data. In mathematics and statistics, it is a basic and useful measure of the central tendency of data. The mean is calculated from the summation of the values divided by the number of values via equation 1:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

where $\bar{x}$ is an arithmetic mean, x is the observation, and n is the number of data. Nevertheless, the distance from the mean is not robust enough as a measure of outliers, which are observations that are distant from other values of data. In this case, an outlier could result from several causes, i.e. insufficient knowledge, tiredness, or forgetfulness of an expert while giving a weighting value. Hence, it is necessary to remove it before calculating an integrated weighting value.

To improve the way of detecting an outlier, a clustering algorithm will be adapted to find dense area of weight values. Possible weighting values are {0, 1, 2, 3, 4, 5} which is a one dimensional scale. Therefore to find an outlier, we should start at the extreme values, the minimum and maximum values, which are assumed to be centroids of two clusters. The density at each is calculated in order to determine which one of the weight values is an outlier. Figure 3.5 displays the diagram showing the four steps of the algorithm.

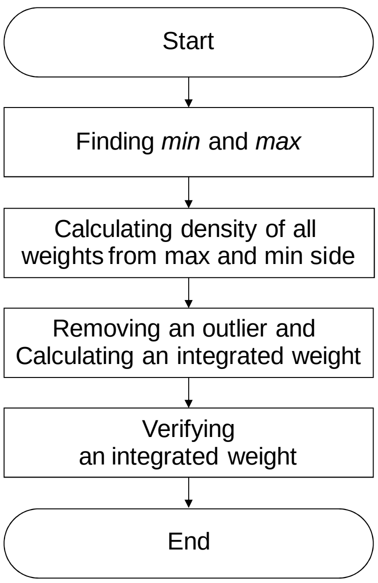


Figure 3.5 The diagram describing the four steps of a majority density algorithm

According to Figure 3.5, the algorithm consists of four steps: finding min and max, calculating density of all input weights from max and min sides, calculating an integrated weighting value not including an outlier, and verifying whether the integrated value should be used or all weighting values should be reconsidered.

Find min and max:

After a set of adjusted weighting values is obtained, the maximum and the minimum need to be found because an extreme value can be either the maximum or the minimum of all weighting values. For a set of weights *W*, equations that are used to find the maximum and the minimum are defined in equation 2 and equation 3 respectively:

$$\max(W) = \max_{w \in W} w \tag{2}$$

$$\min(W) = \min_{w \in W} w \tag{3}$$

For example, if five experts give “5”, “5”, “4”, “4”, and “1” as their weighting values, and all of them give their weighting values with high confidence (so all values need not be adjusted), max(*adjW*) will be “5” and min(*adjW*) will be “1”. These extreme values will be assumed as centers of all weights.

Calculating Density from Max and Min Sides:

In this step, based on a set of adjusted weighting values $adjW_{Q_jC_k}$, the density around the maximum adjusted weighting value $dnstMX_{Q_jC_k}$ and the density around the minimum adjusted weighting value $dnstMN_{Q_jC_k}$ will be calculated in order to define which one is an outlier as shown in equation 8 and equation 9:

$$\mathrm{sum}(W) = \sum_{w \in W} w \tag{4}$$

$$\text{avg}(W) = \frac{\text{sum}(W)}{|W|} \tag{5}$$

$$\text{vmad}(W, x) = \frac{\sum_{w \in W} |w - x|}{|W|} \tag{6}$$

$$\text{dnst}(W, x) = 1 - \frac{\text{vmad}(W, x)}{m} \tag{7}$$

$$dnstMX_{Q_j C_k} = \text{dnst}(adjW_{Q_j C_k}, \max(adjW_{Q_j C_k})) \tag{8}$$

$$dnstMN_{Q_j C_k} = \text{dnst}(adjW_{Q_j C_k}, \min(adjW_{Q_j C_k})) \tag{9}$$

where w is a weighting value, $|W|$ is the cardinality of a set W, sum(W) and avg(W) are the summation and average of all values in a set of weighting values, $\max(adjW_{Q_j C_k})$ and $\min(adjW_{Q_j C_k})$ are the maximum and minimum of the adjusted weighting values obtained from the previous step, m is the maximum rating scale (in this case, m=5). If *dnstMX* is greater than *dnstMN*, weighting values around the maximum value are denser than those around the minimum value and vice versa.

Removing an Outlier and Calculating an Integrated Weight:

The goal of this step is to remove an outlier by comparing *dnstMX* and *dnstMN*, and to calculate an integrated value using 2 steps:

Step 1: If *dnstMX* ≠ *dnstMN*, remove the outlier from $adjW_{Q_j C_k}$. The *outlier* is the maximum value if *dnstMX* is less than *dnstMN*. On the other hand, the *outlier* is the minimum value if *dnstMX* is more than *dnstMN*. If *dnstMX* = *dnstMN*, there is no *outlier*.

Step 2: Calculating an integrated weight *iWeight* using the arithmetic mean (see equation 5) to average all values in a set $adjW_{Q_j C_k}$

Verifying an Integrated Weight:

iWeight, calculated in the previous step, is only a temporary integrated weight and could not be used until the density of weighting values from majority is calculated to consider whether *iWeight* can be properly used as an integrated value or all experts will be asked to discuss, recheck, and reconsider their weighting values. The density of majority opinions $dmo_{Q_jC_k}$ will be defined by equation 10.

$$dmo_{Q_jC_k} = \mathrm{dnst}(adj_{Q_jC_k}, iWeight) \tag{10}$$

A threshold, θ, is the value that indicates the acceptable agreement level. In this case, the value of θ is assumed to be 0.85. If $dMO_{Q_jC_k} < \theta$, the experts will be asked to reconsider and discuss their weighting values; otherwise, the integrated weighting value will be used in any testing and diagnostic learning systems. On the other hand, *iWeight* can be properly used as an integrated weighting value if $dmo_{Q_jC_k}$ is equal to or more than θ because the opinions are similar.

3.1.3 An example of using the exploratory algorithm

To observe the effectiveness of the proposed algorithm to integrate weighting values from multiple experts, several input cases were used to calculate the integrated weighting values, and compared to that of the Panjaburee et al's approach (2010) in Table 3.1.

Table 3.1 The comparison of the results of integrating weighting values of the proposed method and Panjaburee et al's work (2010)

Cases	**Opinion from multiple experts**					**Integrated values**	
	E_1	E_2	E_3	E_4	E_5	**Previous work**	**Proposed method**
1	**1(S)**	4(N) =3.5	4(N) =3.5	4(S)	4(S)	1	3.75
2	**1(S)**	4(S)	4(S)	4(S)	4(S)	Reconsidering opinion	4
3	2(N) =2.5	3(N) =2.5	3(S)	4(N) =3.5	4(N) =3.5	Reconsidering opinion	3
4	**0(S)**	3(N)	3(S)	3(S)	3(S)	0	2.88

Cases	Opinion from multiple experts					Integrated values	
	***E*$_1$**	***E*$_2$**	***E*$_3$**	***E*$_4$**	***E*$_5$**	**Previous work**	**Proposed method**
		=2.5					
5	**1(S)**	1(N) =1.5	4(N) =3.5	4(N) =3.5	5(S)	Reconsidering opinion	Reconsidering opinion
6	**0(S)**	4(S)	4(S)	5(S)	5(S)	Reconsidering opinion	4.5
7	0(N) =0.5	1(S)	2(S)	2(S)	**5(S)**	5	1.38

According to Table 3.1, there were seven cases, and five experts gave their individual weighting values. First step, to prepare those values, the values with "low confidence" was adjusted based on the scale in Figure 3.4; for instance, in case 1, expert 2 and expert 3 gave "4" with "low confidence" which was adjusted to "3.5". Then, an outlier was detected and an integrated weighting value was calculated; the bold characters represent an outlier detected by the algorithm. In case 1, assuming that expert 1 gave the weighting value "1" with "high confidence" owing to misconception, the algorithm could detect "1" as an outlier and it was removed; the integrated value would result from the average of the reemaining weighting values, i.e., $\frac{3.5+3.5+4+4}{4}=3.75$, which was closer to the opinions of the majority than the result of the previous work. It implied that our approach could handle cases in which an opinion was distant from others.

Another interesting case was case 3 in which the distribution of all weights was on a medium level ranging from 2 to 4. There was no outlier and the proposed method provided "3" as the integrated value, while the result of the previous approach was "Reconsidering weight" generated from Rule 11. That rule needed only one value on the weak side, and one value from the strong side to generate "reconsidering weight"; however, it could be an unreasonable result because all opinions could be in the same area. This case showed the usefulness of the proposed algorithm which used density as one main factor to generate reasonable integrated weights. It implied that the proposed method could handle cases in which all weights were in a middle-valued dense area.

In case 5, an integrated weighting value of the proposed algorithm was "Reconsidering weight" although an outlier was removed. However, it was acceptable because all weighting values were not in a dense area and were divided into two sides. Moreover, the density, used to verify an integrated weight, was 0.775 which was less than 0.85. It implied that our approach could reasonably decide when to reconsider weights.

From the integrated weights of the seven cases in Table 3.1, results of the proposed method were different from those generated using the previous approach in almost all cases except case 5. Interestingly, in the proposed method, the density of weights was calculated, an outlier was also removed, and good-quality integrated weights were calculated from the average of remaining weights; it might diagnose students' learning problems more accurately, generate more accurate learning paths, and give more useful learning suggestions for students.

3.1.4 The finding and limitation of the algorithm

In the exploratory research, MATLAB 7.0.4 was used to reimplement Panjaburee et al.'s (2010) method to investigate the number of reconsidering weights from multiple experts. Moreover, it was used for comparing to preliminary results of integrating weighting values using the proposed method.

In the experiment, artificial data were generated as weighting values of multiple experts by randomizing values of weights and confidence degrees. All random data were uniformly distributed. The number of experts, the number of items, and the number of concepts were varied to observe their impacts. 1000 sets of data were generated for each variety. The results of the number of reconsidering weighting values from multiple experts are shown in Table 3.2.

Table 3.2 The comparison of results of integrating weighting values (the number of reconsidering weighting values)

Case	# of experts	# of items	# of concepts	# of reconsidering weights (%)	
				Panjaburee et al's method (2010)	Proposed method
1	3	30	10	40.58%	6.98%
2	4	30	10	58.17%	35.30%
3	5	30	10	70.01%	43.00%
4	6	30	10	77.77%	62.09%
5	7	30	10	83.15%	67.34%
6	8	30	10	86.95%	77.17%
7	9	30	10	90.11%	80.01%
8	10	30	10	92.14%	86.12%
9	3	40	10	43.6%	6.98%
10	3	50	10	43.7%	6.98%
11	3	60	10	43.6%	6.98%
12	3	30	15	43.6%	6.98%
13	3	30	20	43.5%	6.98%
14	3	30	25	43.6%	6.98%

As illustrated in Table 3.2, changing the number of experts, shown in case 1 to case 8, was a factor leading to different results. They demonstrated that for both methods, the percentage of reconsidering weights increased when the number of experts increased. However, the proposed algorithm outperformed the previous method in all cases with lower values of percentages; These results implied that majority consideration was an important factor directly resulting in lower numbers of reconsidering weighting values.

Although the proposed algorithm yielded better results compared to Panjaburee et al's method (2010) about integrated weighting values and the number of reconsidering cases, it still had some limitations when the number of experts was more than five because the number of reconsidering cases dramatically increased as shown in Figure 3.6.

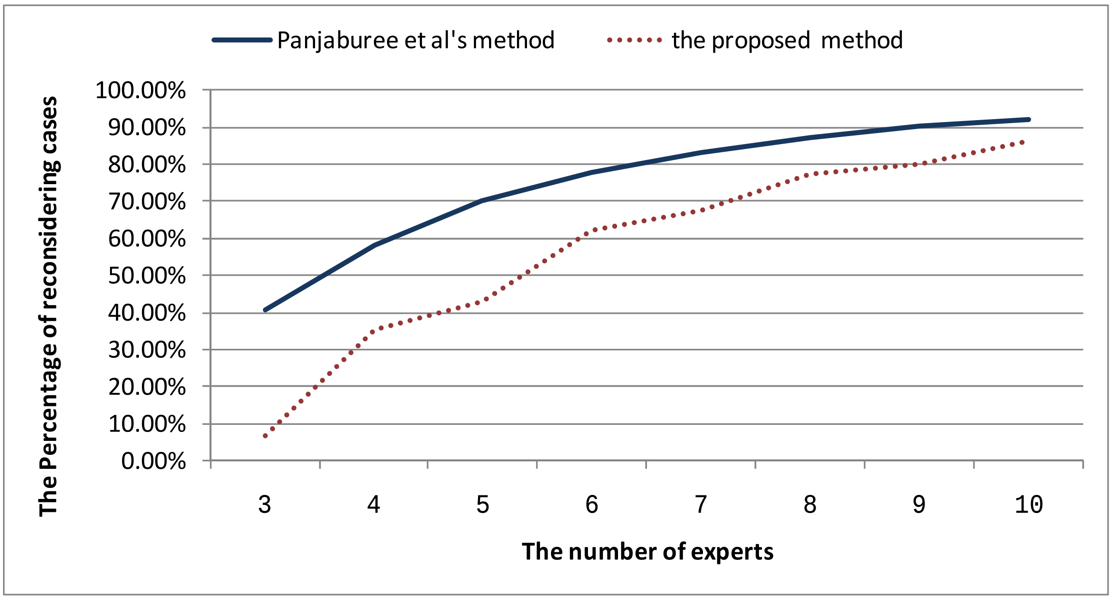


Figure 3.6 A line chart representing the percentage of reconsidering cases when varying the number of experts

The cause of the limitation resulted from the fact that the proposed algorithm could remove only one outlier; consequently, a lot of "reconsidering opinion" cases appeared. A good case for showing the drawback is that if the number of experts is seven, and each expert provides their weighting values for test item Q_2 concept C_2, and all values are adjusted to be:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7
$W_{Q_2C_2}$	0(S)	0(N)	5(N)	5(S)	5(S)	5(S)	5(S)
$adjW_{Q_2C_2}$	0	0.5	4.5	5	5	5	5

Based on the adjusted weighting values $adjW_{Q_2C_2}$, $\max(adjW_{Q_2C_2})$ is 5 and $\min(adjW_{Q_2C_2})$ is 0. Hence, the $dnstMX_{Q_2C_2}$ and $dnstMN_{Q_2C_2}$ values can be calculated as follows:

$$\text{sum}(adjW_{Q_2C_2}) = 0 + 0.5 + 4.5 + 5 + 5 + 5 + 5 = 25$$

$$\text{avg}(adjW_{Q_2C_2}) = \frac{25}{7} = 3.571$$

$$dnstMX_{Q_2C_2} = 1 - \frac{1.428}{5} = 0.714$$

$$dnstMN_{Q_2C_2} = 1 - \frac{3.571 - 0}{5} = 0.286$$

The value of $dnstMX_{Q_2C_2}$ is greater than that of $dnstMN_{Q_2C_2}$; that is, the majority opinion is closer to the maximum value 5 than to the minimum value 0. The latter defined as an *outlier*. After that, the integrated weighting value and the density of the majority opinion can be calculated excluding the outlier as follows:

$$iWeight = \frac{25 - 0}{7 - 1} = 4.16$$

$$dmo_{Q_2C_2} = 1 - \frac{1.645}{5} = 0.756$$

The density of the majority opinion $dmo_{Q_2C_2}$, 0.756, is less than the threshold θ, 0.85; it means that all weighting values from multiple experts will be reconsidered, rechecked and discussed again. The proposed algorithm can handle only cases in which there is one outlier; in this example, both 0 and 0.5 should be detected and removed as outliers.

Moreover, another limitation resulting from removing only one outlier is that it cannot provide a reasonable integrated weighting value. A good case for showing the drawback is that if the number of experts is eight and each expert provides their weighting value for test item Q_1 concept C_1 and all values are adjusted to be:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
$W_{Q_1C_1}$	1(N)	2(S)	5(N)	5(S)	5(S)	5(S)	5(S)	5(S)
$adjW_{Q_1C_1}$	1.5	2	4.5	5	5	5	5	5

Based on the adjusted weighting values $adjW_{Q_1C_1}$, $\max(adjW_{Q_1C_1})$ is 5 and $\min(adjW_{Q_1C_1})$ is 1.5. Hence, the $dnstMX_{Q_1C_1}$ and $dnstMN_{Q_1C_1}$ values can be calculated as follows:

$$\text{sum}(adjW_{Q_1C_1}) = 1.5 + 2 + 4.5 + 5 + 5 + 5 + 5 + 5 = 33$$

$$\text{avg}(adjW_{Q_1C_1}) = \frac{33}{8} = 4.125$$

$$dnstMX_{Q_1C_1} = 1 - \frac{0.875}{5} = 0.825$$

$$dnstMN_{Q_1C_1} = 1 - \frac{2.625}{5} = 0.475$$

The value of $dnstMX_{Q_1C_1}$ is greater than that of $dnstMN_{Q_1C_1}$; that is, the majority opinion of eight experts is closer to the maximum value 5 than to the minimum value 1.5 and the latter is defined as an *outlier*. After that, the integrated weighting value and the density of the majority opinion can be calculated excluding the outlier as follows:

$$iWeight = \frac{33 - 1.5}{8 - 1} = 4.5$$

$$dmo_{Q_1C_1} = 1 - \frac{0.714}{5} = 0.857$$

The density of the majority opinion $dmo_{Q_1C_1}$,0.857, exceeds the threshold θ, 0.85; however, the integrated weighting value $iWeight$ of 4.5 is not a reasonable result because the value does not reflect the majority value; that is “5”; in other words, there are five experts giving their opinions as “5”.

3.2 The majority density algorithm (enhanced version)

According to the drawback mentioned in section 3.1.4, it is necessary to improve the proposed algorithm to be able to remove outliers in the case that there is more than one outlier. A summary of the enhanced method is shown in Figure 3.7. In the first step, for each test item and each concept, weighting values with low confidence will be adjusted. Then, a potential outlier will be detected and it will be removed if necessary; this step is repeated until no potential outlier is detected or a potential outlier is not considered as a true outlier. Finally, an integrated weighting value will be calculated and verified whether it is reasonable or it needs reconsidering.

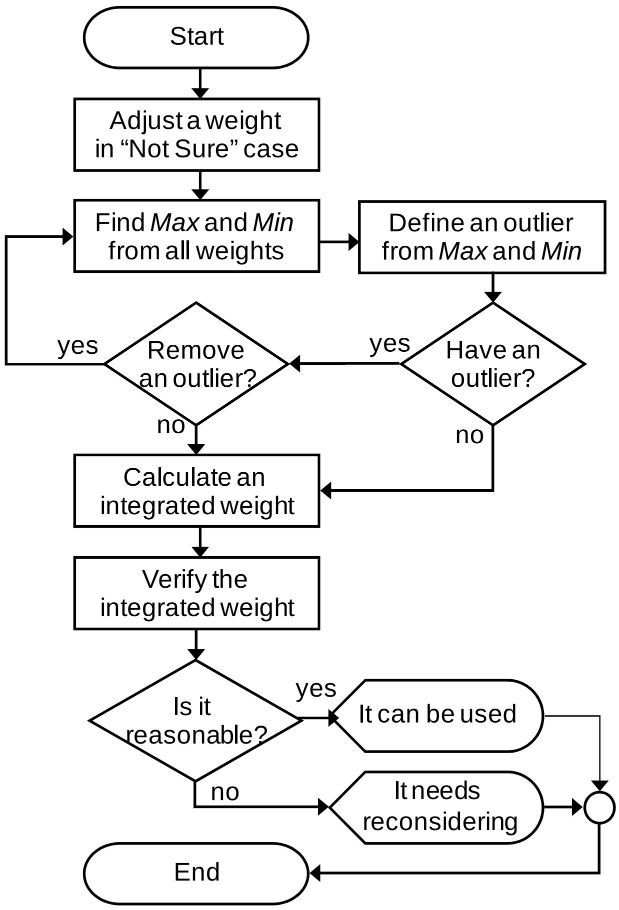


Figure 3.7 Flowchart of the enhanced majority density algorithm

The input of this approach is the same as that of the algorithm in section 3.1.

Step 1: Adjusting a weighting value with "low confidence"

All weighting values $W_{Q_jC_k}(E_i)$ with low confidence will be adjusted to be $adjW_{Q_jC_k}(E_i)$ by using the same criteria as that in exploratory version; i.e. if an expert gives a weighting value that is less than 3 with low confidence, it will be increased by 0.5; furthermore, if a given weighting value from an expert is more than or equal to 3 with low confidence, it will be decreased by 0.5. However, a weight with high confidence will not be adjusted anything.

Step 2: Defining a potential outlier

In this step, a potential outlier is detected by calculating the density level around the maximum and minimum values using equations 8 and 9. If $dnstMX_{Q_jC_k}$ is greater than $dnstMN_{Q_jC_k}$, the minimum value, $\min(adjW_{Q_jC_k})$, is considered a potential *outlier*, which represents an extreme weighting value among all weighting values for test item Q_j and concept C_k given by n experts. On the contrary, if $dnstMN_{Q_jC_k}$ is greater than $dnstMX_{Q_jC_k}$, the maximum value, $\max(adjW_{Q_jC_k})$ is considered a potential *outlier*. If $dnstMX_{Q_jC_k}$ is equal to $dnstMN_{Q_jC_k}$, there is no *outlier* among the weighting values. In the next step, we consider whether to remove the potential outlier from the set $adjW_{Q_jC_k}$. Step 3 is unnecessary in the case that there is no *outlier*; consequently, all weighting values will be used to calculate an integrated weighting value in Step 4.

Step 3: Verifying a potential outlier

Based on the potential outlier, this step is used for verifying whether it is distant enough from others and proper to be removed. $\mathrm{MAD}(W)$ represents the mean absolute deviation of all weighting values in a set *W*, which is defined by equation 14. In addition, the distance $distOW_{Q_jC_k}$ between the potential outlier and the average of the remaining adjusted weighting values for test item Q_j and concept C_k is defined by equation 12:

$$\mathrm{MAD}(W) = \frac{1}{|W|}\sum_{w \in W}|w - \mathrm{avg}(W)| \tag{11}$$

$$distOW_{Q_jC_k} = \left| outlier - \mathrm{avg}(adjW_{Q_jC_k} - \{outlier\}) \right| \tag{12}$$

where *outlier* represents the extreme value detected in Step 2. The condition for removing the outlier is:

IF $distOW_{Q_jC_k} > 1.25$ **AND** $distOW_{Q_jC_k} > \eta \times \mathrm{MAD}(adjW_{Q_jC_k} - \{outlier\})$

THEN $adjW_{Q_jC_k} = adjW_{Q_jC_k} - \{outlier\}$

REPEAT Step 2

ELSE **GOTO** Step 4

The value 1.25, which is one-fourth of the maximum rating scale *m* (*m* = 5), is used because any value within $\mathrm{avg}(adjW_{Q_jC_k} - \{outlier\}) \pm 1.25$ should not be considered a true outlier; ± 1.25 covers only a half of the scale. The first condition is used for preventing against false removal of the potential outlier by the second condition, which detects its relative distance from the rest of the weights. The symmetric triangular distribution is used for simulating the experts' behavior. Because sampling from the same population with smaller sample size yields smaller variation, this effect for the triangular distribution whose range is [0,1] is estimated by averaging 10,000 mean absolute deviations (MAD) of independent samples for each sample size *n*. Since a value that is more than 0.5 away from the average (of 0.5) will be outside the range of the triangular distribution, that value could be considered an outlier. That is, using MAD as the unit of measurement, a value which is more than 0.5/MAD away from the average could be considered an outlier. Thus, for each sample size n, we define η to be half the range of the triangular distribution divided by the average MAD. That is, $\eta \times \mathrm{MAD}(adjW_{Q_jC_k} - \{outlier\})$ is the critical value beyond which a weight could be considered a true outlier. If both conditions are met, the outlier will be removed from the set of adjusted weighting values $adjW_{Q_jC_k}$. Moreover, Step 2 will be repeated to detect other outliers. On the other hand, if the potential outlier is not suitable to be removed, all weighting values will be used to calculate an integrated weight in Step 4. Table 3.3 denotes the values of average MAD of the continuous triangular distribution and the values of η when n = 2 to 15.

Table 3.3 The average MAD and the value of η

Sample size	Average MAD (10,000 rounds)	η = (0.5/average MAD)
2	0.117	4.29
3	0.135	3.70
4	0.143	3.49
5	0.148	3.37
6	0.152	3.30
7	0.153	3.26
8	0.155	3.23
9	0.157	3.19
10	0.158	3.16
11	0.160	3.13
12	0.160	3.13
13	0.160	3.13
14	0.160	3.13
15	0.160	3.13

Step 4: Verifying an integrated weighting value

After the outliers were removed, the integrated adjusted weighting value for test item Q_j and concept C_k, $iWeight_{Q_jC_k}$, is calculated by equation 5.

To verify that the value of $iWeight_{Q_jC_k}$ is reasonable, the density of majority opinion $dMO_{Q_jC_k}$ is calculated as same as equation 10 of the algorithm in section 3.1.2.

The majority density algorithm in PHP script language is shown in Appendix F. The steps of the majority density algorithm are shown in a pseudocode as follows:

```
Algorithm: majority-density (input:weighting values from experts)
 1  Adjust weighting values with low confidence
 2  REPEAT
 3     Detect an outlier
 4     IF an outlier was detected
 5        Verify an outlier
 6        IF the value should be removed
 7           Remove it from other weighting values
 8        ELSE
 9           Skip to line 15
10        END IF
11     ELSE
12        Skip to line 15
13     END IF
14  UNTIL no outlier was not detected or was not removed
15  Calculate an integrated weight
16  Verify an integrated weight
17  IF the value is proper
18     Use it in testing and diagnostic system
19  ELSE
20     Ask experts to reconsider the values
21  END IF
```

3.3 Examples of using the majority density algorithm

The first example is the cases in which seven experts give their weighting values representing the relationship between concept C_3 and test item Q_3. The values and the degrees of confidence are assumed to be as follows:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7
$W_{Q_3C_3}, CD_{Q_3C_3}$	0,S	1,N	4,N	4,S	5,N	5,N	5,S

Step 1: Based on the weighting values obtained from seven experts, four experts, i.e., E_2, E_3, E_5 and E_6, have determined the weighting values for test item Q_3 and concept C_3 with low confidence; consequently, these weighting values are adjusted as follows:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7
$adjW_{Q_3C_3}$	0	1.5	3.5	4	4.5	4.5	5

Step 2: Based on the values of $adjW_{Q_3C_3}$, the maximum value, max($adjW_{Q_3C_3}$), is 5 and the minimum value, min($adjW_{Q_3C_3}$), is 0. Hence the density values of the max side and min side, $dnstMX_{Q_3C_3}$ and $dnstMN_{Q_3C_3}$, are evaluated as follows:

$$\text{avg}(\{0,1.5, 3.5, 4, 4.5, 4.5, 5\}) = 23/7=3.285$$

$$dnstMX_{Q_3C_3} = 1 - \frac{1.714}{5} = 0.66$$

$$dnstMN_{Q_3C_3} = 1 - \frac{3.285}{5} = 0.34$$

The density value of the max side, $dnstMX_{Q_3C_3}$, is more than that of the min side, $dnstMN_{Q_3C_3}$; that is, the majority opinion from seven experts is closer to the maximum value 5 than to the minimum value 0, and the latter is defined as a potential *outlier*. Therefore, the outlier 0 is verified in the next step.

Step 3: Based on the outlier 0, $distOW_{Q_3C_3}$ and MAD($adjW_{Q_3C_3} - \{outlier\}$) are calculated to verify whether it is distant enough from other weighting values and proper to be removed.

$$distOW_{Q_3C_3} = |0 - \text{avg}(\{1.5, 3.5, 4, 4.5, 4.5, 5\})| = 3.83$$

$$\eta \times \text{MAD}(\{1.5, 3.5, 4, 4.5, 4.5, 5\}) = 3.30 \times 0.89 = 2.93$$

Because $distOW_{Q_3C_3}$ is greater than 1.25 (3.83 > 1.25) and η × MAD($adjW_{Q_3C_3}$ $-\{outlier\}$) (3.83 > 2.93), the outlier 0 is removed from the set $adjW_{Q_3C_3}$. Therefore, Step 2 is used once again to detect another outlier.

Step 2 (the 2nd round): After removing the outlier, the set $adjW_{Q_3C_3}$ = {1.5, 3.5, 4, 4.5, 4.5, 5}. The maximum value, max($adjW_{Q_3C_3}$), is 5 and the minimum value, min($adjW_{Q_3C_3}$), is 1.5. Hence the density values of the max side and min side, $dnstMX_{Q_3C_3}$ and $dnstMN_{Q_3C_3}$, are evaluated as follows:

$$\text{avg}(\{1.5, 3.5, 4, 4.5, 4.5, 5\}) = 23/6$$

$$dnstMX_{Q_3C_3} = 1 - \frac{1.166}{5} = 0.77$$

$$dnstMN_{Q_3C_3} = 1 - \frac{2.333}{5} = 0.53$$

The density value from max side, $dnstMX_{Q_3C_3}$, is more than that from min side, $dnstMN_{Q_3C_3}$; that is, the majority opinion from six experts is closer to the maximum value 5, and the minimum value 1.5 is defined as a potential *outlier*. Therefore, the outlier 0 is verified in step3.

Step 3 (the 2nd round): Based on the outlier 1.5, $distOW_{Q_3C_3}$ and MAD($adjW_{Q_3C_3}$ $-\{outlier\}$) are calculated to verify whether it is distant enough from other weighting values and proper to be removed.

$$distOW_{Q_3C_3} = |1.5 - \text{avg}(\{3.5, 4, 4.5, 4.5, 5\})| = 2.80$$

$$\eta \times \text{MAD}(\{3.5, 4, 4.5, 4.5, 5\}) = 3.37 \times 0.44 = 1.48$$

Because $distOW_{Q_3C_3}$ is greater than 1.25 (2.80 > 1.25) and η × MAD($adjW_{Q_3C_3}$ $-\{outlier\}$) (2.80 > 1.48), the outlier 1.5 is removed from the set $adjW_{Q_3C_3}$. Therefore, Step 2 is used once again to detect another outlier.

Step 2 (the 3rd round): After removing the outlier, the set $adjW_{Q_3C_3}$ = {3.5, 4, 4.5, 4.5, 5}. The maximum value, max($adjW_{Q_3C_3}$), is 5 and the minimum value, min($adjW_{Q_3C_3}$), is 3.5. Consequently, the density values of the max side and min side, $dnstMX_{Q_3C_3}$ and $dnstMN_{Q_3C_3}$, are calculated as following:

$$\text{avg}(\{3.5, 4, 4.5, 4.5, 5\}) = 4.3$$

$$dnstMX_{Q_3C_3} = 1 - \frac{0.7}{5} = 0.86$$

$$dnstMN_{Q_3C_3} = 1 - \frac{0.8}{5} = 0.84$$

The density value of the max side, $dnstMX_{Q_3C_3}$, is more than that of the min side, $dnstMN_{Q_3C_3}$; that is, the majority opinion is closer to the maximum value 5 than to the minimum value 3.5, and the latter is defined as a potential *outlier*. Therefore, the outlier 3.5 is verified in step 3.

Step 3 (the 3rd round): Based on the outlier 3.5, $distOW_{Q_3C_3}$ and MAD($adjW_{Q_3C_3}$ $-\{outlier\}$) are calculated to verify whether it is distant enough from other weighting values and proper to be removed.

$$distOW_{Q_3C_3} = |3.5 - \text{avg}(\{4, 4.5, 4.5, 5\})| = 1$$

$$\eta \times \text{MAD}(\{4, 4.5, 4.5, 5\}) = 3.49 \times 0.25 = 0.87$$

Because $distOW_{Q_3C_3}$ is not greater than 1.25 (1 < 1.25)), the value 3.5 is not removed from the set $adjW_{Q_3C_3}$. The remaining weighting values will be used to calculate an integrated weighting value in Step 4.

Step 4: Because there is no outlier being removed from the previous step, the set of adjusted weighting values $adjW_{Q_3C_3}$ still is {3.5, 4, 4.5, 4.5, 5}. Therefore, the integrated weighting value for test item Q_3 and concept C_3 is calculated as follows:

$$iWeight_{Q_3C_3} = \text{avg}(\{3.5, 4, 4.5, 4.5, 5\}) = 4.3$$

The value of $dMO_{Q_3C_3}$ is calculated to verify that the value of $iWeight_{Q_3C_3}$ is reasonable as follows:

$$dMO_{Q_3C_3} = 1 - \frac{0.44}{5} = 0.91$$

The value of $dMO_{Q_3C_3}$ is greater than the threshold value, 0.85; that is, the experts have agreed on the value 4.3 as the integrated weighting value.

From a set of weighting values given by seven experts for test item Q_3 and concept C_3, it should be noted that the integrated weighting value "0" for test item Q_3 and concept C_3 using Rule 9A of Panjaburee et al's work and "experts need to reconsider their weighting value" using Rule 7 are unreliable because there are two rules, "Rule 7" and "Rule 9A", that can be chosen for handling the set of weighting values. Interestingly, the integrated weighting value "0" calculated by "Rule 9A" is on the weak side and almost all weighting values are on the strong side. That is, one gets an unreliable integrated weighting value while using the Panjaburee et al.'s rules because their rules do not consider the majority opinion. The integrated weighting value "4.3" calculated using the majority density algorithm is clearly more reliable, because it considers the majority opinion for giving the integrated weighting value. Therefore, it can overcome the drawbacks of the set of rules and can provide better integrated weighting values for diagnosing learning problems in testing and diagnostic learning systems based on the CER model.

Another example is the case in which eight experts give their weighting values representing the relationship between concept C_4 and test item Q_4. The values and the degrees of confidence are assumed to be as follows:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
$W_{Q_4C_4}, CD_{Q_4C_4}$	0,S	0,S	0,N	1,S	5,N	5,S	5,S	5,S

Step 1: Based on the weighting values obtained from eight experts, two experts, i.e., E_3 and E_5, have determined the weighting values for test item Q_4 and concept C_4 with low confidence; consequently, these weighting values are adjusted as follows:

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
$adjW_{Q_4C_4}$	0	0	0.5	1	4.5	5	5	5

Step 2: Based on the values of $adjW_{Q_4C_4}$, the maximum value, max($adjW_{Q_4C_4}$), is 5 and the minimum value, min($adjW_{Q_4C_4}$), is 0. Hence the density values of the max side and min side, $dnstMX_{Q_4C_4}$ and $dnstMN_{Q_4C_4}$, are evaluated as follows:

$$\text{avg}(\{0, 0, 0.5, 1, 4.5, 5, 5, 5\})=21/8=2.625$$

$$dnstMX_{Q_4C_4} = 1 - \frac{2.375}{5} = 0.53$$

$$dnstMN_{Q_4C_4} = 1 - \frac{2.625}{5} = 0.48$$

The density value of the max side, $dnstMX_{Q_4C_4}$, is more than that of the min side, $dnstMN_{Q_4C_4}$; that is, the majority is closer to the maximum value 5 than to the minimum value 0, and the latter is defined as a potential *outlier*. Therefore, the outlier 0 is verified in the next step.

Step 3: Based on the outlier 0, $distOW_{Q_4C_4}$ and MAD($adjW_{Q_4C_4}$-{*outlier*}) are calculated to verify whether it is distant enough from other weighting values and proper to be removed as follows:

$$distOW_{Q_4C_4} = |0 - \text{avg}(\{0, 0.5, 1, 4.5, 5, 5, 5\})| = 3$$

$$\eta \times \text{MAD}(\{0, 0.5, 1, 4.5, 5, 5, 5\}) = 3.26 \times 2.14 = 6.99$$

Because $distOW_{Q_4C_4}$ is greater than 1.25 (3 > 1.25), but it is still less than η × MAD($adjW_{Q_4C_4} - \{outlier\}$) (3 < 6.99), the outlier 0 is not removed from the set $adjW_{Q_4C_4}$. Therefore, all weighting values in the set will be used for calculating an integrated weighting value in step 4.

Step 4: Because there is no outlier being removed from the previous step, the set of adjusted weighting values $adjW_{Q_4C_4}$ still is $\{0, 0, 0.5, 1, 4.5, 5, 5, 5\}$. Therefore, the integrated weighting value for test item Q_4 and concept C_4 is calculated as follows:

$$iWeight_{Q_4C_4} = \text{avg}(\{0, 0, 0.5, 1, 4.5, 5, 5, 5\}) = 2.63$$

The value of $dMO_{Q_4C_4}$ is calculated to verify that the value of $iWeight_{Q_4C_4}$ is reasonable as follows:

$$dMO_{Q_4C_4} = 1 - \frac{2.142}{5} = 0.55$$

The value of $dMO_{Q_3C_3}$ is less than the threshold value, 0.85; that is, the experts need to check, discuss, and reconsider their weighting value.

For test item Q_4 and concept C_4, there are three different integrated weighting values using Panjaburee et al.'s method, i.e. "5" using Rule 8B, "0" using Rule 9B, and "experts need to reconsider their value" using Rule 7, which are unreliable because more than one rule can be used for handling the set of weighting values. Because the weighting values of multiple experts can be separated into two groups, one on the weak side and the other on the strong side, clearly, the result "experts need to reconsider their value" determined by the majority density algorithm is reasonable.

3.4 Developing testing and diagnostic learning system

In this section, an online full-scaled testing and diagnostic learning system based on the majority density algorithm was developed using these following tools:

(1) Apache or Apache HTTP server: an open source web server software that can run on UNIX and Windows servers

(2) PHP: a programming language that can be embedded as part of hypertext markup language (HTML) to provide dynamic web pages. It is a server-side language and is interpreted by a web servers.

(3) MySQL: a free and one of the most wildly used relational database management systems (RDBMS). It can also support Structure Query Language (SQL)

(4) phpMyAdmin: a user-friendly tool developed by PHP programming language to handle the administration of MySQL

The system was developed as an online system and was composed of two important parts: the first part for teachers or experts was Knowledge Elicitation and Integration System for determining the weights of Concepts (KEISC) and the other part for students was Testing and Diagnostic Learning Problem system (TDLP). An overview of the system comprising the two parts was shown in Figure 3.8. According to several components of the system, the CER model was the first thing that had to be constructed by teacher(s) or expert(s) to show related concepts and the relationship among them. In this research work, the CER model consisted of nine concepts in a computer programming course as depicted in Figure 3.9. This model was used as a tool to diagnose learning problems of students later.

Based on a developed CER model, a test sheet in the topic "computer programming" for analyzing learning problems of students was constructed and shown in Appendix B. This test consisted of 30 test items, each comprising four choices. Both the test sheet and the CER model were used in TDLP and KEISC by students and experts.

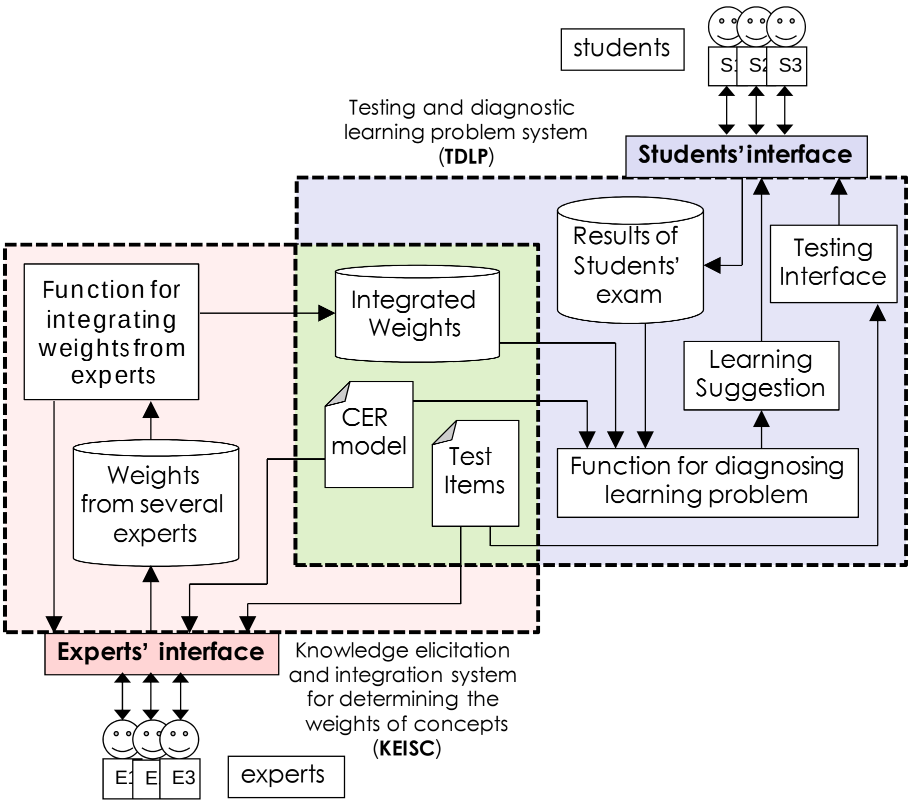


Figure 3.8 An overview of a testing and diagnostic learning system supporting multiple experts

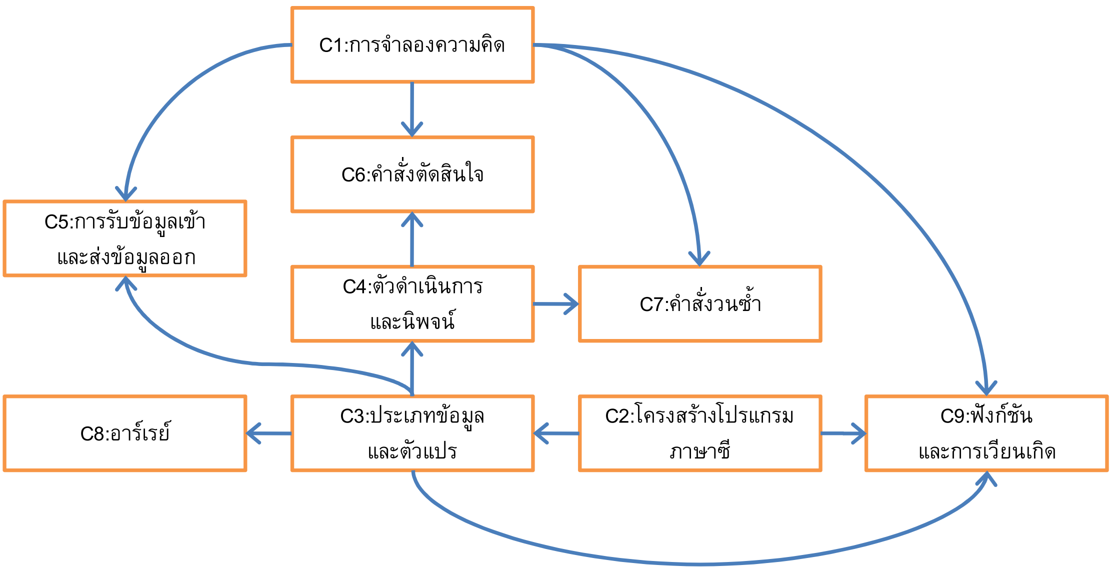


Figure 3.9 The CER model for computer programming course

3.4.1 Knowledge Elicitation and Integration System for determining the weights of Concepts (KEISC)

KEISC was used by multiple experts and could be accessed via the URL: http://www.il.mahidol.ac.th/keisc/. All experts were requested to log on before using it. The screen for logging in was shown in Figure 3.10.

Figure 3.10 The login page of KEISC

After that, experts would see the main screen of KEISC consisting of three important areas as shown in Figure 3.11.

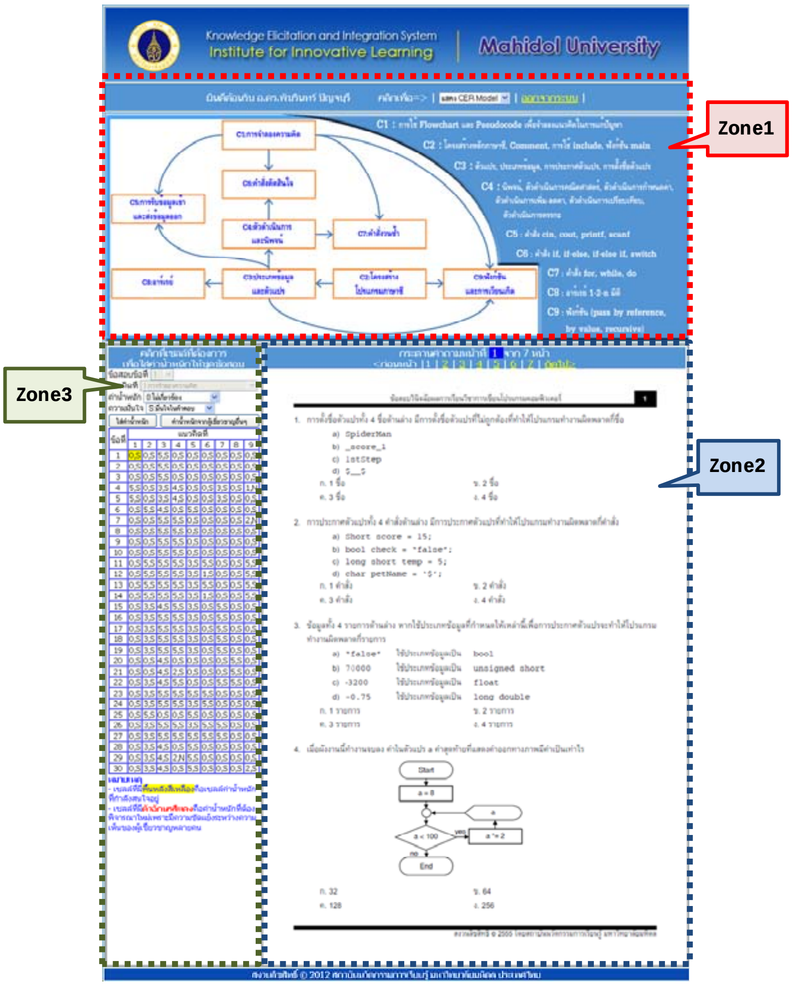


Figure 3.11 Main screen of KEISC

The first area, zone 1, showed the name of the expert who connected to the system. There was two components i.e., the combo box used to show/hide the CER model and the descriptions of the nine concepts and the hyperlink used to log off of the system. This part of KEISC is depicted in Figure 3.12.

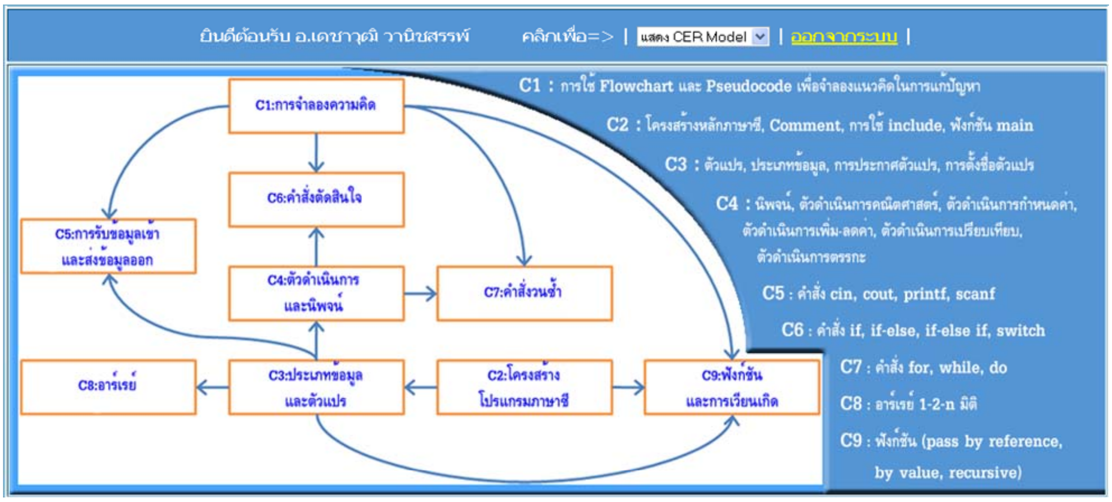


Figure 3.12 The first area of KEISC

Another area was zone 2 used for viewing test items spread over several pages. There were several hyperlinks to the pages of the test sheet. The hyperlinks are depicted in Figure 3.13.

Figure 3.13 The navigator to change page of a test sheet

The last and most important area is shown in Figure 3.14. It was used for managing weighting values by an expert, who determined an opinion representing the relationship of each concept and each test item. After reading a test item, the expert considered which concepts between the CER model related to the item and provided a value ranging from 0 to 5, representing “no relationship”, “very weak relationship”, “weak relationship”, “medium relationship”, “strong relationship”, and “very strong relationship”; in addition, that value was given together with the confidence degree, i.e., “1” for high confidence and “0” for “low confidence”, to form a weighting value.

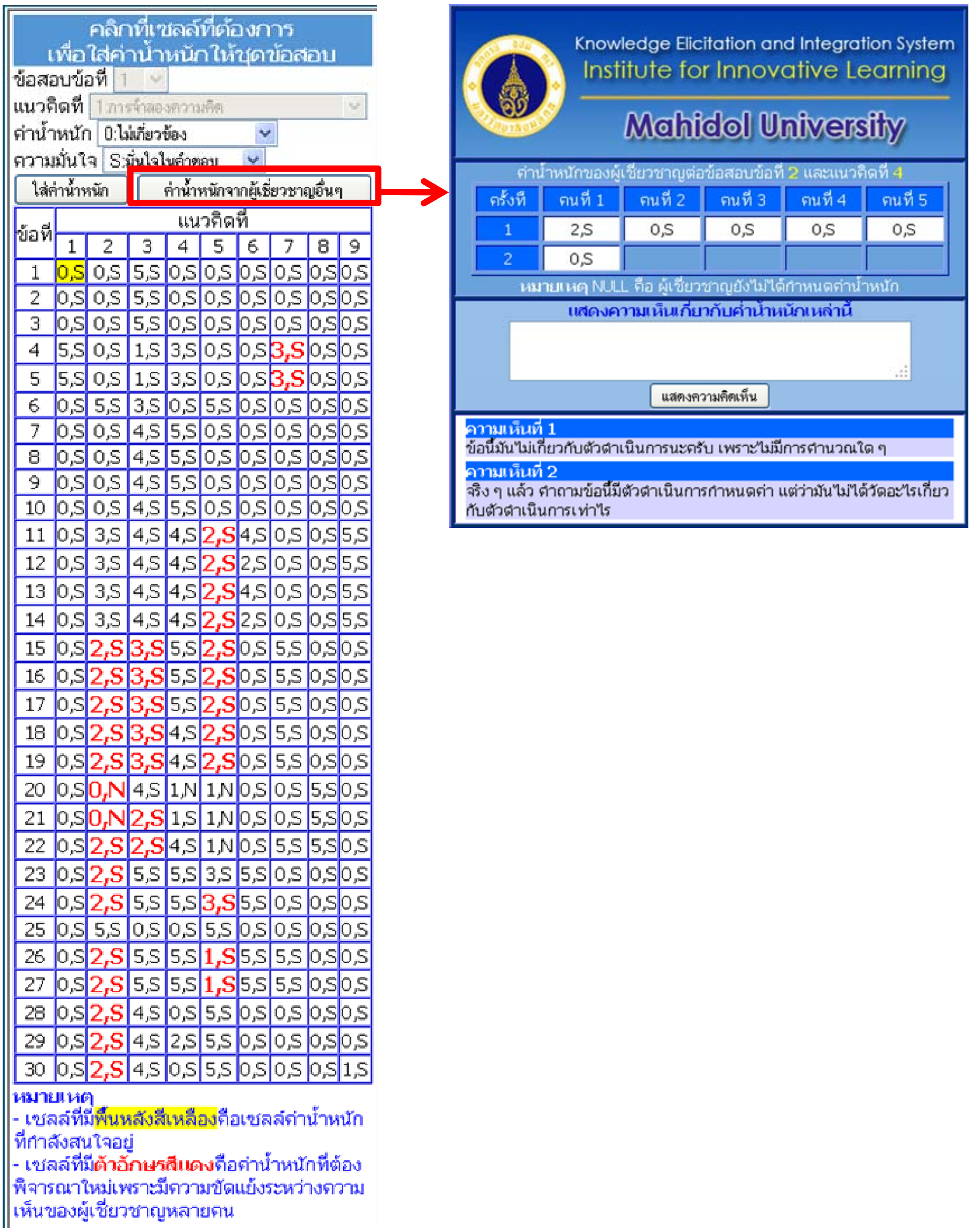


Figure 3.14 The area for assigning opinions and the discussion box

An expert could choose an item and a concept so as to assign those a value by clicking any cell in the table and whose background color would be changed to yellow. Then two combo boxes were used for choosing a weighting value and confidence degree. After that, the left command button, under the two combo boxes, was used for putting the chosen values into the chosen cell. After obtaining the weighting values from all the experts, the values were integrated using the majority density algorithm (see details in Section 3.2). If a reconsidering case appeared, the experts had to recheck and reconsider their ratings again; the system would warn by changing the background colors in the cells of the table in this area to be red. To easily solve these cases, an expert could observe others' weighting values and confidence degrees, and could discuss with others about the reasons behind one's own weight by clicking the right command button, adjacent to the aforementioned button for putting weighting values, in order to open the discussion box. The goal of KEISC was to produce reliable integrated weighting values that will be used as one of the important inputs of TDLP.

3.4.2 Testing and Diagnostic Learning Problem (TDLP)

On the other hand, TDLP was used to diagnose students' learning problems and give helpful suggestions to them. All students were requested to log on to do a diagnostic test as shown in Figure 3.15. After logging on, a diagnostic test sheet would be provided for students as depicted in Figure 3.16. In this web page, students could read questions of the test sheet and use the combo box to change the pages of the test sheet. After that, they could select the answers using radio buttons. After completing the test sheet, the test results of the student were displayed as in Figure 3.17.

Figure 3.15 The login page of TDLP

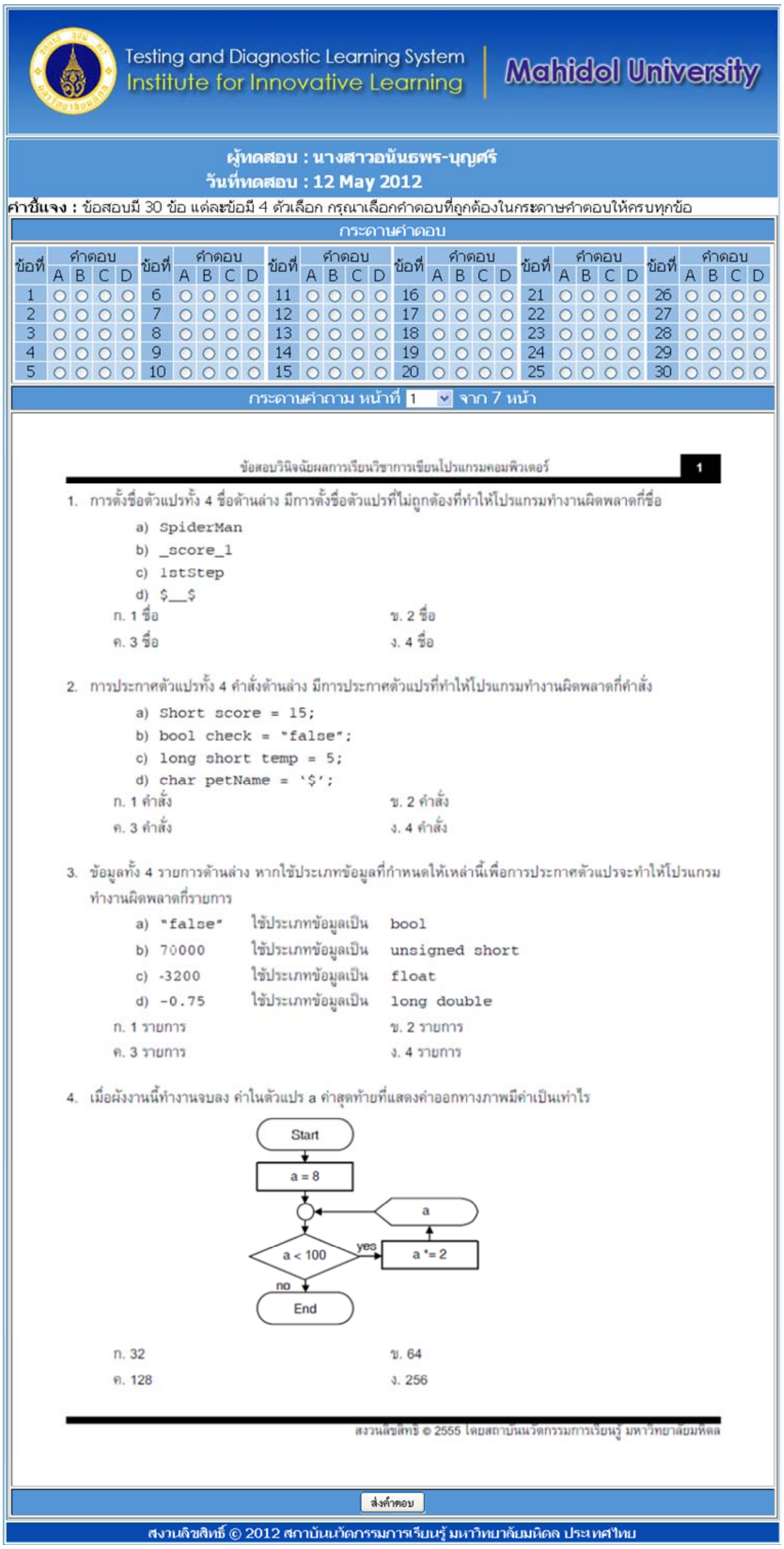


Figure 3.16 Test module and a diagnostic test sheet

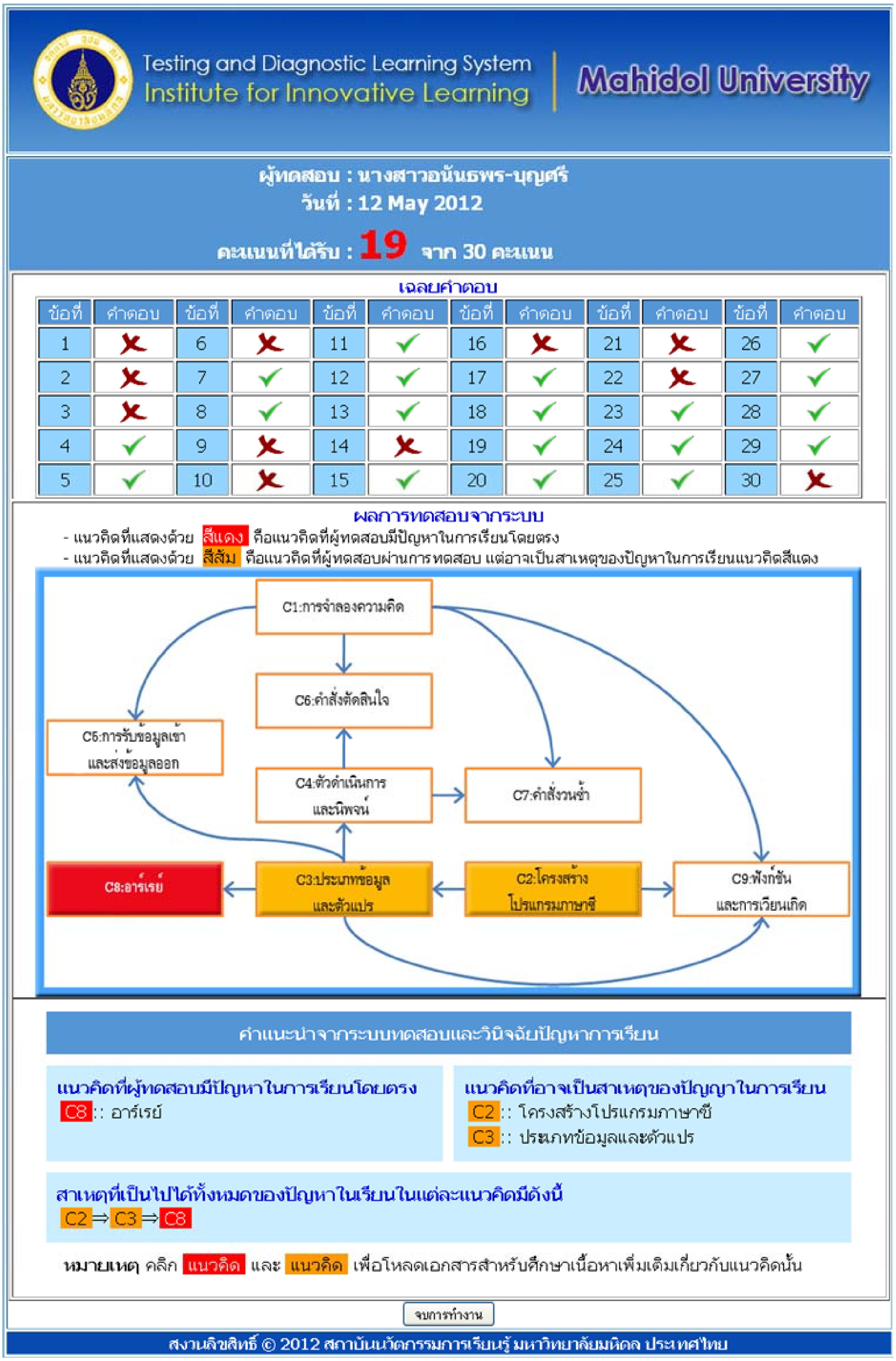


Figure 3.17 Test results and suggestion from TDLP

According to Figure 3.17, total test score would be shown. The most important result in this web page was learning problems and learning paths of students. TDLP would analyze students' answer and provide leaning guidance based-on the concept-effect relationship (CER) model. Students could easily focus on concepts that they failed and could know a real cause of learning problems so as to improve their learning achievements. A concept in red color represented concept that a student directly failed and might directly impact to learning problem. A concept in orange color represented concept that a student indirectly failed but it contained a fundamental knowledge of fail items. A concept that a student has sufficient knowledge was shown in white color. For every fail concepts, shown in red and orange, a student could download a supplementary sheet for improving knowledge by clicking the concept.

3.5 Experimental design of the main study

To evaluate the performance of this novel approach, an experiment was conducted to investigate the following research questions:

(1) To what extent is the testing and diagnostic learning system using the majority density approach helpful to the students in improving their conceptual learning outcome?

(2) Is the testing and diagnostic learning system using a majority density approach more effective in improving students' achievement compared to the system using Panjaburee et al. (2010)'s approach?

(3) To what extent can the testing and diagnostic learning system using the majority density approach reduce conflicting cases and the number of reconsidering cases compared with the system using Panjaburee et al. (2010)'s approach?

(4) How do students feel after receiving learning guidances from the testing and diagnostic learning system?

The domain experts in a computer programming course were five instructors from Rambhai Barni Rajabhat University and Mahidol University, Thailand. All of them hold at least a master's degree in computer science (CS), information technology (IT), or in related fields.

3.5.1 Measuring tools

The instruments which were used for measuring the learning outcomes of students were conceptual tests (pre- and post-tests). Both of them were developed and verified by the domain experts and were composed of thirty multiple-choice questions relating to nine concepts of "basic computer programming". The function of the pre-test was to ensure that participants in the control and experimental groups had equal prior knowledge. This test was also used as the diagnostic test for detecting students' conceptual learning problems. On the other hand, the post-test was used to compare the conceptual learning outcome of students in control group to that of students in the experimental group after receiving conceptual learning guidance from the systems.

Furthermore, a questionnaire was employed to investigate students' satisfaction after receiving the conceptual learning suggestions from the testing and diagnostic learning system. The questionnaire used a 5-point Likert scale ranging from 1 to 5, which indicated "strongly disagree", "disagree", "neutral", "agree", and "strongly agree" respectively. The questionnaire is shown in Appendix E.

3.5.2 Participants

122 undergraduate students in this research were from Rambhai Barni Rajabhat University, consisting of 28 males and 84 females. Among these students, the average age was 19 years old. All of them were first-year or second-year undergraduate students in various programs related to computer such as mathematics, computer business, and computer education. They had already studied the "Computer Programming I" course.

3.5.3 Treatments

The experimental design was depicted in Figure 3.18. All students were evaluated for their basic knowledge using the pre-test with a 120-minute time limit. The test was also used as the diagnostic test in the testing and diagnostic learning systems. Next, all of them were randomly separated into control (56 students) and experimental groups (56 students). The different treatments between the two groups were the learning suggestions generated from the testing and diagnostic systems. The students in the *experimental group* would use the testing and diagnostic system employing the majority density approach to integrate opinions and receive learning suggestions from it, while the students in the *control group* received learning suggestions generated from the system using Panjaburee et al. (2010)'s set of rules to integrate weighting values. After students in both groups knew their conceptual learning problems, they needed to practice using the learning materials following the suggestions from the testing and diagnostic systems within 15 days, and they would be evaluated again by taking the post-test with the same time limit. The experiment was conducted in April, 2012.

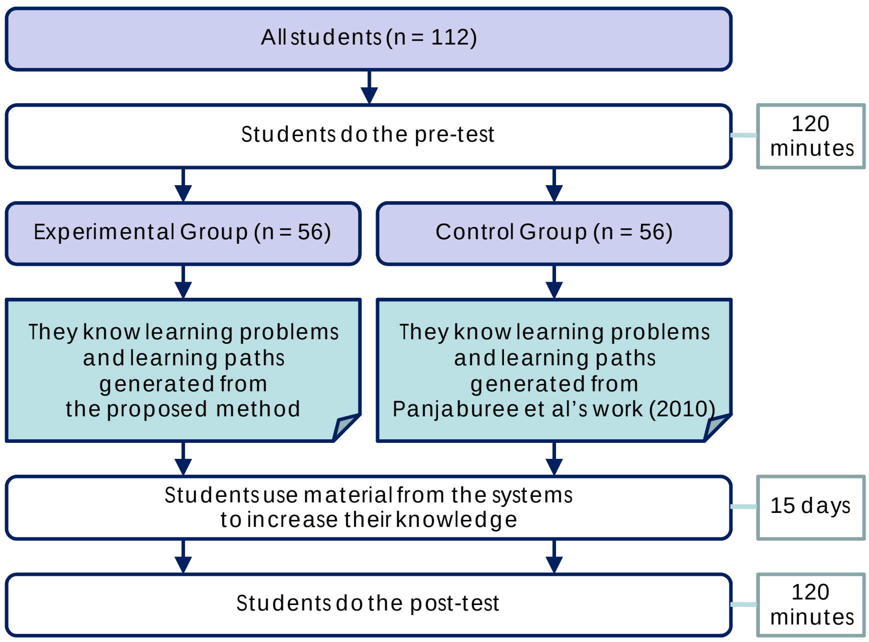


Figure 3.18 Experiment procedure

In this chapter, the majority density algorithm for improving the effectiveness of the knowledge-integration method with the cooperation of multiple experts was described in deep detail. A testing and diagnostic system was also implemented based on the proposed approach. Moreover, an experiment was conducted to compare the performance of the proposed approach and the multi-expert approach of Panjaburee et al. (2010). The experimental results are depicted in the next chapter.

CHAPTER IV
RESULTS AND DISCUSSION

In this chapter, the usefulness of the testing and diagnostic learning system based on the majority density algorithm was evaluated through the conceptual learning outcome and satisfaction of students using several instruments, i.e., the pre-test, post-test, and questionnaire. In addition, the discussion of the experimental results was mentioned in this chapter. The guideline for students and teachers to effectively use this system was also presented.

4.1 Effect of treatment

To evaluate students' prior knowledge in both the experimental and control groups, the pre-test was used for verifying that their basic knowledge was equivalent before receiving services from the testing and diagnostic learning systems in the "computer programming course". The average pre-test score of the control group was 10.16 (SD = 8.65), and the average pre-test score of the experimental group was 10.5 (S.D = 11.35). The F-test was applied to examine whether the variances between the two groups were equal. The result of this test was not significant ($p = .158 > .05$), which indicated that variances of students' pre-test score in both groups were statistically indifferent. Therefore, the independent sample t-test assuming equal variances was applied to examine the difference between the average pre-test scores of the students. The result of analyzing the pre-test revealed that there was no statistically significant difference ($p=.285 > .05$) between the average scores of both groups. Therefore we could claim that prior knowledge of students in the two groups was equal.

After finishing the learning activity, the average post-test score of the control group was 15.27 (SD = 21.47), and the average post-test score of the experimental group was 17.09 (SD = 22.67). The post-test results were analyzed by the independent sample *t*-test. First, the *F*-test was necessary to be used for examining whether the variances between the two groups were equal. The result of the *F*-test showed that there was no difference between the variances of both groups (p=.420 > .05). After that, the independent sample *t*-test assuming equal variances was applied to examine the difference between the average post-test scores of the students in the control and experimental groups. The result of analyzing the post-test (p=.02 < .05) showed that students in the experimental group could improve the learning achievement more than those in the control group. Consequently, it could be concluded that the testing and diagnostic learning system using the majority density approach to provide learning suggestions and learning materials could be more helpful for students than the system using Panjaburee et al (2010)'s approach (see Table 4.1).

Table 4.1 Mean and standard deviation of students in control and experimental groups for the pre-test and post test results focusing on the effect of treatments

		Group		
		Control (N=56)	Experimental (N=56)	*t*
Pre-test	Mean	10.16	10.5	-0.5678
	SD	2.156	3.368	
Post-test	Mean	15.27	17.09	-2.0516*
	SD	4.633	4.761	

*p < .05

4.2 Conceptual learning score

Table 4.2 represents the difference between the pre- and post-test results of students in the control and experimental groups. The paired sample *t*-test was applied to examine difference between the pre- and post-test scores of the students in both groups. After receiving learning suggestions from the system, for the control group, the result of using the *t*-test (p=4.737×10^{-13} < .05) showed that students could significantly improve the conceptual learning outcome. For the experimental group, the result of using the *t*-test (p=4.763×10^{-15} < .05) also showed that students in this group could gain more knowledge and improve the learning outcome after receiving learning suggestions from the system. Consequently, it could be concluded that both testing and diagnostic learning systems were useful and helpful for students to improve their knowledge in the computer programming course.

Table 4.2 Mean and standard deviation of students in control and experimental groups for the pre-test and post test results focusing on the learning improvement of students

Group	Pre-test		Post-test		Learning improvement	*t*
	Mean	SD	Mean	SD		
Experimental (N=56)	10.5	3.368	17.09	4.761	+6.59	-10.497*
Control (N = 56)	10.16	2.941	15.27	4.633	+5.11	-9.213*

*p < .05

4.3 Learning outcome vs. knowledge level

To further investigate the effect of the treatment, the students in both the control and experimental groups were separated based on their knowledge levels into three levels, that is, high-achieving, medium-achieving and low-achieving students. The ratio of students in each group was approximately 30%:40%:30%, and the criteria to group them was depicted in Table 4.3.

Table 4.3 The criteria to classify students into three categories

Treatments	Range of pre-test score	Cluster	# of students in each group
Control	0-8	low-achievement	14
	9-11	medium- achievement	25
	12-30	high- achievement	17
Experiment	0-8	low- achievement	17
	9-11	medium- achievement	24
	12-30	high- achievement	15

Table 4.4 Comparison of the post-test scores of control and experimental groups in different knowledge levels

Cluster	Group	N	Mean	SD	*t*
High-achievement	(a)	17	18.82	5.757	-0.175
	(b)	14	19.14	4.054	
Medium-achievement	(a)	24	16.79	4.88	1.814*
	(b)	25	14.4	4.339	
Low-achievement	(a)	15	15.6	2.414	1.988*
	(b)	17	13.35	3.74	

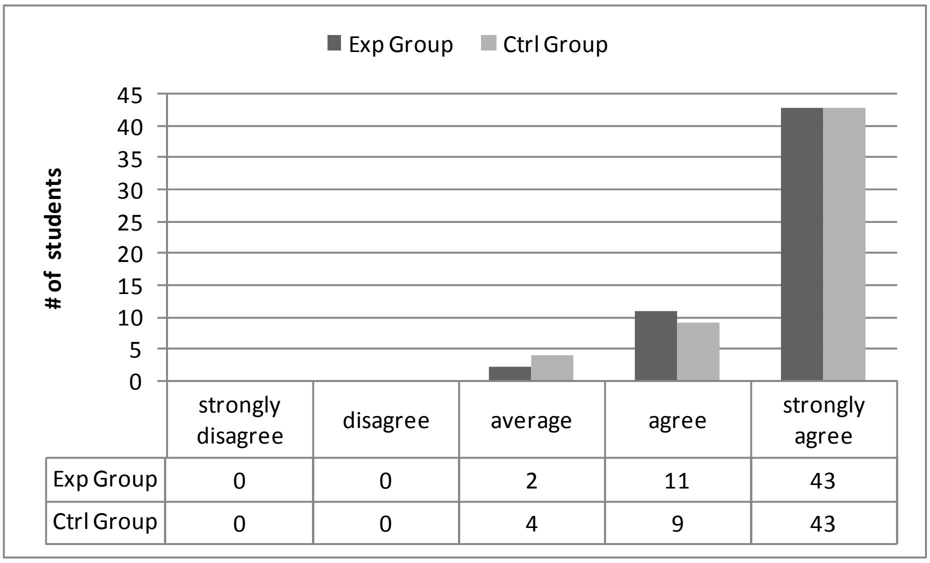
(a) Experimental group and (b) Control group

Table 4.4 represents the comparison between the post-test scores of students with different knowledge levels in the control and experimental groups. Students in all clusters in both groups had an improvement in their learning outcomes which reflected that the testing and diagnostic learning system could be useful and could provide helpful suggestions for students to improve their knowledge and to know the real cause of their conceptual learning problems.

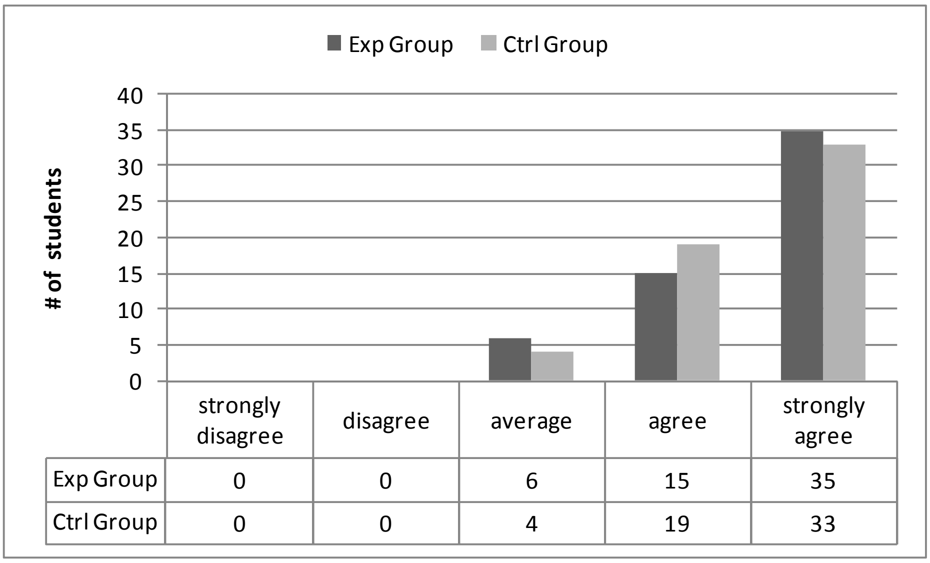
Interestingly, students in low-achieving and medium-achieving clusters in the experimental group, who used the testing and diagnostic learning system based on the majority density algorithm, had significantly higher post-test scores than those with the same knowledge levels in the control group did. Especially for low-achieving students, teachers played an important role and directly impacted on their learning achievement (Roueche, Baker, & Roueche, 1984; Sanders, Wright, & Horn, 1997). In this case, a tutoring system or learning center could substitute for teachers to improve the knowledge of low-achieving students (Alfred & Lum, 1988), and the system that could provide accurate suggestions to students was so helpful in improving their knowledge. These results implied that the testing and diagnostic learning system based on the majority density algorithm could accurately diagnose conceptual learning problems and provide better suggestions for students especially in low- and medium-achievement clusters. These results supported previous works showing that the testing and diagnostic learning system was a useful tool for low-achieving students compared to high-achieving students (Chen, 2011). On the other hand, for students in the high-achievement cluster, although the learning achievement of students in the experimental group was slightly lower than that of those in the control group, the difference did not reach statistical significance.

4.4 Perceptions toward learning guidance

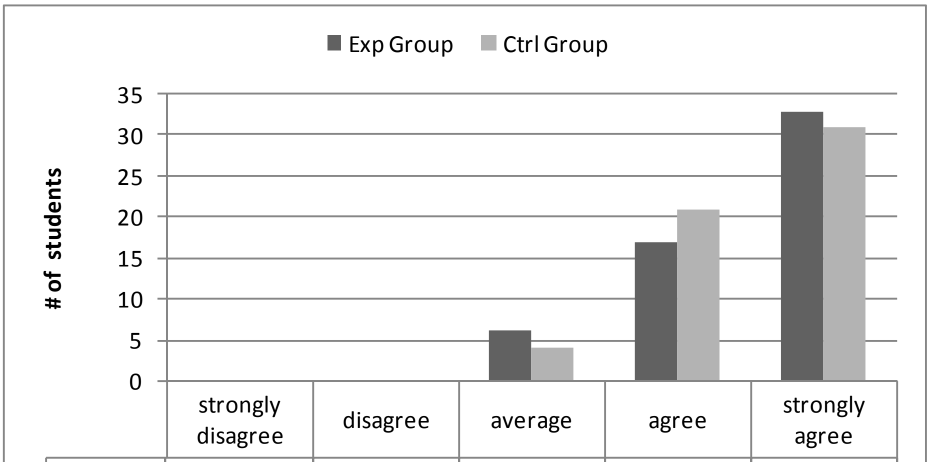
According to Figures 4.1(a) to 4.1(i), the responses of students in both the control and experimental group to the questionnaire for investigating their satisfaction with the testing and diagnostic learning systems were depicted in bar graphs. The majority of the students in the control group rated the satisfaction level as “strongly agree” for all the items of the questionnaire, they seemed satisfied with the learning suggestions provided by the system, and thought that the suggestions were useful and helpful in improving their conceptual learning outcome, understanding and learning confidence. They felt that their learning confidence increased and they could know the real causes of the learning problems for this topic. In addition, they would recommend using the system to other students. Similarly, the majority of the students in the experimental group also rated the satisfaction level as “strongly agree” for all the items of the questionnaire, they seemed satisfied with learning suggestions provided by the system, and thought that the suggestions were useful and helpful in improving their learning outcome, understanding and learning confidence. They felt that their learning confidence increased and they could know the real causes of the learning problems for this topic. Moreover, they would recommend using the system to other students.



(a) Q1: I am satisfied with the conceptual learning suggestions provided by the TDLP

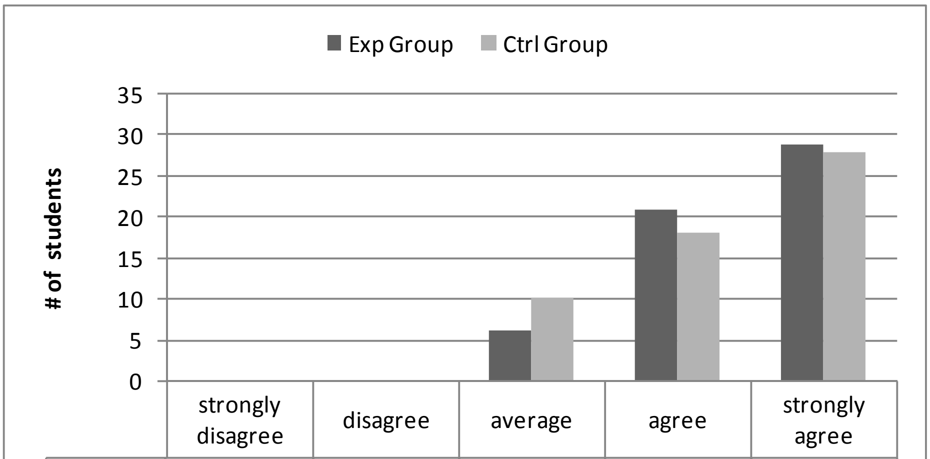


(b) Q2: I agree that the conceptual learning suggestions provided by the TDLP are useful to my learning in the computer programming course



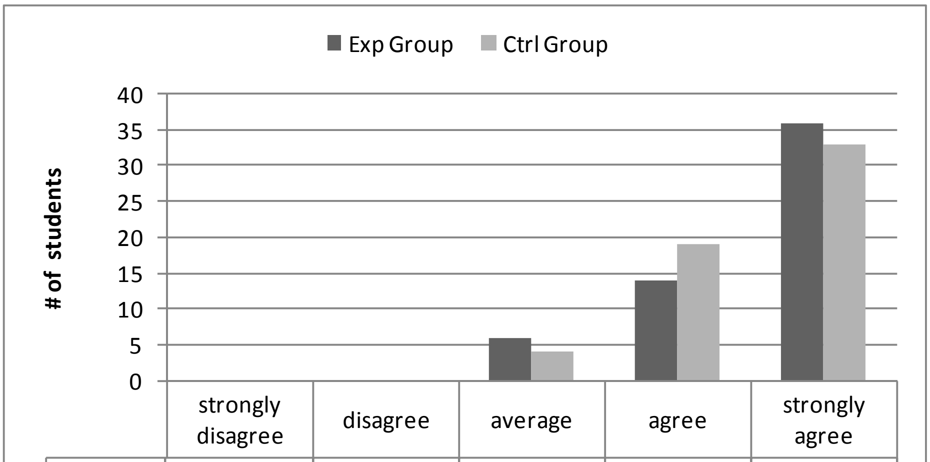
	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	6	17	33
Ctrl Group	0	0	4	21	31

(c) Q3: I agree that the conceptual learning suggestions provided by the TDLP are helpful for me to know the real cause of my conceptual learning problem in the computer programming course



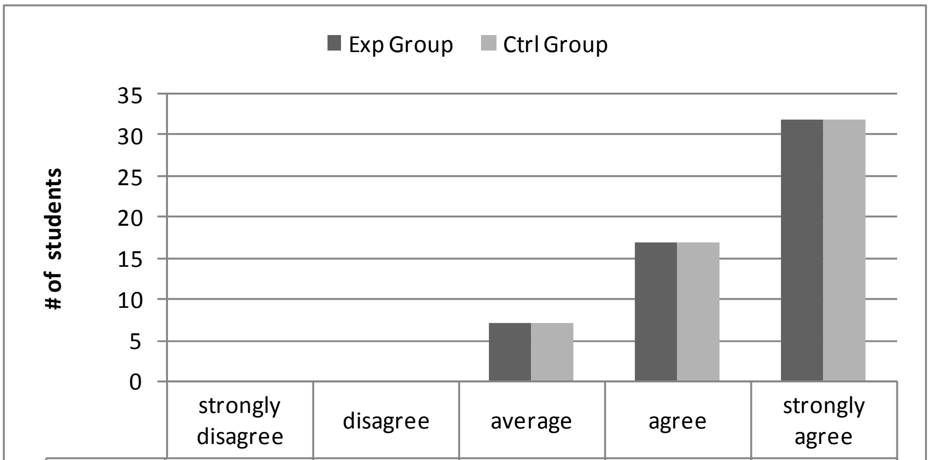
	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	6	21	29
Ctrl Group	0	0	10	18	28

(d) Q4: I agree that the conceptual learning suggestions provided by the TDLP are helpful to improve my learning outcome in computer programming course



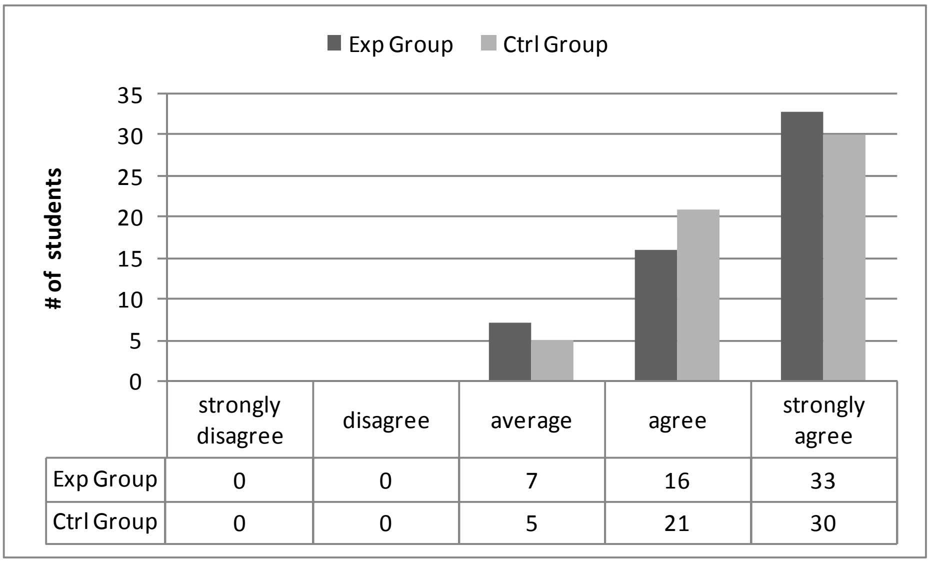
	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	6	14	36
Ctrl Group	0	0	4	19	33

(e) Q5: I agree that the conceptual learning suggestions provided by the TDLP can promote my learning confidence in the computer programming course



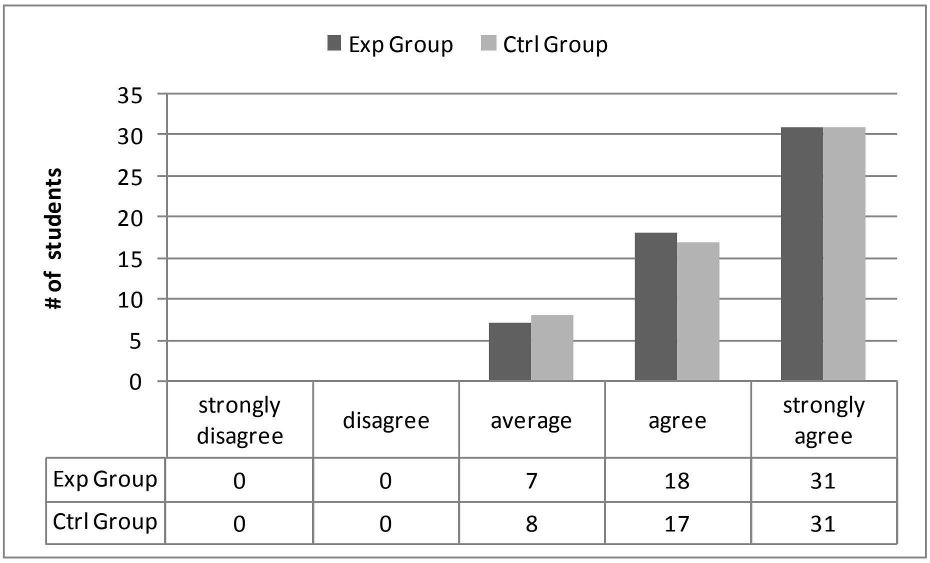
	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	7	17	32
Ctrl Group	0	0	7	17	32

(f) Q6: I agree that I can learn the computer programming course better if I receive the conceptual learning suggestions provided by the TDLP



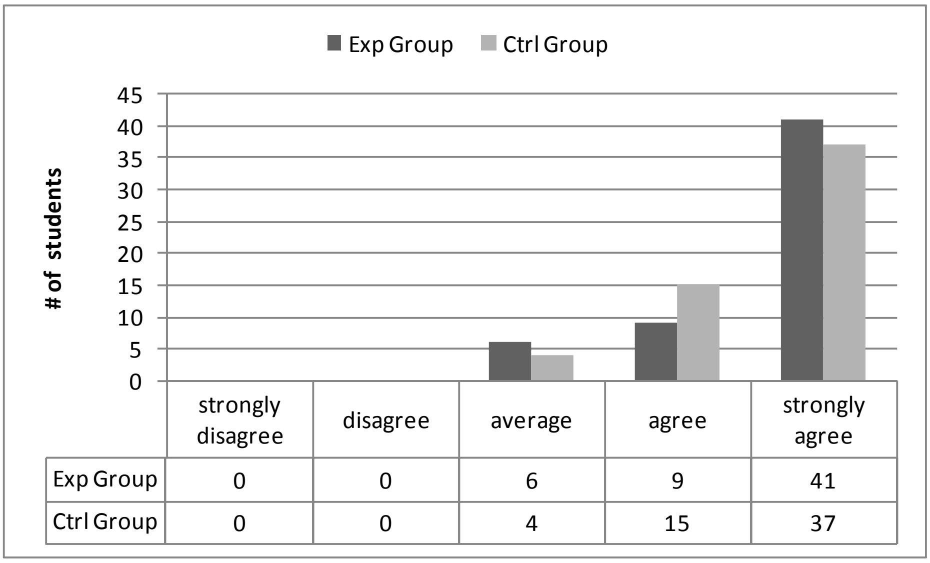
	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	7	16	33
Ctrl Group	0	0	5	21	30

(g) Q7: I agree that the learning activities relevant to the conceptual learning suggestions provided by the TDLP can support successful my learning in the computer programming course



	strongly disagree	disagree	average	agree	strongly agree
Exp Group	0	0	7	18	31
Ctrl Group	0	0	8	17	31

(h) Q8: I agree that the learning materials relevant to the conceptual learning suggestions provided by the TDLP can help to increase understanding in the computer programming course



(i) Q9: I will recommend the TDLP to other students

Figure 4.1 Bar graph of the responses to the questionnaire for evaluating students' satisfaction with learning guidance in the computer programming course

4.5 Time consumption in reconsidering weights

For a testing and diagnostic learning system that can support multiple experts, there are often a lot of conflicting cases. Experts have to spend their time to recheck, discuss and consider their weighting values again until all opinions are in the same direction. The more conflicting cases happen, the more time experts have to spend.

Table 4.5 The percentage of conflicting cases and the number of reconsidering weights

Testing and diagnostic system	% of conflicting cases	Times of reconsidering weights of experts
Integrating values using the majority density algorithm	(53/270) = 19.6%	103 times
Integrating values using the 14 rules Panjaburee et. al.'s method (2010)	(128/270) = 47.4%	263 times

Table 4.5 showed that the percentage and the number of conflicting cases (the number of reconsidering weighting values) resulting from Panjaburee et al (2010)'s approach to integrate weighting values were much higher than those resulting from the system using the majority density approach. There were $9 \times 30 = 270$ cases that experts had to determine their weighting values, since the concept-effect relationship (CER) model of the computer programming course was composed of nine concepts and the diagnostic test (pre-test) was composed of thirty test items. These results reflected that the testing and diagnostic learning system based on the majority density approach could help multiple experts to save time to reconsider weighting values; at the same time, it showed better performance at enhancing student knowledge in the computer programming course compared to the system based on Panjaburee et al (2010)'s set of rules.

From the analysis of the experimental data, there were four observations:

1. From the difference between the pre-test and the post-test results, the learning suggestions from the testing and diagnostic learning system based on the majority density approach could significantly improve the learning outcome of students.
2. From the comparison between the results of students in the control and experimental groups, it could be seen that the students who received the conceptual learning suggestions from the testing and diagnostic learning system based on the majority density approach significantly improved their learning outcome compared with those who received the conceptual learning suggestions from the testing and diagnostic learning system based on Panjaburee et al (2010)'s approach.
3. From the questionnaire results, the students who received the learning guidance from the testing and diagnostic learning system based on the majority density approach were satisfied with such conceptual learning guidance and believed that the system could help them to improve their knowledge.

4. Using the majority density approach to integrate weighting values from multiple experts could reduce the number of reconsidering cases compared with using Panjaburee et. al.'s approach.

In conclusion, the testing and diagnostic learning system based on the majority density approach could help students to improve the learning outcome after receiving conceptual learning suggestions and knowing their conceptual learning drawbacks. Due to the limitations of Panjaburee et al's approach, some unreliable integrated weighting values were generated; in addition, a large number of reconsidering cases appeared so that experts had to spend their time to resolve. On the contrary, the proposed system could generate high-quality integrated weighting values that were an important factor in improving the effectiveness of the system; that is, more accurate learning suggestions were provided to the students in the experimental group. Consequently, the students who received the learning guidance given by the proposed system could improve their learning achievement more than those who received the learning suggestions from the system based on the previous model could.

To effectively apply the testing and diagnostic learning system based on the majority density algorithm, these guidelines should be followed by teachers:

Step 1: One or more teachers extract all related concepts from the focused topic

Step 2: One or more teachers define the relationships among the concepts to construct a concept-effect relationship (CER) model

Step 3: One or more teachers design test items covering all concepts in the topic

Step 4: Multiple experts determine the relationship between each item and each concept by using KEISC part of the system (all values will be integrated using the majority density algorithm)

Step 5: All experts recheck, discuss, and reconsider their weighting values if necessary via KEISC part of the system

Moreover, these are guidelines for students particularly in the testing and diagnostic learning system based on the majority density algorithm:

Step 1: Receive introduction about the CER model that shows related concepts and their meanings in the topic that students are tested

Step 2: Do all test items by using TDLP part of the system in order to know the learning problems and their causes based on the CER model

Step 3: Pay attention to the suggestions from the system and follow them in order to solve the learning problems and improve learning achievement

Step 4: Do the post-test after receiving the suggestions from the system (if the teacher would like to compare students' learning achievement)

The majority density algorithm could be applied to a multi-expert testing and diagnostic learning system based on a concept-effect relationship (CER) model in other subjects such as mathematics, physics, chemistry and life sciences. In addition, it could be useful in other cooperative approaches such as group decision-making, and multi-expert systems in other domains, e.g., diagnostic system in the medical domain.

4.6 Sample cases that outliers appear

In developing the practical system, a lot of outliers easily appeared when the minority of the experts had different opinions to almost the majority of experts. This section reported some of the cases that outliers were detected in Table 4.6.

Table 4.6 The detected outliers in working of the practical system

Cases	Adjusted weighting value from experts					Detected outlier
	E_1	E_2	E_3	E_4	E_5	
1	0	5	5	5	5	0
2	1	5	5	5	5	1
3	2	5	5	5	5	2

Cases	Adjusted weighting value from experts					Detected outlier
	E_1	E_2	E_3	E_4	E_5	
4	3	5	5	5	5	3
5	0	0	1	1	5	5
6	0	0	1	1	4	4
7	0	0	1	1	3	3
8	3	3	3	4	5	5
9	0	3	5	5	5	0
10	-	3	5	5	5	3
11	3.5	5	5	5	5	0
12	2.5	3	3	3	5	5
13	1	1	1	1	2.5	2.5
14	0	0	0	3	5	5
15	0	0	0	3	-	3
16	0	0	0	2	4	4
17	0	0	0	2	-	2
18	0	0	0.5	1	3	3
19	0	0	0.5	1	2	2
20	0	0	0	0	4	4
21	0	0	0	0	3	3
22	0	0	0	0	2	2
23	2	2	2.5	3.5	4.5	4.5
24	2	2	2.5	3.5	-	3.5
25	0	0	0.5	0.5	2	2
26	0	0	0	1	2	2
27	0	0	0	0	1.5	1.5
28	0	2	3	3	3	0
29	0	1	1	1	3	3
30	0	0	0	1	5	5

CHAPTER V
CONCLUSIONS AND FUTURE WORKS

Diagnosing a student's learning problems is a very crucial task for a teacher. In a large classroom, it is not easy for the teacher to probe individual weakness of all students because it is time-consuming. Recently, there have been many studies which developed an online testing and diagnostic learning system for detecting the learning barriers of students and improving their learning outcomes in diverse fields, e.g. natural science, computer science, mathematics, physics, engineering, and health science. Among those systems, one of the well-known systems was proposed by Hwang (2003). The system introduced how to construct a testing and diagnostic learning system based on the concept-effect relationship (CER) model and it was widely adopted by many researchers; however, the limitation of this work is that it could support only a single expert to feed the knowledge and only one expert might provide some imprecise test item–concept relationships due to tiredness, carelessness or subjectivity; furthermore, researchers with different experiences could have different expertise or understanding of each portion of the knowledge. Therefore, a testing and diagnostic learning system that could support the cooperation of multiple experts was developed by Panjaburee et.al. (2010). The core of this system was to use a set of rules to integrate weighting values from multiple experts into one value. However, the method omitted majority consideration while integrating relationships; consequently, it might cause imprecise learning problem diagnosis and learning suggestions.

Because the integration of weighting values is an important issue for testing and diagnostic systems based on the CER model, the innovated procedure for integrating weighting values of the associated test items for each concept from multiple experts was developed. The proposed method considers the degree of confidence in making the decision about the weighting value, the majority opinion, and the reliability of the integrated weighting value. It provides a useful way to integrate the weighting values while developing testing and diagnostic systems based on the CER model.

The performance of this innovative approach has been compared with that of the existing approach by conducting an experiment on a computer programming course for undergraduate students in Rambhai Barni Rajabhat university. The results reveal that students in the experimental group, using a testing and diagnostic learning system constructed by the proposed approach, got significantly better learning outcome than those in the control group, using a system developed by the previously proposed approach. Interestingly, they were satisfied with the learning guidance and would recommend the system to other students for improving the learning outcome. Not only can the approach give more reliable or better quality integrated weighting values to enhance the entire learning diagnostic procedure based on the CER model, but the approach can also reduce reconsidering time of multiple experts.

Although our new approach revealed good performance at improving students' knowledge, the average post-test scores of them was not high (17.09 out of 30). To learn computer programming, a student should have the ability to abstract because it is a vital skill. Therefore, the supplemental materials provided by the system should be active and should include exercises or activities that encourage them to improve their thinking and problem-solving skills. Moreover, other useful information about students besides their learning problems is their learning styles and learning behavior. It is a good idea to integrate these factors into the system in order that it can be an adaptive learning system that is suitable for individual students.

Finally, although the original purpose of this research was to propose a cooperative work environment for integrating test item–concept relationships in education, the proposed approach could be applied to other cooperative applications such as group decision-making, and multi-expert knowledge elicitation in other domains, e.g., diagnostic systems in the medical domain.

REFERENCES

Aggarwal, C. C. (2005). *On abnormality detection in spuriously populated data streams.* Paper presented at the Proceedings of ACM SIAM Conference on Data Mining.

Agrawal, R., & Srikant, R. (1994). *Fast Algorithms for Mining Association Rules in Large Databases.* Paper presented at the Proceedings of the 20th International Conference on Very Large Databases.

Alfred, R. L., & Lum, G. D. (1988). Remedial program policies, student demographic characteristics, and academic achievement in community colleges. *Community Junior College Research Quarterly of Research and Practice 12*(2), 107-120.

Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor: it approaches the effectiveness of a human tutor. *Journal BYTE - Lecture notes in computer science, 10*(4), 159-175.

Atallah, M., Gwadera, R., & Szpankowski, W. (2004). *Detection of significant sets of episodes in event sequences.* Paper presented at the Proceedings of the Fourth IEEE International Conference on Data Mining.

Bai, S.-M., & Chen, S.-M. (2008a). Automatically constructing concept maps based on fuzzy rules for adapting learning systems *Expert Systems with Applications, 35*(1-2), 41–49.

Bai, S.-M., & Chen, S.-M. (2008b). Automatically constructing grade membership functions of fuzzy rules for students' evaluation *Expert Systems with Applications, 35*(1408–1414), 399–410.

Bai, S.-M., & Chen, S.-M. (2008c). Evaluating students' learning achievement using fuzzy membership functions and fuzzy rules *Expert Systems with Applications, 34*(1), 399–410.

Basu, S., & Meckesheimer, M. (2007). Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems, 11*(2), 137-154.

Branch, J., Szymanski, B., Giannella, C., Wolff, R., & Kargupta, H. (2006). *In-network outlier detection in wireless sensor networks.* Paper presented at the 26th IEEE International Conference on Distributed Computing System.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction, 6*, 87-129.

Casamayor, A., Amandi, A., & Campo, M. (2009). Intelligent assistance for teachers in collaborative e-learning environments. *Computers & Education, 53*(4), 1147–1154.

Chen, C.-M. (2008). Intelligent web-based learning system with personalized learning path guidance. *Computers & Education, 51*(2), 787–814.

Chen, L.-H. (2011). Enhancement of student learning performance using personalized diagnostic and remedial learning system. *Computers & Education, 56*(1), 289–299.

Chen, S.-M., & Bai, S.-M. (2009). Learning barriers diagnosis based on fuzzy rules for adaptive learning systems *Expert Systems with Applications, 36*(8), 11211–11220.

Cheng, S.-Y., Lin, C.-S., Chen, H.-H., & Heh, J.-S. (2005). Learning and diagnosis of individual and class conceptual perspectives: an intelligent systems approach using clustering techniques. *Computers & Education, 44*(3), 257–283.

Chu, H.-C., & Hwang, G.-J. (2008). A Delphi-based approach to developing expert systems with the cooperation of multiple experts. *Expert Systems with Applications, 34*(4), 2826–2840.

Du, W., Fang, L., & Ning, P. (2006). Lad: localization anomaly detection for wireless sensor networks. *Journal of Parallel and Distributed Computing, 66*(7), 874-886.

Eckerdel, A. (2009). *Novice programming students' learning of concepts and practise.* Upsalla University, Sweden.

Gerber, M., Grund, S., & Grote, G. (2008). Distributed collaboration activities in a blended learning scenario and the effects on learning performance. *Journal of Computer Assisted Learning, 24*(3), 232–244.

Ghosh, S., & Reilly, D. L. (1994). *Credit card fraud detection with a neural-network.* Paper presented at the Proceedings of the 27th Annual Hawaii International Conference on System Science.

Günel, K., & Aşlıyan, R. (2010). Extracting learning concepts from educational texts in intelligent tutoring systems automatically. *Expert Systems with Applications, 37*(7), 5017–5022.

Gwadera, R., Atallah, M. J., & Szpankowski, W. (2005). Reliable detection of episodes in event sequences. *Knowledge and Information Systems, 7*(4), 415-437.

Huang, L.-L., & Shimizu, A. (2006). A multi-expert approach for robust face detection. *Pattern Recognition, 39*(9), 1695-1703

Hwang, G.-H., Chen, J.-M., Hwang, G.-J., & Chu, H.-C. (2006). A time scale-oriented approach for building medical expert systems. *Expert Systems with Applications, 31*(2), 299–308.

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

Hwang, G.-J. (2007). Gray forecast approach for developing distance learning and diagnostic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37*(1), 98–108

Hwang, G.-J., Wu, P.-H., & Ke, H.-R. (2011). An interactive concept map approach to supporting mobile learning activities for natural science courses. *Computers & Education, 57*(4), 2272–2280.

Jong, B.-S., Chan, T.-Y., & Wu, Y.-L. (2007). Learning log explorer in e-Learning diagnosis. *EEE Transactions on Education, 50*(6), 216-228.

Jong, B., Lin, T., Wu, Y., & Chan, T. (2004). Diagnostic and remedial learning strategy based on conceptual graphs. *Jourmal of Computer Assiisted Learning, 20*, 377-386.

Kaburlasos, V. G., Marinagi, C. C., & Tsoukalas, V. T. (2008). Personalized multi-student improvement based on Bayesian cybernetics. *Computers & Education, 51*(4), 1430–1449.

Kao, G. Y.-M., Lin, S. S. J., & Sun, C.-T. (2008). Breaking concept boundaries to enhance creative potential: Using integrated concept maps for conceptual self-awareness. *Computers & Education, 51*(4), 1718–1728.

Keogh, E., Lonardi, S., & Chiu, B. Y.-c. (2002). *Finding surprising patterns in a time series database in linear time and space.* Paper presented at the Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data mining.

Klausmeier, H. J. (1976). Individually guided education: 1966-1980. *Journal of Teacher Education, 27*, 199-205.

Lee, C.-H., Lee, G.-G., & Leu, Y. (2009). Application of automatically constructed concept map of learning to conceptual diagnosis of e-learning. *Expert Systems with Applications, 36*(2), 1675-1684.

Léger, B., & Naud, O. (2009). Experimenting statecharts for multiple experts knowledge elicitation in agriculture. *Expert Systems with Applications, 36*(8), 11296-11303

Lemone, K. A., & Ching, W. (1996). Easing into C++: experiences with RoBOTL. *ACM SIGCSE Bulletin, 28*(4), 45-49.

Lin, J., Keogh, E., Fu, A., & Herle, H. V. (2005). *Approximations to magic: Finding unusual medical time series.* Paper presented at the Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems.

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, 57*(2), 1557–1570.

Liu, C.-C., Don, P.-H., & Tsai, C.-M. (2005). Assessment based on linkage patterns in concept maps. *Journal of Information Science and Engineering, 21*(5), 873-890.

Lo, J.-J., Wang, H.-M., & Yeh, S.-W. (2004). Effects of confidence scores and remedial instruction on prepositions learning in adaptive hypermedia. *Computers & Education, 42*(1), 45-63.

Manevitz, L. M., & Yousef, M. (2002). One-class svms for document classification. *The Journal of Machine Learning Research, 2*, 139-154.

Manning, S., & Dix, A. (2008). Identifying students' mathematica lskills from a multiple-choice diagnostic test using an iterative technique to minimise false positives. *Computers & Education, 51*(3), 1154–1171.

Martin, B. I. (2001). *Intelligent Tutoring Systems: The Practical Implementation of Constraint-based Modelling.* University of Canterbury.

Medsker, L. (1995). Knowledge acquisition from multiple experts: problems and issues. *Expert Systems with Applications, 9*(1), 35–40.

Mittal, S., & Dym, C. L. (1985). Knowledge acquisition from multiple experts. *AI Magazine, 6*(2), 32-36.

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

Paolucci, R. (1998). *Hypermedia and learning: The relationship of cognitive style and knowledge structure.* Paper presented at the Proceedings of ED-MEDIA 98.

Pokrajac, D., Lazarevic, A., & Latecki, L. J. (2007). *Incremental local outlier detection for data streams.* Paper presented at the IEEE Symposium on Computational Intelligence and Data Mining.

Roueche, J. E., Baker, G. A., & Roueche, S. D. (1984). *College responses to low-achieving students: A national study*. (H. M. S. Corp o. Document Number)

Sanders, W. L., Wright, S. P., & Horn, S. P. (1997). Teacher and classroom context effects on student: achievement: implications for teacher evaluation. *Journal of Personnel Evaluation in Education, 11*, 57-67.

Singh, S., & Markou, M. (2004). An Approach to Novelty Detection Applied to the Classification of Image Regions. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 396-407

Soloway, E., & Spohrer, J. C. (1989). *Studying the novice programmer*. Hillside, NJ: Erlbaum.

Srivastava, A. (2006). *Enabling the Discovery of Recurring Anomalies in Aerospace System Problem Reports using High-Dimensional Clustering Techniques.* Paper presented at the IEEE Aerospace Conference.

Stankov, S., Rosić, M., Žitko, B., & Grubišić, A. (2008). Tex-Sys model for building intelligent tutoring systems. *Computers & Education, 51*(3), 1017–1036.

Tseng, J. C. R., Chu, H.-C., Hwang, G.-J., & Tsai, C.-C. (2008). Development of an adaptive learning system with two sources of personalization information. *Computers & Education, 51*(2), 776-786.

Tseng, S. S., Sue, P. C., Su, J. M., Weng, J. F., & Tsai, W. N. (2007). A new approach for constructing the concept map. *Computers & Education, 49*(3), 691-707.

Winslow, L. E. (1996). Programming pedagogy. *ACM SIGCSE Bulletin, 28*(3), 17-22.

Wong, W.-K., Moore, A., Cooper, G., & Wagner, M. (2003). *Bayesian network anomaly pattern detection for disease outbreaks.* Paper presented at the Proceedings of the 20th International Conference on Machine Learning.

APPENDICES

APPENDIX A

Form of Informed and Voluntary Consent to Participate in Research

DATE………. /………. /……….

My name is.., aged............. years old, now living at the address no.......................road/street….. sub-district/tambon…………................. district/amphur………...……………………… province.................................... Postal code……….…..... Tel. No.:................................

I hereby express my consent to participate as a subject in the research project entitled a majority density approach with the cooperation of multiple experts for developing testing and diagnostic learning systems based on a concept-effect relationship model.

In so doing, I am informed of the research project's origin and purposes; its procedural details to carry out or to be carried out; its expected benefits and risks that may occur to the subjects, including methods to prevent and handle harmful consequences; and remuneration, and expense. I thoroughly read the detailed statements in the information sheet given to the research subjects. I was also given explanations and my questions were answered by the head of the research project.

I therefore consent to participate as a subject in this research project

On the condition that I have any questions about the research procedures, or on the condition that I suffer from an undesirable side effect from this research, I can contact Mr. Dechawut Wanichsan, Tel 66-1-5783107.

On the condition that I am not treated as indicated in the information sheet distributed to the subjects, I can contact the Chair of Mahidol University Institutional Review Board (MU-IRB) at the office of MU-IRB, Research Administration Division, Office of the President, Mahidol University, Tel 66-2-8496223-5, Fax 66-2-8496223.

I am aware of my right to further information concerning benefits and risks from the participation in the research project and my right to withdraw or refrain from the participation anytime without any consequence on the service or health care I am to receive in the future. I consent to the researchers' use of my private information obtained in this research, but do not consent to an individual disclosure of private information. The information must be presented as part of the research results as a whole.

I thoroughly understand the statements in the information sheet for the research subjects and in this consent form. I thereby give my signature.

Signature... Participants

(...) Date.......................................

Signature... Head of Research Project

(Mr. Dechawut Wanichsan) Date.......................................

Participant Information Sheet

In this document, there may be some statements that you do not understand. Please ask the principal investigator or his/her representative to give you explanations until they are well understood. To help your decision making in participating the research, you may bring this document home to read and consult your relatives, intimates, personal doctor or other doctor.

Title of Research Project : A majority density approach with the cooperation of multiple experts for developing testing and diagnostic learning systems based on a concept-effect relationship model

Name of Researcher : Mr. Dechawut Wanichsan

Research Site : Institute for Innovative Learning, Mahidol University
999 Phuttamonton 4 Road, Salaya, Phuttamonton,
Nakorn Pathom 73170
Tel. 081-578-3107 E-mail address. kook260g@hotmail.com

This research project aims to develop a testing and diagnostic learning system that could provide appropriate materials and beneficial suggestions for students to improve their knowledge and learning achievement. The expected benefits is the developed system that could diagnose freshman students' learning problems in a "Computer programming I" course, and could provide useful and appropriate suggestions for students to improve their knowledge in troubled topics.

You are invited to participate in this research project because you are a freshman and passed a "Computer programming I" course in Rambhai Barni Rajabhat University; moreover you can give useful data useful for conducting the research project.

There will be 60 participants that were randomly separated as Control Group and Experimental Group; each group is composed of 30 participants. The research project will last 4 weeks. If you decide to participate this research, the steps in conducting research are:

1. You take a pre - test which is also used as a diagnostic test comprising 30 test items from developed testing and diagnostic learning system in 2 hours.
2. All Participating students will be randomly divided into two groups i.e., Control Group and Experimental Group; each group comprises 30 students.
    a) Control group : you receive learning suggestions from the concept effect model that integrated <test item, concept> weightings from multiple experts using the previous method, and follow them to improve your knowledge in 15 days.
    b) Experimental group : you receive learning suggestions from the concept effect model that integrated <test item, concept> weightings from multiple experts using a majority-density approach, and follow them to improve your knowledge in 15 days.
3. You take post-test comprising 30 test items from developed testing and diagnostic learning system in 2 hours.
4. You take 18-item questionnaire used to investigate the students' satisfaction about the learning guidance.
5. You may be interviewee to provide more opinions about using a testing and diagnostic system; all obtained data will be demolished after finishing the research.

There are no remuneration and expense for the participant to be responsible in this research project. If adverse events/unanticipated events occur, having any questions about the research procedures, or on the condition that I suffer from an undesirable side effect of this research, I can contact Mr. Dechawut Wanichsan, Tel 66-1-5783107.

If relevant information arises about the benefits and risks of the research project, the researcher will inform the participant immediately and without concealment.

The participant's private information will be kept confidential, it will not be subject to an individual disclosure, but will be included in the research report as part of the overall results. Individual information may be examined by groups of persons e.g. from a funding organization, a government agent in charge, the ethics committee, etc.

The participant has the right to withdraw from the project at any time without prior notice. And the refusal to participate or the withdrawal from the research project will not at all affect the proper service or treatment that he/she will receive.

On the condition that you are not treated as indicated in this information sheet, you can contact the Chair of Mahidol University Institutional Review Board (MU-IRB) at the office of MU-IRB, Research Administration Division, Office of the President, Mahidol University, Tel 66-2-8496223-5, Fax 66-2-8496223.

I thoroughly read the details in this document.

Signature………………………… Participant

(..)

Date………………….....................................

COA. No. MU-IRB 2011/164.1808

Certificate of Approval
Mahidol University Institutional Review Board (MU-IRB)

Title of Project: A Majority Density Approach with the Coooperation of Multiple Expert for Developing Testing and Diagnostic Learning Systems Based on a Concept-Effect Relationship Model
(Thesis for Ph.D.)

Principal Investigator: Mr. Dechawut Wanichsan

Co-Investigators: Dr. Patcharin Panjaburee
Miss Sasithorn Chookaew

Name of Institution: Institute for Innovative Learning

Approval includes: 1) MU-IRB Submission form version received date 17August 2011
2) Participant Information Sheet version date 23 May 2011
3) Informed Consent form version date 23 May 2011
4) Assessment form for Screening version date 23 May 2011
5) Interview Guideline version received date 23 May 2011
6) Questionnaire version received date 23 May 2011

Mahidol University Institutional Review Board is in full compliance with International Guidelines for Human Research Protection such as Declaration of Helsinki, The Belmont Report, CIOMS Guidelines and the International Conference on Harmonization in Good Clinical Practice (ICH-GCP)

Date of Approval: 18 August 2011

Date of Expiration: 17 August 2012

Signature of Chair: ....
(Professor Shusee Visalyaputra)

Signature of Head of the Institute: ....
(Professor Sansanee Chaiyaroj)
Vice President for Research and Academic Affairs

Office of the President, Mahidol University, 999 Phuttamonthon 4 Rd., Salaya, Phuttamonthon District, Nakhon Pathom 73170. Tel. (662) 8496223-5 Fax. (662) 8496223

APPENDIX B

Pre-test on topic "Computer Programming I"

First year students Computer Programming

30 Test Items 120 Minutes

Instructions

1. This pre-test consists of 30 multiple-choice test items concerning the basic knowledge related to the study of computer programming. Each item was scored one point.
2. Please select the best choice for each test item. When you complete the pre-test, please submit your answers to a Testing and Diagnostic System (TDLP).
3. The TDLP will analyze the pre-test answers and provide personalized learning guidance.

1. According to a list of variable names, how many names causing error are there?
    a) `SpiderMan`
    b) `_score_1`
    c) `1stStep`
    d) `$__$`

    1) 1 name
    2) 2 names
    3) 3 names
    4) 4 names

2. According to list of declaring variables, how many commands causing error are there?

   a) `Short score = 15;`

   b) `bool check = “false”;`

   c) `long short temp = 5;`

   d) `char petName = ‘$’;`

   1) 1 command

   2) 2 commands

   3) 3 commands

   4) 4 commands

3. According to list of assigning a data type, how many improper cases are there?

   a) `“false”` is `bool` as a data type

   b) `70000` is `unsigned short` as a data type

   c) `-3200` is `float` as a data type

   d) `-0.75` is `long double` as a data type

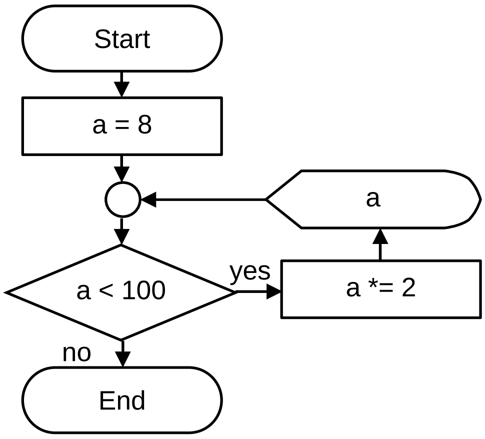
   1) 1 case

   2) 2 cases

   3) 3 cases

   4) 4 cases

4. After a flowchart was executed, what is the last value of *a* printed on a screen?



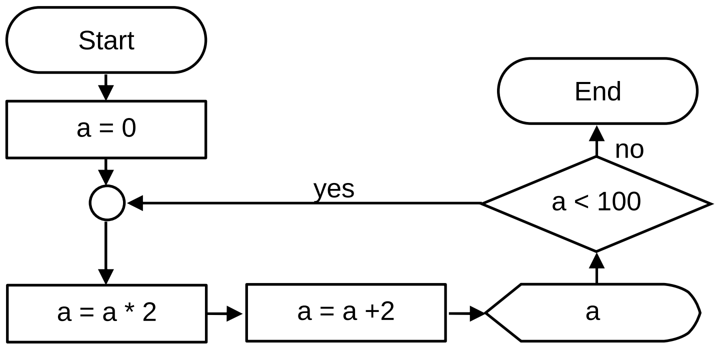
1) 32

2) 64

3) 128

4) 256

5. After a flowchart was executed, how many times was *a* printed on screen?



1) 4 times

2) 5 times

3) 6 times

4) 7 times

6. How many lines of error does the program have?

```
#include “stdio.h”;
int main(){
   int age;
   scanf(“%d”,a);
   printf(your age is %d,a);
   getch();
}
```

1) 1 line

2) 2 lines

3) 3 lines

4) 4 lines

Use the following commands to answer question number 7-10

```
int u = 1, v = 3, w = 5, x = 2, y = 4, z = 6;
int ans = 0;
float a = 1, b = 2;
```

7. From three commands, how many commands causing error are there?

    a) `a += w * v / 2.0;`

    b) `ans = (0 = true)&&(1 = false);`

    c) `y %= z – v * x + w;`

    1) 1 command

    2) 2 commands

    3) 3 commands

    4) All can correctly work

8. What is a value of *a* calculated from `a += w * v / b;` ?

    1) All choices are incorrect

    2) 7

    3) 7.5

    4) 8.5

9. What is a value of *x* calculated from `u += x *= z – u * v;` ?

    1) All choices are incorrect

    2) 15

    3) 6

    4) 30

10. What is a value of *y* calculated from `y += z * x += w * u;` ?

    1) All choices are incorrect

    2) 38

    3) 42

    4) 46

Use the following program to answer question number 11-14

```
1  int func_a(int m, int &n){
2     if (m==1){
3        return 2;
4     } else {
5        return m + func_a(m-1,n);
6     }
7  }
8  bool func_b(int &x,int y){
9     int temp;
10    temp = x;
11    x = y;
12    y = temp;
13    if (x < y){
14       return 999;
15    } else {
16       return 0;
17    }
18 }
19 int main(){
20    int a = 4, b = 7;
21    cout << func_a(a,b);
22    cout << b << “\t” << a;
23    cout << func_b(a,b);
24    cout << a << “\t” << b;
25 }
```

11. What is a result of command line#21?

1) 7

2) 9

3) 10

4) 11

12. What is a result of command line#22?

1) 4 4

2) 7 4

3) 4 7

4) 11 11

13. What is a result of command line#23?

1) 999

2) `true`

3) `false`

4) All choices are incorrect

14. What is a result of command line#24?

1) 7 4

2) 7 7

3) 4 4

4) 4 7

15. What is a result of command line#6?

```
1 int main(){
2    int sum = 0;
3    for (int i=4;i<19;i+=3){
4       sum+=i;
5    }
6    cout << sum;
7 }
```

1) 34

2) 50

3) 69

4) All choices are incorrect

16. How many times are command line#4 executed?

```
1 int main(){
2    int x = 5;
3    while (++x < 10){
4       cout << “O”;
5    }
6 }
```

1) 4

2) 5

3) 6

4) All choices are incorrect

17. What is a result of command line#7?

```
1 int main(){
2    int sum = 0, i = 2;
3    do{
4       sum+=i;
5       i*=2;
6    }while(sum<35);
7    cout << sum;
8 }
```

1) 30

2) 182

3) 306

4) All choices are incorrect

Use the following program to answer question number 18-19

```
int main(){
   for (int i=3;i<8;i++){
      for(int j=i;j<6;j++)
         cout << “x”;
      }
      cout << “O”;
   }
}
```

18. How many times is the character ‘x’ printed on screen?

1) 5

2) 6

3) 7

4) All choices are incorrect

19. How many times is the character `'o'` printed on screen?

    1) 5
    2) 6
    3) 7
    4) 8

Use the following program to answer question number 20-22

```
1 int main(){
2    int myArray[X][Y]={{5,2},{3,4},{0,7},{1,6},{9,8}};
3    int a = 5;
4    do{
5       a-=1;
6       myArray[a][1] = 15;
7    }while(a != 0);
8 }
```

20. What are the suitable values for *X* and *Y* for command line#2?

    1) `x = 2, y = 5`
    2) `x = 1, y = 4`
    3) `x = 5, y = 2`
    4) `x = 4, y = 1`

21. In command line#2, If the value `0` in *myArray* was changed to be `11`, what would be the command used for changing the value?

    1) `myArray[3][2]=11;`
    2) `myArray[2][1]=11;`
    3) `myArray[3][1]=11;`
    4) All choices are incorrect

22. After the program was executed, how many values in *myArray* being `15` are there?

    1) 5
    2) 3
    3) 1
    4) All choices are incorrect

Use the following program to answer questions number 23-24

This Program used for categorizing score of a student to a grade; criteria for the category were shown below:

```
 1  int main(){
 2     int score;
 3     cin >> score;
 4     if((score >-1) && (score <= 100)){
 5        if(score >90){
 6           cout << “you got A”;
 7        } else if (score <=54){
 8           cout << “you got F”;
 9        } else if (score >84){
10           cout << “you got B+”;
11        } else if (score <=60){
12           cout << “you got D”;
13        } else if (score <66){
14           cout << “you got D+”;
15        } else if (score >=81){
16           cout << “you got B”;
17        } else if (score >73){
18           cout << “you got C+”;
19        } else {
20           cout << “you got C”;
21        }
22     } else {
23        cout << “error”;
24     }
25  }
```

23. If `73` were entered in command line#3, what would be a result of the program?

1) you got B

2) you got C+

3) you got C

4) you got D+

24. If `61` were entered in command line#3, what would be a result of the program?

1) you got B

2) you got C+

3) you got C

4) you got D+

25. According to sentences, how many correct statements are there?

   a) The first function of C computer programming language is main()

   b) The symbol /*   */ can be used for commenting a single-line command

   c) include“stdio.h” has to be declared before using getch()

   d) include<iostream> has to be declared before using scanf()

1) 1 statement

2) 2 statements

3) 3 statements

4) 4 statements

Use the following program to answer question number 26-27

```
1 int main(){
2    for (int i=1;i<=15;i++){
3       if ((i%2 == 0)||(i%3 == 0)){
4          cout << i << “\t”;
5       }
6    }
7 }
```

26. After the program was executed, how many printed values on screen are there?

1) 5 values

2) 7 values

3) 10 values

4) 12 values

27. If we want to print a sequence of numbers as

1   2   5   7   10   11   13   14

In command line#3, what is an expression of if?

1) (i%3 != 0)||(i%4 != 0)

2) (i%3 != 0)&&(i%4 != 0)

3) (i%3 == 0)&&(i%4 == 0)

4) (i%3 == 0)||(i%4 == 0)

Use the following program to answer question number 28-30

```
1 int main(){
2    int a = 3, b = 4, c = 5;
3    char s = ‘o’;
4    cout << “Hello World”;
5    cout << a+b+c;
6    cin >> s;
7 }
```

28. What is the command that can give the same result as command line#4?

    1) `printf(Hello World);`

    2) `cout << “Hello” << “World”;`

    3) `cout << “Hello\tWorld”;`

    4) `printf(“Hell%c W%crld”,s,s);`

29. What is the command that can give the same result as command line#5?

    1) `printf(“a+b+c”);`

    2) `printf(%d,3+4+5);`

    3) `scanf(%d,3+4+5);`

    4) `printf(“%d”,a+b+c);`

30. What is the command that has the same working as command line#6?

    1) `scanf(“%c”,&s);`

    2) `scanf(s);`

    3) `scanf(“%c”,s);`

    4) `scanf(“%s”,&s);`

APPENDIX C

Pre-test on topic "Computer Programming I"

First year students | Computer Programming

30 Test Items | 120 Minutes

Instructions

1. This post-test consists of 30 multiple-choice test items concerning the basic knowledge related to the study of computer programming. Each item was scored one point.
2. Please select the best choice for each test item. When you complete the pre-test, please submit your answers to a Testing and Diagnostic System (TDLP).
3. The TDLP will analyze the pre-test answers and provide personalized learning guidance.

---

1. What is a result of command line#6?

```
1  int main(){
2     int sum = 1;
3     for (int i=2;i<12;i+=2){
4        sum+=i;
5     }
6     cout << sum;
7  }
```

1) All choices are incorrect

2) 30

3) 31

4) 43

2. How many times are command line#4 executed?

```
1 int main(){
2    int x = 3;
3    while (++x < 7){
4       cout << “O”;
5    }
6 }
```

1) All choices are incorrect

2) 4

3) 5

4) 6

3. What is a result of command line#7?

```
1 int main(){
2    int sum = 2, i = 0;
3    do{
4       i+=2;
5       sum*=i;
6    }while(sum<45);
7    cout << sum;
8 }
```

1) 48

2) 60

3) 62

4) All choices are incorrect

4. According to the list of commands for declaring variables, how many commands causing error are there?

e) `float score = 15;`

f) `bool check = 1;`

g) `long double = 50.75;`

h) `char *petName = ‘Bobby’;`

1) 1 command
2) 2 commands
3) 3 commands
4) 4 commands

5. According to a list of variable names, how many names causing error are there?

e) `$25baht`
f) `12call`
g) `Room39`
h) `_error_`

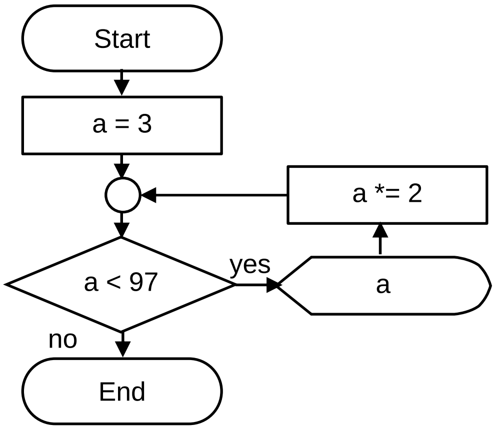
1) 1 name
2) 2 names
3) 3 names
4) 4 names

6. According to list of assigning a data type, how many improper cases are there?

a)	`“true”`	is	`bool`	as a data type
b)	`-36`	is	`unsigned short`	as a data type
c)	`-32.75`	is	`float`	as a data type
d)	`-3.25`	is	`long long`	as a data type

1) 1 case
2) 2 cases
3) 3 cases
4) 4 cases

7. After a flowchart was executed, what is the last value of *a* printed on a screen?



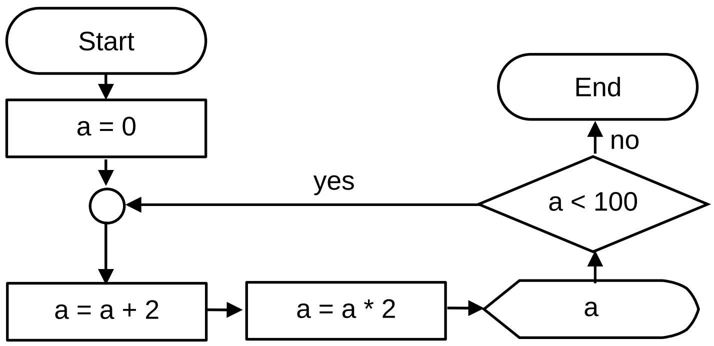
1) 24

2) 48

3) 96

4) 192

8. After a flowchart was executed, how many times was *a* printed on screen?



1) 4 times

2) 5 times

3) 6 times

4) 7 times

Use the following commands to answer question number 9-12

```
int u = 1, v = 3, w = 5, x = 2, y = 4, z = 6;
int ans = 0;
float a = 1, b = 2;
```

9. From three commands, how many commands causing error are there?

   d) `x = u % ans;`

   e) `ans = (2 == true)&&(0 == false);`

   f) `b /= w + v;`

   1) All can correctly work
   2) 1 command
   3) 2 commands
   4) 3 commands

10. What is a value of *b* calculated from `b /= w + v;` ?

   1) All choices are incorrect
   2) 0
   3) 0.25
   4) 0.5

11. What is a value of *x* calculated from `x += u *= z – u *= v;` ?

   1) All choices are incorrect
   2) 15
   3) 6
   4) 30

12. What is a value of *y* calculated from `z += y *= x = w * u;` ?

   1) All choices are incorrect
   2) 20
   3) 26
   4) 30

Use the following program to answer question number 13-15

```
1  int main(){
2     int myArray[X][Y]={{5,2},{3,4},{1,7},{1,6},{9,8}
      ,{6,0},{2,8}};
3     int a = 4;
4     do{
5        a-=1;
6        myArray[a][1] = 15;
7     }while(a > 0);
8  }
```

13. What are the suitable values for `X` and `Y` for command line#2
    1) `x = 2, y = 7`
    2) `x = 1, y = 6`
    3) `x = 7, y = 2`
    4) `x = 6, y = 1`

14. In command line#2, If the value `0` in *`myArray`* was changed to be `11`, what would be the command used for changing the value?
    1) `myArray[6][2]=11;`
    2) `myArray[2][6]=11;`
    3) `myArray[5][1]=11;`
    4) `myArray[1][5]=11;`

15. After the program was executed, how many values in *`myArray`* being `15` are there?
    1) 4
    2) 3
    3) 2
    4) All choices are incorrect

Use the following program to answer question number 16-18

```
1 int main(){
2    int a = 3, b = 4, c = 5;
3    char *o = “o”;
4    cout << “Hello World”;
5    cout << a+b+c;
6    cin >> o;
7 }
```

16. What is the command that can give the same result as command line#4?
    1) `printf(“Hell%o W%orld”,o,o);`
    2) `cout << “Hello” <<“World”;`
    3) `cout << “Hello\tWorld”;`
    4) `printf(“Hell%s W%srld”,o,o);`

17. What is the command that can give the same result as command line#5?
    1) `printf(“a+b+c”);`
    2) `printf(%d,3+4+5);`
    3) `printf(“3+4+5”);`
    4) `printf(“%d”,a+b+c);`

18. What is the command that has the same working as command line#6?
    1) `scanf(“%c”,&o);`
    2) `scanf(o);`
    3) `scanf(“%s”,&o);`
    4) `scanf(“%s”,o);`

Use the following program to answer question number 19-20

```
1 int main(){
2    for (int i=1;i<=15;i++){
3       if ((i%2 == 0)&&(i%3 != 0)){
4          cout << i << “\t”;
5       }
6    }
7 }
```

19. After the program was executed, how many printed values on screen are there?

    1) 5 values

    2) 7 values

    3) 10 values

    4) 12 values

20. If we want to print a sequence of numbers as

    `3    4    6    8    9    12    15`

    In command line#3, what is an expression of `if`?

    1) `(i%3 != 0)||(i%4 != 0)`

    2) `(i%3 != 0)&&(i%4 != 0)`

    3) `(i%3 == 0)&&(i%4 == 0)`

    4) `(i%3 == 0)||(i%4 == 0)`

21. According to sentences, how many correct statements are there?

    e) The first function of C computer programming language is `main()`

    f) The symbol `\\` can be used for commenting a single-line command

    g) `include“iostream”` has to be declared before using `getch()`

    h) `include<stdio.h>` has to be declared before using `endl`

    1) 1 statement

    2) 2 statements

    3) 3 statements

    4) 4 statements

22. How many lines of error does the program have?

```
#include “stdio.h”
#include “conio.h”
int main(){
   int age;
   scanf(“%d”,a);
   printf(your age is “%d”,a);
   getch();
}
```

1) 1 line

2) 2 lines

3) 3 lines

4) 4 lines

Use the following commands to answer question number 23-26

```
int func_a(int m, int &n){
   if (m==1){
      n = m;
      return 5;
   } else {
      return func_a(m-1,n) * m;
   }
}
bool func_b(int &x,int &y){
   int temp = 7;
   x = temp;
   temp = y;
   y = x;
   if (x < y){
      return 1;
   } else {
      return 0;
   }
}
int main(){
   int a = 3, b = 7;
   cout << func_a(a,b);
   cout << a << “\t” << b;
   cout << func_b(a,b);
   cout << a << “\t” << b;
}
```

23. What is a result of command line#22?

    1) 6
    2) 12
    3) 18
    4) 30

24. What is a result of command line#23?

    1) All choices are incorrect
    2) 3 1
    3) 3 7
    4) 3 3

25. What is a result of command line#24?

    1) All choices are incorrect
    2) 999
    3) `true`
    4) `false`

26. What is a result of command line#25?

    1) 3 3
    2) 1 3
    3) 7 3
    4) 7 7

Use the following commands to answer questions number 27-28

```
int main(){
   for (int i=4;i<=7;i++){
      for(int j=i;j<=6;j++)
         cout << “x”;
      }
      cout << “O”;
   }
}
```

27. How many times is the character ‘x’ printed on screen?

    1) 3

    2) 4

    3) 5

    4) All choices are incorrect

28. How many times is the character ‘O’ printed on screen?

    1) 4

    2) 5

    3) 6

    4) 7

Use the following commands to answer questions number 29-30

This Program used for categorizing score of a student to a grade; criteria to category were shown below:

```
int main(){
   int score;
   cin >> score;
   if((score >-1) && (score <= 100)){
      if(score >92){
         cout << “you got A”;
      } else if (score <=54){
         cout << “you got F”;
      } else if (score >85){
         cout << “you got B+”;
      } else if (score <=60){
         cout << “you got D”;
      } else if (score <66){
         cout << “you got D+”;
      } else if (score >=81){
         cout << “you got B”;
      } else if (score >73){
         cout << “you got C+”;
      } else {
         cout << “you got C”;
      }
   } else {
      cout << “error”;
   }
}
```

29. If `79` were entered in command line#3, what would be a result of the program?

    1) you got C

    2) you got C+

    3) you got B

    4) you got B+

30. If `85` were entered in command line#3, what would be a result of the program?

    1) you got B+

    2) you got B

    3) you got C+

    4) you got C

APPENDIX D

Test score of students for "Computer Programming I"

Experimental Group			Control Group		
ID	Pre test	Post test	ID	Pre test	Post test
A001	7	16	B001	8	16
A002	6	13	B002	8	11
A003	9	12	B003	11	17
A004	16	10	B004	5	16
A005	9	13	B005	9	9
A006	8	18	B006	9	17
A007	10	14	B007	7	11
A008	11	19	B008	8	11
A009	8	14	B009	11	14
A010	12	26	B010	10	18
A011	11	14	B011	6	13
A012	11	21	B012	8	12
A013	9	28	B013	11	15
A014	10	22	B014	14	19
A015	8	16	B015	9	12
A016	8	21	B016	10	9
A017	12	17	B017	13	21
A018	11	20	B018	6	19
A019	8	19	B019	10	20
A020	7	16	B020	11	20
A021	10	11	B021	17	24
A022	12	19	B022	14	24
A023	14	23	B023	15	21

Experimental Group			Control Group		
ID	Pre test	Post test	ID	Pre test	Post test
A024	13	21	B024	16	15
A025	4	15	B025	10	13
A026	9	17	B026	10	19
A027	11	17	B027	13	20
A028	13	17	B028	10	23
A029	11	17	B029	14	17
A030	12	15	B030	13	19
A031	8	15	B031	7	10
A032	12	10	B032	6	20
A033	7	14	B033	9	17
A034	11	18	B034	9	11
A035	8	17	B035	12	20
A036	10	16	B036	8	21
A037	24	23	B037	10	19
A038	12	24	B038	11	11
A039	9	21	B039	11	11
A040	14	22	B040	8	12
A041	11	28	B041	10	19
A042	12	18	B042	12	18
A043	22	29	B043	8	12
A044	12	22	B044	19	25
A045	7	13	B045	11	8
A046	11	10	B046	9	16
A047	9	12	B047	6	9
A048	12	15	B048	13	15
A049	10	13	B049	8	10
A050	11	20	B050	9	11
A051	9	14	B051	6	10
A052	14	9	B052	10	10
A053	7	15	B053	7	14

Experimental Group			Control Group		
ID	Pre test	Post test	ID	Pre test	Post test
A054	11	13	B054	10	13
A055	6	12	B055	14	10
A056	9	13	B056	10	8

APPENDIX E

MAHIDOL UNIVERSITY
Wisdom of the Land

Questionnaire for Students' Satisfaction about Learning Suggestions Provided by a Testing and Diagnosis Learning Problem (TDLP) System in the Computer programming course

Objective:

The questionnaire was employed to investigate students' satisfaction after they received the remedial learning suggestions for individual students. Your answers will be kept in secret. It will be used in research that does not have any effects on an individual or school.

Instruction:

Please consider each statement and use ✓ in the space of the table which best describes your level of agreement about the learning suggestions and learning materials provided by the testing and diagnostic learning problem system. The criterion of each level of agreement is following:

5 means strongly agree

4 means agree

3 means neutral

2 mean disagree

1 means strongly disagree

Part 1: Personal Data

Gender ☐ Male ☐ Female

Student's ID……………………………… Year…………..……..……..……………..

Major ……………………………………….. Faculty…………..……..……..……..

Cumulative Grade Point Average

☐ < 1.00 ☐ 1.00 – 1.99 ☐ 2.00 – 2.99 ☐ 3.00 – 4.00

Part 2: Students' Satisfaction about Learning Suggestions provided by a TDLP system

Questions	**Level of agreement**				
	5	**4**	**3**	**2**	**1**
1. I am satisfied with the conceptual learning suggestions provided by the TDLP					
2. I agree that the conceptual learning suggestions provided by the TDLP are useful to my learning in the computer programming course					
3. I agree that the conceptual learning suggestions provided by the TDLP are helpful for me to know the real cause of my conceptual learning problem in the computer programming course					
4. I agree that the conceptual learning suggestions provided by the TDLP are helpful to improve my learning outcome in the computer programming course					
5. I agree that the conceptual learning suggestions provided by the TDLP can promote my learning confidence in the computer programming course					
6. I agree that I can learn the computer programming course better if I receive the conceptual learning suggestions provided by the TDLP					

Questions	**Level of agreement**				
	5	**4**	**3**	**2**	**1**
7. I agree that the learning materials relevant to the conceptual learning suggestions provided by the TDLP can support successful learning in the computer programming course					
8. I agree that the learning materials relevant to the conceptual learning suggestions provided by the TDLP can help to increase understanding in the computer programming course					
9. I will recommend using the TDLP with other students					

APPENDIX F

Algorithm : majority-density algorithm (PHP script)

```
function mad($listData){
	$temp = 0;
	$avg = array_sum($listData) / sizeof($listData);
	foreach ($listData as $e){
		$temp += abs($e - $avg);
	}
	return $temp = $temp / sizeof($listData);
}

function eta($nOP){
	$nMad = 0;
	if ($nOP == 2){
		$nMad = 4.29;
	} else if ($nOP == 3){
		$nMad = 3.70;
	} else if ($nOP == 4){
		$nMad = 3.49;
	} else if ($nOP == 5){
		$nMad = 3.37;
	} else if ($nOP == 6){
		$nMad = 3.30;
	} else if ($nOP == 7){
		$nMad = 3.26;
	} else if ($nOP == 8){
		$nMad = 3.23;
	} else if ($nOP == 9){
		$nMad = 3.19;
	} else if ($nOP == 10){
		$nMad = 3.16;
	} else if ($nOP >= 11){
		$nMad = 3.13;
	}
	return $nMad;
}

function newMethod($w,$c){
	$nOpinion = sizeof($w);

	// Step 1:: Adjust Weighting Value (Case "Not Sure") ---
	for ($i = 0; $i < $nOpinion; $i++){
		if ($c[$i] == 0){
			if ($w[$i] >= 3){
				$w[$i] = $w[$i] - 0.5;
			} else if ($w[$i] <= 2){
				$w[$i] = $w[$i] + 0.5;
			}
		}
	}

```

Algorithm : majority-density algorithm (PHP script)

```
	// Step 2:: Detecting and Removing an outlier ---
	sort($w);
	$haveOutlier = true;

	while ($haveOutlier == true){
		$dnstMX = 1 - (((max($w) * $nOpinion) - array_sum($w)) / (5 * $nOpinion));
		$dnstMN = 1 - ((array_sum($w) - (min($w) * $nOpinion)) / (5 * $nOpinion));
		$tWeight = array();
		if ($dnstMX == $dnstMN){
			$haveOutlier = false;
		} else {
			if ($dnstMX > $dnstMN){
				$tWeight = array_slice($w,1,sizeof($w));
				$avg = array_sum($tWeight) / sizeof($tWeight);
				if (($avg - min($w) > 1.25) && ($avg - min($w) > eta(sizeof($tWeight)) * mad($tWeight))){
					$w = $tWeight;
					$nOpinion -= 1;
				} else {
					$haveOutlier = false;
				}
			} else {
				$tWeight = array_slice($w,0,sizeof($w)-1);
				$avg = array_sum($tWeight) / sizeof($tWeight);
				if ((max($w) - $avg > 1.25) && (max($w) - $avg > eta(sizeof($tWeight)) * mad($tWeight))){
					$w = $tWeight;
					$nOpinion -= 1;
				} else {
					$haveOutlier = false;
				}
			}
		}
	}

	// Step 3:: Verifying an integrated weight ---
	$iWeight = 0;
	if (1 - (mad($w) / 5) >= 0.85){
		$iWeight = array_sum($w) / sizeof($w);
	} else {
		$iWeight = 999;
	}
	return $iWeight;
}
```

BIOGRAPHY

NAME	Dechawut Wanichsan
DATE OF BIRTH	4 December 1983
PLACE OF BIRTH	Chonburi, Thailand
INSTITUTIONS ATTENDED	Phranakorn Si Ayutthaya Rajabhat University, 2001-2005, Bachelor of Science with First class honors (Computer Science)
	Mahidol University, 2005-2006, Graduate Diploma (Teaching Science)
	Chulalongkorn University, 2006-2008, Master of Science (Computer Science)
	Mahidol University, 2009-2012, Doctoral of Philosophy (Science and Technology Education)
HOME ADDRESS	15/1 Thedsaban 3 Road, Thamai, Thamai District, Chantaburi Province, 22120, Thailand
	Tel. 66-81-578-3107
	E-mail : kook260g@hotmail.com
EMPLOYMENT ADDRESS	Rambhai Barni Rajabhat University, 41 moo 5, Thachang, Muang, Chantaburi, 22000, Thailand

PUBLICATION / PRESENTATION

1. **Wanichsan, D.** & Laosinchai, P. (2010, 19-20 March). A program used to design concept of programming. In the 5th Conference on Science and Technology for Youth. Bangkok International Trade and Exhibition Center, Bangna, Bangkok, Thailand.
2. **Wanichsan, D.** & Laosinchai, P. (2010, 1-2 April). A novel program to enhance students' abilities in drawing correct flowcharts. In Proceedings of the 2nd Annual International Research Conference on Social Sciences and Humanities. Bangkok, Thailand.
3. **Wanichsan, D.** Panjaburee, P., Laosinchai, P. & Chookaew, S. (2011). A majority density approach for developing testing and diagnostic system. Lecture Notes in Computer Science, doi: 10.1007/978-3-642-23863-5_14
4. **Wanichsan, D.** Panjaburee, P., Laosinchai, P., Triampo, W. & Chookaew, S. (2012). A majority-density approach to developing testing and diagnostic systems with the cooperation of multiple experts based on an enhanced concept-effect relationship model. Expert Systems with Applications, doi: 10.1016/j.eswa.2012.01 .182

AWARD RECEIVED

The third prize at the e-learning competition of the 4th national conference on “Media knight (Media Integration : Education, Communications and Information Technology)”