

**The Assessment Agent System:
Assessing Comprehensive Understanding Based on Concept Maps**

Jianhua Liu

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Curriculum and Instruction
(Instructional Design and Technology)

Kenneth R. Potter, Chairperson
Barbara B. Lockee
David M. Moore
Jeffrey Todd Ogle

September 10, 2010
Blacksburg, Virginia

Keywords: assessment agent, software agent, learning assessment, concept map

Copyright 2010, Jianhua Liu

**The Assessment Agent System:
Assessing Comprehensive Understanding Based on Concept Maps**

Jianhua Liu

ABSTRACT

This dissertation explores the feasibility of employing software agent technology to support large-scale assessment. The research included the design, development, and evaluation of the Assessment Agent System for assessing comprehensive understanding based on concept maps.

The system was designed by following an agent-oriented software design method. The Assessment Agent System is composed of five types of software agents: instructor agent, student agent, management agent, assessment agent, and reporting agent. Each of these agents was designed to possess different capabilities. Software agents in the system, through communication and cooperation, collectively provide the functionalities of user-system interaction, user management, task authoring and management, assessment delivery, task presentation, response collection, automatic assessing with feedback, and reporting. Through the process of design, development, and evaluation of the Assessment Agent System, this study demonstrates an approach that employs an agent-oriented software design method to produce sophisticated educational software applications.

Furthermore, this study explored the concept map assessing method for the Assessment Agent System. When node terms and linking phrases are provided, assessing student concept maps can be automated by comparing student concept maps with assessment criteria, proposition by proposition. However, the usefulness of the proposition-comparing method depends heavily on the accuracy and thoroughness of the criterion propositions. Therefore, assessment criteria need to be continually refined and improved through examining student-created propositions.

Acknowledgements

I, first and foremost, want to thank my advisor Dr. Ken Potter for his invaluable advice and support through my study in the Instructional Design and Technology program. I sincerely express my thanks for my committee members. I would like to thank Dr. Barbara Lockee, Dr. Mike Moore, and Dr. Todd Ogle, for their ongoing guidance, support and encouragement.

I would also like to thank Todd Bowden for his hard work in programming the software developed for my dissertation. Finally, I appreciate all the faculty, students, staff, and others who helped me during my study at Virginia Tech.

Table of Contents

Chapter One: Introduction	1
Need for the Study	1
Purpose of the Study	4
Research Questions	4
Importance of the Study	4
Research Methodology of This Study	6
Dissertation Outline	8
Chapter Two: Literature Review.....	9
Overview of Concept Maps	9
Components of Concept Maps	10
Structures of Concept Maps	10
Concept Maps As An Assessment Tool	13
Assessment Task Formats	15
Scoring Methods	19
Computer-based Assessment Systems based on Concept Maps	24
SemNet Software	24
Mind Matrix	24
Human Performance Knowledge Mapping Tool (HPKMT).....	27
Concept Connector Tools (C-TOOLS)	30
Concept Map Assessment Tool (COMPASS)	31
Overview of Agents	34
Definition of Agent	35
Important Features of Agents	37
Agent Architectures	38
Software Agents	39
Multi-Agent Systems	41
Agent-Oriented Software Design Methodology	42
Features of Agent-Oriented Software Design	42
Prometheus: An Agent-Oriented Software Design Method	43
The Use of Software Agents in Learning Assessment	45
Summary of Literature Review	47
Chapter Three: System Design, Development, and Evaluation.....	49
The Design of the Assessment Agent System	49
Phase 1: System Specification	49
Phase 2: System Architecture Design	56
Phase 3: Detailed Design of Agents	67
The Development of the Assessment Agent System	81
Database Development	82
User Interface Development	82
The Formative Evaluation of the Assessment Agent System	84
The First One-To-One Evaluation	84
The Second One-To-One Evaluation	86
Chapter Four: Results	87
The Summative Evaluation of the Assessment Agent System	90

Participants.....	90
Materials and Settings.....	90
Data Collection	91
Procedure	91
Results of the Group Evaluation	92
Chapter Five: Discussions	94
Discussion of Research Results.....	94
System Functionality	94
System Architecture	95
Concept Map Assessment Method.....	96
Summary of Significant Contributions.....	98
References.....	99
Appendix A: Scenarios	112
Appendix B: One-To-One Evaluation	120
Appendix C: Group Evaluation.....	127
Appendix D: Student Directions for Evaluating the Assessment Agent System	132
Appendix E: Task Description	140
Appendix F: Reading Material.....	141
Appendix G: System Evaluation Questionnaire	143
Appendix H: Participant Responses to Items in the Questionnaire	146
Appendix I: Institutional Review Board Approval Memorandums.....	149
Appendix J: Informed Consent Form.....	152

List of Figures

Figure 1. An example of concept map with linear structure.....	11
Figure 2. An example of concept map with circular structure.....	11
Figure 3. An example of concept map with hub structure.....	11
Figure 4. An example of concept map with tree structure.....	12
Figure 5. An example of concept map with network structure.....	12
Figure 6. Framework considering some aspects about the nature of the mapping assessment tasks, response formats, and scoring system.....	14
Figure 7. Protocol for the Relational Scoring Method.....	23
Figure 8. Top-level flow for creating a concept mapping task.....	29
Figure 9. An agent in its environment.	37
Figure 10. Five types of agents.....	57
Figure 11. Interaction diagram of assessment task submission.....	59
Figure 12. Interaction diagram of assessment task assignment.....	60
Figure 13. Interaction diagram of task presentation.....	61
Figure 14. Interaction diagram of student map submission.....	62
Figure 15. Interaction diagram of assessment report request.....	63
Figure 16. Interaction diagram of report request.....	64
Figure 17. Overview of the assessment agent system structure.....	66
Figure 18. The instructor agent overview diagram.....	67
Figure 19. The student agent overview diagram.....	69
Figure 20. The management agent overview diagram.....	71
Figure 21. The assessment agent overview diagram.....	73
Figure 22. An expert concept map.....	75
Figure 23. A student concept map.....	77
Figure 24. The reporting agent overview diagram.....	80
Figure 25. The concept map creation web page.....	83

List of Tables

Table 1. The Major Research Process.....	7
Table 2. Relational Categories and Examples of Linking Phrases	17
Table 3. Classification of Concept Map Scoring Methods	19
Table 4. Definition of Accuracy of Propositions	21
Table 5. Six Scoring Methods.....	22
Table 6. Features of Mind Matrix	27
Table 7. Features of Human Performance Knowledge Mapping Tool (HPKMT)	30
Table 8. Features of Concept Connector Tools (C-TOOLS).....	31
Table 9. Features of the Concept Map Assessment Tool (COMPASS)	32
Table 10. Strengths and Improvement Possibilities of Mind Matrix, HPKMT, C-TOOLS, and COMPASS	34
Table 11. Characteristics of Software Agent	41
Table 12. Features of Multi-Agent Systems	42
Table 13. The Assessment Agent System Design Process	49
Table 14. Expectations for Computer-based Assessment Systems	51
Table 15. Assessment Agent System Requirements.....	52
Table 16. Functionalities and Goals of the Assessment Agent System.....	53
Table 17. Steps in the Student Map Submission Scenario.....	55
Table 18. Plans of Instructor Agents.....	68
Table 19. Plans of Student Agents.....	70
Table 20. Plans of the Management Agent.....	72
Table 21. Node Terms.....	74
Table 22. Linking Phrases.....	75
Table 23. Propositions in the Expert Concept Map	76
Table 24. Propositions in the Student Concept Map.....	77
Table 25. Assessment Results of Propositions in the Student Concept Map.....	78
Table 26. Improved Assessment Criteria.....	79
Table 27. Final Assessment Results of the Student Map.....	79
Table 28. An Example of Assessment Report	81
Table 29. Assessment results of the Student Concept Map	81
Table 30. The System Development Process.....	82
Table 31. The Developed Functionalities of the Assessment Agent System	88
Table 32. Patterns of Incorrect Proposition in Student Concept Maps.....	98

Chapter One: Introduction

Need for the Study

Large-scale assessments occur at all levels of education. In settings ranging from large college classrooms to state and national tests administered at various grade levels, multiple-choice tests are commonly used. However, multiple-choice tests are frequently criticized as a format that only assesses students' fragmented knowledge (Shepard, 1989). The limitations of standardized multiple-choice testing and dissatisfaction of traditional assessment methods present the need to explore alternative forms of achievement assessment. Concept maps are a potential means to respond to such a need (Ruiz-Primo & Shavelson, 1996).

A concept map is a diagram in which nodes represent concepts, and labeled directional links that connect nodes represent relationships between concepts. Concept mapping is a straightforward way for students to demonstrate their understanding of concepts as well as relationships among concepts with graphical representation (Novak & Gowin, 1984). Thus, concept maps offer ways to assess students' integrated conceptual understanding.

The uses of concept maps for assessing students' understanding of domain knowledge have been widely investigated (Herl, O'Neil, Chung, & Schacter, 1999; McClure et al., 1999; Novak & Gowin, 1984; Ruiz-Primo, Schultz, Li, & Shavelson, 2001; Ruiz-Primo & Shavelson, 1996; Ruiz-Primo, Shavelson, Li, & Schultz, 2001; Yin et al., 2005). The research suggests that concept maps can be a valuable tool for assessing students' knowledge structures and diagnosing their misconceptions and missing concepts (Fisher, 2000; Novak & Gowin, 1984; Yin et al., 2005).

There are several advantages in the use of concept maps for evaluating student learning on the level of understanding (Fisher, 2000). First, assessment tasks are generative, not simply responsive. The construction of concept maps requires students to precisely understand the contents and explicitly express their understandings. Second, concept mapping tasks not only assess the identification of isolated concepts and relations but also the organization of concepts and relations. Third, the process of creating concept maps facilitates students to shape their systematic and holistic views on learning contents.

Concept mapping is a viable option for evaluating student content understanding. In fact, Novak (1995) advocates utilizing concept mapping as a testing form in national achievement examinations. However, three issues should be considered if concept maps are used in large-scale assessment. These issues are validity, technical adequacy, and implementation (Tindal & Haladyna, 2002).

It is not efficient to create, evaluate and manage paper-based concept maps in large-scale assessment. In particular, manually assessing concept maps is time-consuming, subjective, inconsistent, and error-prone. These are important factors that negatively impact widely employing concept mapping in assessment (Chang, Sung, & Chen, 2001; Hoeft et al., 2003; Luckie, Batzli, Harrison, & Ebert-May, 2003; McClure, Sonak, & Suen, 1999). Therefore, it is worthwhile to develop a computer-based assessment system for assuring the validity of assessment, reducing workload and increasing the efficiency when concept maps are used for large-scale assessment.

Computer systems for large-scale assessment need to handle a large number of concurrent users with satisfactory system performance. Distributed or geographically dispersed users, including assessment developers and assessment takers, may create and

submit various tasks and responses. Moreover, the needs of real-time decision-making for scoring and providing feedback require a dynamic system. Further, systems usually require upgrades with reasonable cost and minimal system impact. The requirements of scalability, accessibility, time efficiency and cost efficiency of large-scale assessment systems call for employing appropriate technology to build high quality systems.

The use of software agent technology in education has received increasing attention from researchers. A software agent is a software entity that functions autonomously to meet its design objectives or to interact with others to achieve common goals in a particular environment. Software agents have been used to support teaching and learning in various ways (Baylor, 2002; Baylor & Kim, 2005; Johnson, Rickel, & Lester, 2000; Lin, 2005). One potential application of software agent technology is to support large-scale assessment.

Software agents have three main advantages for building large-scale computer-based assessment systems. First, agents autonomously carry out tasks to effectively reduce the workload of users (Lin, Esmahi, & Poon, 2005). Second, distributing system functionalities to various agents increases modularity and flexibility, thus making the system easier to maintain and upgrade. Third, the use of a multi-agent structure can enhance system performance in terms of computational efficiency, reliability, extensibility, robustness, responsiveness, and reuse (Sycara, 1998). Software agent technology with other computer technologies, such as databases and networks, enrich the means and increase the possibility of developing a large-scale assessment mode based on concept mapping, and provide advantages in terms of cost, utility, validity, reliability, and access (Chung & Baker, 1997).

Therefore, software agent technology is a potentially viable candidate for developing computer-based systems for large-scale assessment based on concept maps. However, there have been few attempts to systematically explore the necessary functionalities and system architecture in building software agent-oriented systems for large-scale assessment.

Purpose of the Study

The purpose of the study was to explore the feasibility of employing software agent technology to support large-scale assessment based on concept maps. The research included the design, development and evaluation of a computer-based system for assessing comprehensive understanding based on concept maps. More specifically, this study aimed to analyze and determine system functionalities, design the system architecture, and explore the method of assessing concept maps for the system.

Research Questions

This study was designed to answer the following research questions:

1. What functionalities are necessary for large-scale assessment based on concept maps in a software agent-oriented system?
2. What system architecture could be appropriate for large-scale assessment based on concept maps in a software agent-oriented system?
3. What design could provide a valid assessing method for large-scale assessment based on concept maps in a software agent-oriented system?

Importance of the Study

The importance of the study is twofold. First, it explored how agent technology can be used to support large-scale assessment. The research helps the community of

educational technology to better understand the way to use the agent-oriented paradigm in educational software design and development. Second, the developed software agent-oriented system facilitates the use of concept maps for large-scale assessment.

The agent-oriented approach has a potential to model and design sophisticated, distributed educational software systems. However, little work has been dedicated to the systematic exploration of designing, developing, and evaluating agent-oriented educational software systems. A major motivation of this study was to explore the use of agent technology to support large-scale assessment. The design, development, and evaluation of the system demonstrated an agent-oriented approach for the educational technology community interested in designing and developing agent-oriented educational applications. Furthermore, the functionalities and architecture of the system can be used as references for designing and developing other agent-oriented systems for large-scale assessment.

The software agent-oriented system facilitates concept map construction, modification, submission, scoring, and result reporting in large-scale assessment. More importantly, the assessment system provides students with opportunities to visualize their thinking and model their knowledge, and it provides an authentic means to assess their performance on knowledge integration. Furthermore, assessing students' complete and accurate understanding of curricula contents with concept mapping supports the development of meaningful learning, and provides a reliable means to measure student achievement.

Research Methodology of This Study

A growing paradigm of educational inquiry is designing, developing, and evaluating curricula, programs, software, tools, models, and school organizations to make research more directly relate to and impact educational practice (Edelson, 2002; Richey & Klein, 2007). Sharing similar goals and approaches but with various focuses, this type of research has been labeled as design research (van den Akker, Cravemeijer, McKenney, & Nieveen, 2006) or developmental research (Richey, Klein, & Nelson, 2004). More recently, Richey and Klein (2007) coined the term “design and development research” to refer to this type of study, and defined it as:

The systematic study of design, development and evaluation processes with the aim of establishing an empirical basis for the creation of instructional and non-instructional products and tools and new or enhanced models that govern their development. (p. 1)

The iterative cycle of analysis, design, development, and evaluation provides researchers with opportunities to advance their understanding and gain knowledge for improving educational practice. To effectively generate research outcomes from the design and development process, Edelson (2002) suggested that the study process should be research-driven as well as systematically documented, employ formative evaluation, and the learned lessons should be generalized to produce theories for guiding educational practice in broad contexts.

Richey and Klein (2007) classified design and development research projects into two categories: (a) product and tool research, and (b) model research. Product and tool research pertains to studies of analyzing, designing, developing, and evaluating

instructional or non-instructional products, programs, or tools. Model research involves constructing, validating, and using design and development models or processes. This study adopted the product and tool research approach. The major research process is presented in Table 1.

Table 1

The Major Research Process

Phase	Research activity
Literature Review	Reviewed research of agents, and identified features and characteristics of agents that should be addressed in the system design
	Investigated agent-oriented system design techniques and selected one appropriate method for designing the assessment system
	Reviewed and analyzed the study of concept maps as an assessment tool to decide assessment task format and scoring method for the system
	Reviewed and analyzed existing computer-based assessment systems based on concept maps to assist decision making on the system design and development
Design	Analyzed requirements for computer-based assessment systems
	Determined system functionalities
	Designed system architecture
	Designed each component of the system
Development	Developed database
	Developed user interface
	Developed system functionalities
Evaluation	Conducted one-to-one evaluation
	Identified problems and improved the system
	Conducted group evaluation
	Analyzed data and summarized results

Dissertation Outline

Chapter Two reviews concept maps, agents, and issues that relate to the design and development of a software agent-oriented system for large-scale assessment based on concept maps. Chapter Three describes the system design, development and evaluation. Chapter Four reports results of the study. Chapter Five presents a discussion of research results.

Chapter Two: Literature Review

The literature review includes major concepts and issues that relate to the design and development of a software agent-oriented system for large-scale assessment based on concept maps. This chapter begins with an overview of concept maps. The subsequent sections examine concept maps as an assessment tool and analyze five representative computer-based assessment systems based on concept maps. Chapter two continues with a general discussion of agents, including definitions, important features, and architectures of agents. The following sections review software agents, multi-agent systems, agent-oriented software design methodology, and employing software agents in learning assessment. The chapter closes with a summary of the literature review.

Overview of Concept Maps

Several types of node-link diagrams have been developed and used as tools for organizing, representing, and communicating information. These tools include concept maps (Novak & Gowin, 1984), semantic maps (Cronin, Meadows, & Sinatra, 1990), semantic networks (Fisher, 1990), feedback loop diagrams (Hyerle, 1996, 2000), and causal chains (Clarke, 1990). Among these node-link diagrams, concept maps are a well-studied tool for teaching, learning, and assessment (Cañas et al., 2003; O'Donnell, Dansereau, & Hall, 2002; Nesbit & Adesope, 2006; Novak, 1990, 1998; Novak & Gowin, 1984; Ruiz-Primo & Shavelson, 1996). A concept is defined as “a perceived regularity in events or objects, or records of events or objects, designated by a label” (Novak, 1998, p. 22). A concept map is a diagram in which nodes represent concepts, and labeled directional links that connect nodes represent relationships between concepts.

Some researchers (Chung et al., 2006; Herl, O'Neil, Chung, & Schacter, 1999; O'Donnell, Dansereau, & Hall, 2002) have used the term knowledge map to refer to this type of node-link diagram. In this study, the term concept map is used as the common label for this type of node-link diagram, including knowledge maps.

Components of Concept Maps

The basic components in a concept map are nodes and links (Novak & Gowin, 1984). A node is composed of a node symbol and a node term. A node symbol is a graphic shape, such as a circle, oval, or rectangle. A node term labels a node. The combination of a node symbol and a node term represents a concept. A link includes a linking line and a linking phrase. A linking line connects two nodes to indicate that there is a relationship between the two concepts. The arrowhead of the linking line indicates the direction of the relationship between the two concepts. A linking phrase specifies the relationship. A node-link-node combination is called a proposition, which is the basic unit of a meaningful statement in a concept map (Ruiz-Primo & Shavelson, 1996).

Structures of Concept Maps

The arrangement of nodes and links determines the structure of a concept map. Novak and colleagues (Novak, 1998; Novak & Gowin, 1984) emphasize hierarchical structure in concept maps. In such a layout, nodes that represent general concepts are organized in upper levels, and those that represent specific concepts are arranged in lower levels in a concept map. Cross-links are used to connect nodes that have relationships but are in different branches of the hierarchical structure.

Although the hierarchical structure is appropriate for representing contents in many domains, the restriction of representation to hierarchy is not necessary, and the

structure of a concept map may depend on the content that it represents (Ruiz-Primo & Shavelson, 1996; Ruiz-Primo, Shavelson, & Schultz, 1997). Yin et al. (2005) identified five primary types of concept map structures: linear, circular, hub, tree, and network. Figure 1, Figure 2, Figure 3, Figure 4, and Figure 5 illustrate the five types of concept maps.

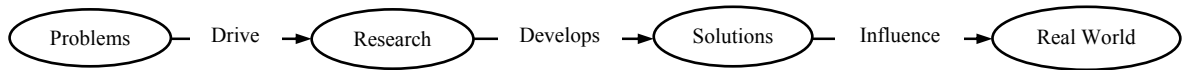


Figure 1. An example of concept map with linear structure.

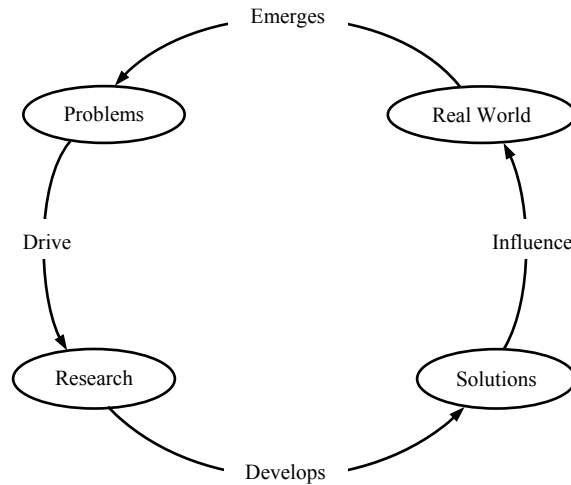


Figure 2. An example of concept map with circular structure.

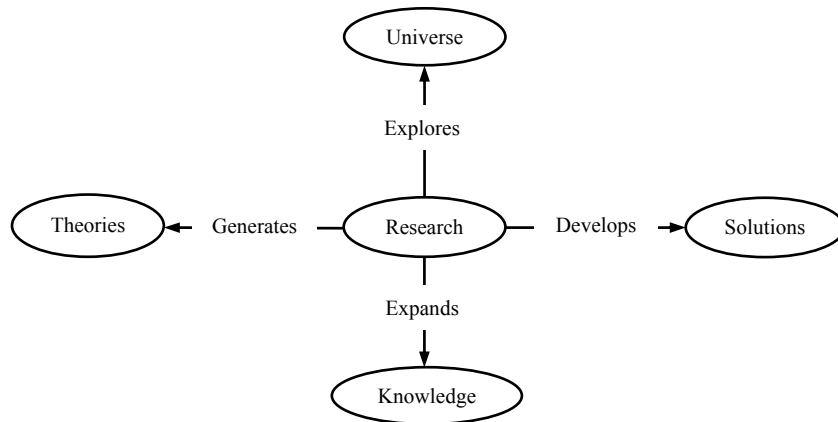


Figure 3. An example of concept map with hub structure.

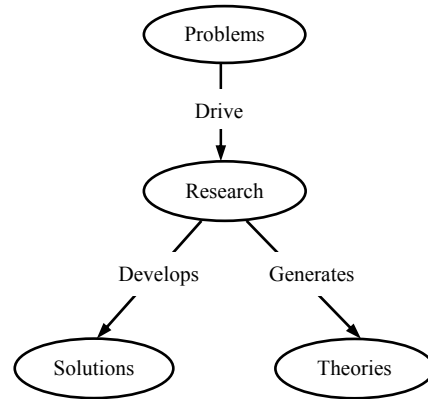


Figure 4. An example of concept map with tree structure.

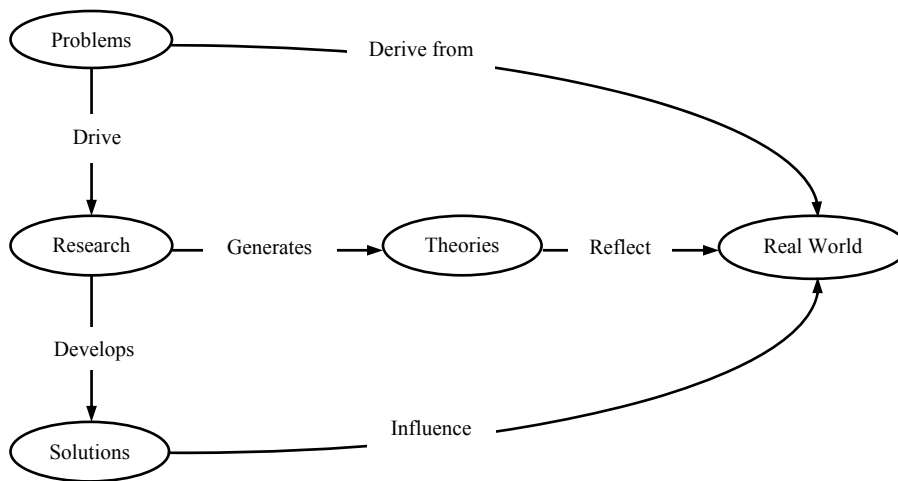


Figure 5. An example of concept map with network structure.

Concept maps can be used as a tool for externalizing thinking, explicating understanding, or representing knowledge. Educational researchers have widely investigated and examined the employment of concept maps in a variety of areas of education (Cañas et al., 2003; O'Donnell, Dansereau, & Hall, 2002; Nesbit & Adesope, 2006; Novak, 1990, 1998; Novak & Gowin, 1984; Ruiz-Primo & Shavelson, 1996). More specifically, concept maps have been used as (a) a curriculum planning tool; (b) an instructional support tool for presenting learning contents; (c) a learning tool for supporting meaningful learning; and (d) an assessment tool for identifying

misconceptions and measuring understanding. For the purpose of this study, the following section focuses on reviewing the use of concept maps as an assessment tool.

Concept Maps As An Assessment Tool

Burkhardt and Pead (2003) identified four key aspects of assessment design: task presentation, student working, student response, and marking. Ruiz-Primo and Shavelson (1996) suggested that a concept map assessment is composed of a task, a response format, and a scoring system. For investigating the characteristics of concept map assessment techniques, Ruiz-Primo (2004) proposed a framework that illustrates the variation of tasks, formats, and scoring methods in concept map assessment (see Figure 6).

Assessment Components		Concept map Components											
		Node Terms			Linking Lines		Linking Phrases			Map Structure			
Response Required	Task												
Construct the Map	What is Provided	Not Provided	Provided		Not Provided	Provided		Not Provided	Provided		Not Provided	Provided	
	How Much is Provided	Few Provided			Few Provided		Few Provided			Partially Provided			
		All Provided			All Provided		All Provided			Completely Provided			
	Relevance of What is Provided	Key Terms			Very Relevant		Deep Phrases			Very Relevant			
Related but not Key Terms			Not Relevant		Superficial Phrases			Related but not Relevant					
Fill-in the Map	What is Required	Few Terms	Provide Terms	Key Terms	Few Lines	Most Relevant Lines	Few Phrases	Provide Phrases	Deep Phrases	Free Structure			
		All Terms	Select Terms	Related but not Key Terms	All Appropriate Lines	All Suitable Lines	All Phrases	Select Phrases	Superficial Phrases	Specific Structure			
	Scoring System												
	Use of a Criterion Map	Not Used	Used		Not Used	Used		Not Used	Used		Not Used	Used	
	What it is Scored	Correctness Relevance Quantity	Correctness Relevance Quantity Similarity	Correctness Relevance Quantity	Correctness Relevance Quantity Similarity	Correctness Quality Quantity	Correctness Quality Quantity Similarity	Correctness Quality Quantity Similarity	Correctness Quality Quantity Similarity	Complexity Type	Complexity Type Similarity		

Figure 6. Framework considering some aspects about the nature of the mapping assessment tasks, response formats, and scoring system

(adapted from Ruiz-Primo, 2004).

Assessment Task Formats

A concept map assessment task may assume a variety of formats. Some examples of concept mapping task format are fill-in-nodes, fill-in-links, construct-a-map with provided node terms and linking phrases, construct-a-map with provided node terms and suggested structure, construct-a-map with provided node terms, and construct-a-map freely (Ruiz-Primo, 2004).

However, different mapping techniques may provide different information about students' knowledge structures. Ruiz-Primo and colleagues (Ruiz-Primo, Shavelson, Li, & Schultz, 2001) examined the effects of fill-in-the-map and construct-a-map-from-scratch techniques on assessing students' connected understanding. In the fill-in-the-map condition, students were asked to fill blank nodes or blank linking phrases in given skeleton maps. Lists of concept terms and linking phrases were provided. In the construct-a-map-from-scratch condition, students were asked to construct a map with provided concept terms. Linking phrases and map structure were not provided. Results indicate that the construct-a-map-from-scratch technique is more effective than the fill-in-the-map technique for revealing the differences among students' knowledge structures.

Chung, Baker, and Cheak (2002) suggest that it is more meaningful to students if they can freely generate and revise concepts and relationships in the process of constructing concept maps. In another study, Yin et al. (2005) investigated “construct-a-map with created linking phrases (C)” and “construct-a-map with selected linking phrases (S)” techniques by comparing map products and map construction processes. In the construct-a-map with created linking phrases (C) condition, students constructed maps with provided concept terms, but they were asked to generate linking phrases by

themselves. In the construct-a-map with selected linking phrases (S), students constructed maps with provided concept terms and linking phrases. Student generated maps were analyzed and compared in terms of total accuracy score, individual proposition scores, proposition choice, map structure complexity, proposition generation rate, and proposition generation procedures.

Yin et al. (2005) conclude that the C and S mapping techniques have different effects on assessing students' knowledge structures. The C technique is more effective than the S technique for capturing students' partial knowledge and revealing their misunderstandings. However, the S technique is more efficient than the C technique for scoring maps. Yin et al. further suggest that the C technique is a promising approach for formative assessment and the S technique is more appropriate for large-scale assessment.

Herl et al. (1999) dichotomized concept map construction formats into two categories: open and closed. In the open concept map construction mode, students create nodes and/or links and build their maps. In the closed concept map construction mode, students build their maps with provided concept terms and linking words. Lambiotte et al. (1989) suggested that a set of systematically named linking phrases provide a guide for map developers to search for and organize relevant information. Researchers have identified relational categories and possible linking phrases (Collins & Quillian, 1972; Dansereau & Holley, 1982; Lambiotte et al., 1989). The relational categories and examples of linking phrases in the literature are presented in Table 2. However, it is impossible to create a set of linking phrases that covers all relationships.

Table 2

Relational Categories and Examples of Linking Phrases

Reference	Relational category	Example of linking phrase
Dansereau & Holley (1982)	Hierarchy	Part of Segment of Portion of
		Type of Category Example of Kind of
	Chain	Leads to Results in Causes Is a tool of Produces
		Similar to Analogous to Like Corresponds to
	Cluster	Has Characterized Feature Property Trait Aspect Attribute
		Indicates Illustrates Demonstrates Supports Documents Proof of Confirms Evidence of
	Evidence	

Table 2 (continued)

Relational Categories and Examples of Linking Phrases

Reference	Relational category	Example of linking phrase
Collins & Quillian (1972)	Superset (Superordinate)	Is a
		Is a member of
	Subset	Consists of
		Contains
	Similarity	Is like
		Is not like
	Part	Part of
	Proximity (Adjacency)	Is adjacent to
		Is next to
Lambiotte, Dansereau, Cross, & Reynolds (1989)	Consequence (Causality)	Leads to
		Influences
		Causes
	Precedence	Prior to
	Parent	Is belong to
	Dynamic	Influences
		Next
		Leads to
	Static	Type
		Part
		Characteristic
	Instructional	Analogy
		Example
		Comment

Based on the review of literature about assessment task format, to strike a balance between effectively revealing students' knowledge structures and efficiently assessing student concept maps, one viable task format for large-scale assessment is to provide students with node terms and linking phrases, which include some node terms and linking

phrases that serve as distracters. In such an assessment mode, students are required to identify and select node terms as well as linking phrases from the given lists, and build their concept maps for presenting their understandings or knowledge.

Scoring Methods

Ruiz-Primo and Shavelson (1996) classified concept map scoring strategies into three categories: (a) scoring a student map against a list of rubrics or criteria, (b) comparing a student map with a criterion map, and (c) using a combination of both methods. Chung et al. (2006) further categorized concept map scoring methods in two dimensions: referent-free versus referent-based, configural versus semantic. The classification of concept map scoring methods of Chung and colleagues is presented in Table 3.

Table 3

Classification of Concept Map Scoring Methods (adapted from Chung et al., 2006, p. 4)

	Configural	Semantic
Referent-free	Explicitly scores a map or elements of a map on its structural aspect (e.g., considering degree of hierarchical organization).	Explicitly scores a map or elements of a map on its semantic aspect (e.g., scoring quality of propositions).
Referent-based	Compares the network structure of a student's map and the referent map. Does not take into account the meaning of the relationships.	Compares the semantic structure of a student's map and the referent map (e.g., proposition-by-proposition comparison between a student's map and an expert's map). Ignores the configural aspects of the network.

In referent-free methods, student maps are evaluated by using a list of rubrics or criteria. The traditional referent-free scoring system was developed by Novak and Gowin

(1984). In this method, a concept map is scored by counting the number of valid propositions (1 point each), levels of hierarchy (5 points for each level), number of cross-links (10 points for each valid cross-link), and number of specific examples (1 point for each example). The scoring rules in this method address both semantic and structural aspects of concept maps.

Another scheme of referent-free scoring approach is assessing the accuracy of each proposition in a concept map against a proposition inventory (Ruiz-Primo, Schultz, & Shavelson, 1997; Ruiz-Primo, Shavelson, & Schultz, 1997). This method is suitable for the assessment task format where students construct their concept maps with provided node terms and linking phrases. The proposition inventory includes all possible concept-relationship-concept combinations of the terms and phrases given to students. Each concept-relationship-concept combination is assigned a score on a 5-point scale: 0 for “inaccuracy/invalid” proposition, 1 for “don’t care” proposition, 2 for “poor” proposition, 3 for “good” proposition, and 4 for “excellent” proposition. The definition of accuracy of propositions is listed in Table 4. When node term and linking phrase lists include many items, the proposition inventory will include many possible concept-relationship-concept combinations, and creating such an inventory becomes time consuming.

Table 4

Definition of Accuracy of Propositions (adapted from Ruiz-Primo, Shavelson, & Schultz, 1997)

Accuracy of Proposition	Definition
Excellent	Outstanding proposition. Complete and correct. It shows a deep understanding of the relation between the two concepts.
Good	Complete and correct proposition. It shows a good understanding of the relation between the two concepts.
Poor	Incomplete but correct proposition. It shows partial understanding of the relation between the two concepts.
Don't Care	Although valid, the proposition does not show understanding between the two concepts.
Inaccurate/Invalid	Incorrect proposition.

In referent-based scoring methods, expert maps are used as criteria for assessing student maps. In the study of Herl, Baker, and Niemi (1996), the quality of student concept maps were assessed by comparing them with expert maps, including both proposition and structure comparison. In some cases, student concept maps are assessed by matching expert maps proposition by proposition, without considering the concept map structure (Klein, Chung, Osmundson, Herl, & O'Neil, 2002). This is more popular when computers are used to assess concept maps since comparing structure is more difficult to automate.

The use of expert maps reduces arbitrariness in assessing student concept maps, and increases reliability and validity of assessment results (Klein et al., 2002). McClure et al. (1999) examined the reliability and validity of six concept map scoring methods. The six scoring methods include (a) holistic, (b) holistic with master map, (c) relational, (d)

relational with master map, (e) structural, and (f) structural with master map. The procedures of the six scoring methods are listed in Table 5.

Table 5

Six Scoring Methods

Scoring Method	Procedure
Holistic	The rater examined the student map and judged the student's overall understanding of the concepts represented by the map. Each map was assigned a score on a scale from 1 to 10, which was determined based on the rater's subjective judgment.
Holistic with master map	The rater examined the student map and judged the student's overall understanding of the concepts represented by the map with an expert map as reference. Each map was assigned a score on a scale from 1 to 10, which was determined based on the rater's subjective judgment with an expert map as a guide.
Relational	The rater evaluated the student map by following the procedure illustrated in Figure 7. This method focused on assessing the separate propositions in student maps.
Relational with master map	The rater evaluated the student map by following the procedure illustrated in Figure 7 and used an expert map as a guide. This method focused on assessing the separate propositions in student maps.
Structural	The rater evaluated the student map by counting the valid number of propositions, cross-links, examples and levels of hierarchy. This method was adapted from the scoring method developed by Novak and Gowin (1984).
Structural with master map	The rater evaluated the student map by counting the valid number of propositions, cross-links, examples as well as levels of hierarchy, and used an expert map as a guide.

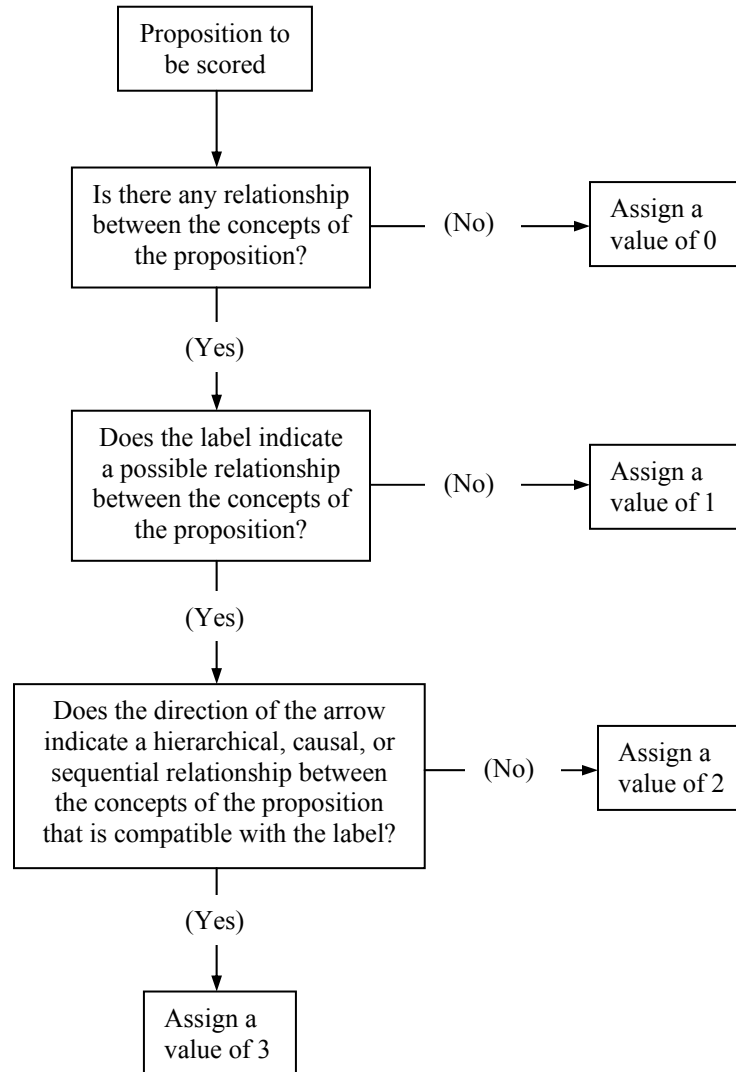


Figure 7. Protocol for the Relational Scoring Method (McClure, Sonak, & Suen, 1999, p. 482).

In the study of McClure et al. (1999), each student created a map with a set of 20 concepts. No linking phrases were provided. Student maps were assessed with the six scoring methods. Results indicated that the relational with master map method yielded the most reliable scores. All methods, except the structural with master map, are valid. The findings of another research (Herl et al., 1999) confirm that the relational with master map scoring method is reliable and valid.

Computer-based Assessment Systems based on Concept Maps

Researchers in a variety of fields have strived to computerize the process of concept map construction and assessment (Anohina, Graudina, & Grundspenkis, 2007; Chang, Sung, & Chen, 2001; Chung & Baker, 1997; Fisher, 1990, 2000; Tsai, Lin, & Yuan, 2001). Several concept map assessment systems have been developed (Anohina & Grundspenkis, 2006, 2007; Chung et al., 2006; Dennis, Stamboltsian, & Huang, 2002; Gouli, Gogoulou, Papanikolaou, & Grigoriadou, 2004; Luckie, Batzli, Harrison, & Ebert-May, 2003). This section reviews five representative computer-based assessment tools, which use node-link diagram as an assessment task format.

SemNet Software

SemNet software was designed as a general-purpose tool for creating knowledge representations in the form of semantic network (Fisher, 1990; Fisher et al., 1990). The SemNet software does not provide an automatic scoring function, but it possesses strong statistical functionality. Fisher (2000) reported the power of using SemNet software to visualize student thinking and assess student knowledge structures. With the assistance of statistical report and display functions of SemNet software, student-produced semantic networks are manually evaluated by (a) examining the number of concepts, relations, and instances; (b) assessing the quality of concept descriptions and relations; (c) reviewing the organization of nets, and (d) examining key concepts included in nets. The SemNet software is an impressive example of exploring the computerizing node-link diagram assessment.

Mind Matrix

Mind Matrix is a networked computer-based assessment system, which intends to

integrate instruction, learning and assessment (Dennis et al., 2002). The Mind Matrix system includes client component and server component. The client side is a Java application that runs on users' computers with two optional modes: student mode and editor mode. In the student mode, students use the Mind Matrix client Java application to access concept mapping tasks, create their maps, submit their maps to the Mind Matrix server, and receive feedback and scores on their performance.

The Mind Matrix does not support assessment takers to create concept maps by typing in their concept terms and relation words. Students are provided with a fixed set of concepts and relationships. Students construct their maps by selecting concepts and connecting them with relationships to present their understanding of contents.

In the editor mode, instructors use a variety of tools to create and set up concept mapping tasks. The Mind Matrix editor not only supports instructors to create concept map tasks but also provides students with opportunities of learning through creating concept map tasks.

When creating a mapping task with Mind Matrix client application, the user first specifies concepts and relationships based on instructional contents and objectives. Concepts and relationships can either be typed into the client application or imported from a plain text file. The assessment developer places concepts onto the mapping canvas and connects them with relationship links to build a criterion map.

There is a map property interface in the Mind Matrix client application for setting up parameters of mapping tasks. Concepts in the criterion map can be added to hints, which are information related to the concepts and serve as a means of performance support. The hints can be texts, URLs, or concept maps.

Staging is a pedagogical strategy to divide a complex mapping task into several manageable steps in Mind Matrix. When a student finishes one stage of the mapping task, the student map is scored and feedback is provided. Moving from one stage to the next stage can be configured with four selectable policies: user's decision, time expiration, quality level, and completion level.

A student map is scored by comparing it with a criterion map. Consistencies and discrepancies between the student map and the criterion map are highlighted with different colors to help students verify their correct concepts and diagnose their misconceptions.

The highlight of Mind Matrix system is the function of reporting students' performance to teachers, which serves as a means for assisting teachers to identify students' understanding of the content, thereby providing individualized feedback and adjusting their instruction to meet students' needs.

The Mind Matrix server collects student task performance data and can generate three types of report: statistic report, proposition report, and student report (Dennis et al., 2002; "Mind Matrix editor authoring manual," 2002). A statistical report provides information about the number of students who took a mapping task, the minimum and maximum scores that students achieved, as well as the mean and standard deviation of student scores. In a proposition report, the number of students who correctly and incorrectly created each proposition is listed on a proposition-by-proposition basis. The third type of report is student report, which offers summarized and detailed information about each individual student's performance on a mapping task. The information includes the number of proposition the student created, the number of misconceptions in a student

map, the student score and score percent, as well as student performance on each proposition by comparing it with the criterion map. The features of Mind Matrix are presented in Table 6.

Table 6

Features of Mind Matrix

Features	Mind Matrix
User interface intuitiveness	Buttons are easy to recognize
Operation and map organization	Clicking and dragging
Node and link format	Rectangles and lines with different colors Linking line curve can be adjusted
Mapping work area	Large
Assessment approach	Automatic
Automatic scoring method	Criterion-map-based comparison
Feedback	Graphical indication Simple Comments
Reporting function	Statistic, proposition, and student report
Task format	Staged present tasks Construct maps based on node term and linking phrase lists
Authoring function	Create node terms, linking phrases, criterion map, and content-specific hints
System architecture	Client Java application for task authoring Client Java application for students responding to tasks
Network accessibility	Client application avoids firewall blocking

Human Performance Knowledge Mapping Tool (HPKMT)

Human Performance Knowledge Mapping Tool (HPKMT) is a computer-based assessment system, which aims to evaluate learners' content understanding with concept maps (Chung et al., 2006). The functionality of HPKMT is derived from the literature

review of concept maps as an assessment tool, existing software packages for creating concept maps, and the authors' previous experiences with online concept mapping assessment. Chung et al. (2006) listed lessons learned from their previous research on using online assessment with a variety of assessment takers, ranging from pupils to adult learners. They conclude that a computer-based assessment system should: have a simple interface to avoid confusing users and save system training time, have strong assessment task authoring capability, avoid conflicting with network security constraints, have automatic scoring capability, and incorporate multiple scoring approaches.

The developers of HPKMT designed a simple and intuitive user interface to reduce training and irrelevant variance of assessment tasks (Chung et al., 2006). The user interface is composed of three areas and adopts a top-left-right frame. The top area houses menus and icons that provide functions for opening, editing, saving and printing files, connecting nodes, and zooming in or zooming out of the concept map. The bottom left area is a symbol space where concept symbols are listed. The bottom right of the screen is a mapping canvas area, in which users create their maps. When an assessment taker creates concept maps in HPKMT, he or she only uses a mouse to click and drag the concept icon onto the canvas, draw straight lines to connect nodes, and select node terms as well as linking phrases from pop-up lists.

The HPKMT system includes a client application and a server. To avoid blocking by firewalls, the client side was developed as a full-featured application with Java Web Start technology. Users can launch the HPKMT client application with any web browser. Moreover, the Java Web Start technology enables the client side application to be downloaded, installed, and upgraded automatically.

In HPKMT, the concept map automatic scoring engine is an independent component, which is separated from databases in which concept map data are stored (Chung et al., 2006). Such a design makes the system easier to maintain and upgrade. There are two referent-based scoring methods in HPKMT. One is exact proposition matching, which is based on a concept map scoring method developed by Herl, Baker and Niemi (1996). Another is synonym-based proposition matching. In a referent map, a concept or a relationship does not only associate with one term, but with a set of synonymic terms. Concepts and relationships in a learner's map are compared with a corresponding set of synonymic terms in the expert's map to determine the correctness.

The HPKMT system includes an assessment task-authoring prototype (Chung, Baker, & Cheak, 2002; Chung et al., 2006). Using a web browser, an assessment developer interacts with a server to specify a mapping task by creating or modifying node terms and linking phrases, assign the task to assessment takers, and make the task to be accessible by authorized users. Figure 8 illustrates the process of task authoring. The features of HPKMT are presented in Table 7.

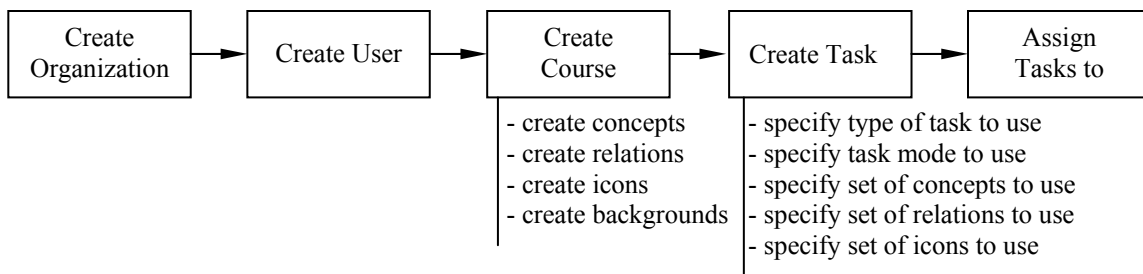


Figure 8. Top-level flow for creating a concept mapping task (Chung et al., 2006, p. 16).

Table 7

Features of Human Performance Knowledge Mapping Tool (HPKMT)

Features	HPKMT
User interface intuitiveness	Buttons are easy to recognize
Operation and map organization	Dragging and dropping Double-clicking or right-clicking
Node and link format	Icon set and color can be selected
Mapping work area	Large
Assessment approach	Automatic
Automatic scoring method	Criterion-map-based exact proposition matching Synonym-based proposition matching
Feedback	Scores and simple comments
Reporting function	No
Task format	Construct concept maps by selecting node terms and linking phrases from lists
Authoring function	Create organizations, users, courses, tasks, and assign tasks to users
System architecture	Full-featured client application for assessment taker Web-based authoring interface Database and scoring engine are separated
Network accessibility	Client application avoids firewall blocking

Concept Connector Tools (C-TOOLS)

Concept Connector Tools (C-TOOLS) is a large-scale online concept map assessment environment with automatic scoring and feedback functionality (Luckie et al., 2003). The C-TOOLS was primarily designed for formative assessment in undergraduate science teaching and learning. This system provides immediate formative feedback, and it encourages students to reflect on their learning and share their thinking with their peers.

In C-TOOLS, the automatic assessment method adopts the concept map scoring system developed by Novak and Gowin (1984). Student maps are scored by counting

valid nodes, links, cross-links and by assessing hierarchical structure. Assessment criteria are based on “expert” faculty’s maps and inputs. The features of C-TOOLS are presented in Table 8.

Table 8

Features of Concept Connector Tools (C-TOOLS)

Features	C-TOOLS
User interface intuitiveness	Menus are easy to recognize
Operation and map organization	Use menus to create and revise maps No dragging and dropping function
Node and link format	Rectangles and lines with different colors
Mapping work area	Small
Assessment approach	Automatic Instructor
Automatic scoring method	Rubric-based Counts valid nodes, links, and cross links and assess hierarchical structure
Feedback	Numeric score Graphical indication and comments
Reporting function	Student performance on a task
Task format	Freely creating concepts and links
Authoring function	Instructor interface allows instructors to view student maps and configure feedback on them
System architecture	Web-based Java applet map drawing tool Cross-linking databases A Robograder Servlets for interactivity Student interface and instructor interface
Network accessibility	Java applet may be blocked by firewalls

Concept Map Assessment Tool (COMPASS)

The Concept Map Assessment Tool (COMPASS) is a web-enabled “learning through assessment” system (Gouli et al., 2004). COMPASS supports a variety of

concept mapping activities, such as evaluating a given map, completing a given map, extending a given map, and freely constructing a map. One strong point of the COMPASS system is that it diagnoses the submitted map and provides adaptive feedback to assist the learner to achieve learning objectives. The features of COMPASS are presented in Table 9.

Table 9

Features of the Concept Map Assessment Tool (COMPASS)

Features	COMPASS
User interface intuitiveness	Menus, buttons, and symbol palette are easy to recognize
Operation and map organization	Clicking and dragging
Node and link format	Rectangles, ovals and lines with different colors
Mapping work area	Large
Assessment approach	Automatic, instructor, or peer
Automatic scoring method	Criterion-map-based similarity comparison
Feedback	Text, graphical, and dialogue feedback Adapted individual feedback
Reporting function	No
Task format	Free construction of a map Evaluation/correction of a given map Extension/completion of a given map Node terms or linking phrases are either freely created or selected from lists
Authoring function	No
System architecture	Web-enabled client application Adopts the classical intelligent tutoring system architecture Six modules: interaction monitoring module, diagnostic module, adaptive feedback generation module, presentation module, domain knowledge, and learner model
Network accessibility	Client application avoids firewall blocking

The tools reviewed in this section provide a variety of functions to computerize the process of using concept maps for assessment. The strengths and improvement possibilities of Mind Matrix, HPKMT, C-TOOLS, and COMPASS are summarized and presented in Table 10. These tools were primarily developed for formative assessment and aim to promote learning and improve instruction by means of concept mapping. However, an improved computer-based system for large-scale assessment based on concept maps should support user management and task authoring, provide students with an easy usability to work on and respond to tasks, possess reliable and valid automatic scoring abilities, and report assessment results. No existing concept map assessment system fully addresses the needs of large-scale assessment based on concept maps.

Software agent technology enriches the means and increases the possibility of developing a satisfactory tool to support large-scale assessment based on concept maps. The following sections review agent related issues.

Table 10

Strengths and Improvement Possibilities of Mind Matrix, HPKMT, C-TOOLS, and COMPASS

System	Strength	Improvement Possibility
Mind Matrix	Intuitive user interface Easy operation Staged task presentation Rich reporting functionality Strong authoring functionality	Could provide distractive node terms and linking phrases
HPKMT	Intuitive user interface Easy operation Synonym-based proposition matching for assessing student maps	Could develop student performance reporting functionality
C-TOOLS	Rich formats of feedback on student maps Freely create node terms and linking phrases	Could improve student interface
COMPASS	Provides adaptive informative, tutoring, and reflective feedback on student maps Supports various concept mapping activities, including constructing maps, completing, extending, or evaluating given maps	Could develop authoring and student performance reporting functionalities

Overview of Agents

The manifestation of the idea of constructing agents in computer systems can be traced back as early as the 1950s. Kay (1984) provided a sketch of the origination of agent technology:

The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology. They had in view a system that, when given a goal, could carry out the details of the appropriate computer operations

and could ask for and receive advice, offered in human terms, when it was stuck.

An agent would be a ‘soft robot’ living and doing its business within the computer’s world. (Kay, 1984, p. 58)

Since the 1990s, agent technology has been a rapidly growing research area (Müller, 1996; Wooldridge, 2002). There is widespread and growing interest in the use of agents for building advanced software systems. In general, agent technology is appropriate for applications in which (a) agents capable of flexible autonomous action are the only solution in open, dynamic, uncertain, or complex environments, (b) multiple agents work together to deal with the distribution of data, control, expertise, or resources, (c) agents provide a natural metaphor for modeling applications and delivering system functionality, or (d) agents enable a number of legacy systems to interwork (Jennings & Wooldridge, 1998a; Wooldridge, 2002).

Agent-oriented systems have been developed for applications in business, education, entertainment, and industry (Jennings & Wooldridge, 1998b; Kirn, Herzog, Lockemann, & Spaniol, 2006; Lin, 2005; Wooldridge, 2002). The range of applications for which agents are employed includes information retrieval and management, industrial systems management, business process management, electronic commerce, human-computer interfaces, distributed education, and computer-based virtual environments.

Definition of Agent

In Merriam-Webster’s collegiate dictionary (2004), there are numerous definitions for an agent. An agent is defined as something that produces an effect, a person who acts for another person or a government, an instrument that is used to achieve a result, and so forth. In particular, an agent is defined as “a computer application

designed to automate certain tasks (as gathering information online)” (Merriam-Webster’s collegiate dictionary, 2004, p. 24).

Researchers have proposed a variety of agent definitions based on their studies and research work (d’Inverno & Luck, 2004; Franklin & Graesser, 1997). Some of the more notable attempts to define agents are discussed in this section.

Russell and Norvig (1995, 2003, 2010) referred to an agent as anything that can perceive its environment through sensors and act upon that environment through actuators. In this definition, if the notions of environment, sensing and acting are not restricted, any computer program can be an agent (Franklin & Graesser, 1997). Nwana (1996) defined an agent as a software and/or hardware component that can act on behalf of its user to accomplish tasks. Nwana’s definition restricts an agent to a computational entity to which tasks can be delegated.

Franklin and Graesser (1997) regarded autonomy as central to the notion of agents, and provided a definition of an autonomous agent:

An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future. (Franklin & Graesser, 1997, p. 25)

More recently, Wooldridge (2002) defined an agent as “a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its design objectives” (p. 15). Based on Wooldridge’s definition, an abstract view of an agent is illustrated in Figure 9. Franklin and Graesser (1997) and Wooldridge (2002) emphasized the property of autonomy and referred to an

agent as a system that is situated in an environment and autonomously senses and acts upon it.

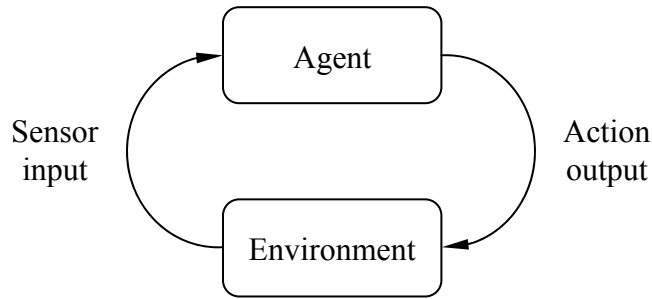


Figure 9. An agent in its environment. The agent takes sensory input from the environment, and produces as output actions that affect it. The interaction is usually an ongoing, non-terminating one. Adapted from Wooldridge, 2002, p.16.

As agent technology is a fairly young area of research, there is not a universally accepted definition of agent in the research community. In this dissertation an agent is defined as a computational entity that functions autonomously to meet its design objectives or to interact with others to achieve common goals in a particular environment.

Important Features of Agents

Many qualities have been proposed by researchers as being necessary for agenthood. Franklin and Graesser (1997) summarized properties of agents based on their review of eleven definitions of agents. These properties include reactivity, autonomy, goal-directedness, temporal continuity, communication, learning, mobility, flexibility, and character. These agent properties illustrate the diversity of agent design as well as the range of agent applications. Autonomy is regarded as a central and necessary property of computational agents (Franklin & Graesser, 1997; Müller, 1998; Wooldridge, 2002). d'Inverno and Luck (2004) maintained that “autonomy is not simply one of many

possible characterising properties or qualities, but is foundational in its significance, and serves as a platform on which other properties can be imposed, just as agency” (p.16).

When considering an agent to be intelligent, Wooldridge (2002) suggested that a computational agent is intelligent if it is capable of autonomous action to meet its design objectives by its pro-activeness, reactive action, and social interaction in groups with other agents and/or human users.

Agent Architectures

The architecture of an agent describes its modules as well as capabilities, and how these components interact and operate together (Müller, 1998). Agent architectures provide methodologies for realizing the features and properties of agents. Architectures for building a single agent can be classified into three categories: deliberative agent architecture, reactive agent architecture, and hybrid agent architecture (Müller, 1998; Wooldridge & Jennings, 1995).

Deliberative agents maintain an internal symbolic representation of their world. These agents have explicit mental states represented as beliefs, goals, or intentions, which can be modified through logical reasoning. Deliberative agents are capable of achieving complex goals, but may fail to respond timely to events in the environment because of the amount of resources and time required for reasoning.

Reactive agents have no internal representation or a very simple internal representation of the world. These agents make their decisions at run-time based on information perceived from the outside world and simple situation-action rules. Reactive agents can respond to environmental changes in a timely fashion, but are not able to achieve sophisticated goal-directed behaviors through complex reasoning.

Hybrid agent architectures are proposed to overcome the shortcomings of deliberative and reactive agent architectures. A hybrid agent includes reactive and deliberative components. The reactive component is able to react to events that occur in the environment without complex reasoning, and the deliberative component is capable of developing plans and making decisions based on its world model. The reactive component and deliberative component interact with each other to enable the agent to achieve both real-time reactive and goal-directed behaviors.

When a system involves multiple agents, the system architecture is named “interacting agent architecture” (Müller, 1998), or “distributed agent architecture” (Luck, Ashri, & d’Inverno, 2004). Interacting agents have the social ability to coordinate their activities with other interacting agents through communication, negotiation and cooperation.

Software Agents

The origination of software agents results from the research on computational models of distributed intelligence. Software agent research aims to simplify distributed computing by overcoming barriers to intelligent interoperability, incorporating agents as resource managers, and extending user interface abilities (Bradshaw, 1997).

A software agent is “a software entity which functions continuously and autonomously in a particular environment, often inhabited by other agents and processes” (Bradshaw, 1997, p.7). Nwana (1996) reviewed the research of software agents and suggested that the term “software agent” might best be viewed as an umbrella term that covers a range of specific agent types.

Researchers have proposed a variety of characteristics of software agents based on various applications. Foner (1993) conducted a case study to analyze the interaction between users and an entertaining software agent, Julia, in a multi-user domain (MUD) environment. Based on the interaction analysis, Foner described nine characteristics that a software agent might possess. These characteristics are autonomy, personalizability, discourse, risk and trust, domain (agents are used in appropriate domains), graceful degradation, cooperation, anthropomorphism, and expectations (agents satisfy users' expectations). Etzioni and Weld (1995) interpreted the term software agent as a computer program that behaves in a manner of a human agent and support a social dialogue between a person and the agent. They proposed six features of software agents. These features are autonomy, temporal continuity, personality, communication ability, adaptability, and mobility. Recent literature suggests that autonomy is the only generally agreed upon feature of software agents (Lockemann, 2006). As Nwana (1996) suggested, the term "software agent" is an umbrella term that covers a variety of specific agent types, such as interface agents (Laurel, 1997), information agents (Klusck, 2001), and pedagogical agents (Johnson, Rickel, & Lester, 2000). Depending on the purpose of application, a software agent may possess one or more of these features. The characteristics of software agent are summarized and presented in Table 11.

Table 11

Characteristics of Software Agent (adapted from Bradshaw, 1997)

Characteristics of software agent	Meaning
Reactivity	The ability to selectively sense and act in an environment
Autonomy	Goal-directedness, proactive and self-starting behavior
Collaboration	Work with other agents to achieve common goals
Communication	The ability to communicate with persons and other agents
Inferential capability	Can act on an abstract task specification using prior knowledge of general goals and preferred methods to achieve flexibility; goes beyond the information given, and may have explicit models of self, user, situation, and/or other agents.
Temporal continuity	Persistence of identity and state over long periods of time
Personality	The capability of manifesting the attributes of a “believable” character, such as emotion
Adaptivity	Being able to learn and improve with experience
Mobility	Being able to migrate in a self-directed way from one host platform to another

Multi-Agent Systems

In interconnection and networking environments, agents need to interact with other agents. A number of agents constitute a multi-agent system through interaction with each other in a distributed environment to pursue common goals or perform tasks (Weiss, 1999). d'Inverno & Luck (2004) presented a definition of multi-agent systems:

A multi-agent system is any system that contains

1. two or more agents;
2. at least one autonomous agents; and
3. at least one relationship between two agents where one satisfies the goal of the other. (d'Inverno & Luck, 2004, p. 42)

Each agent in a multi-agent system is an independent problem-solver and these agents come together to constitute a whole system. Multi-agent systems have features that single agent systems do not possess (d'Inverno & Luck, 2004). These features are presented in Table 12.

Table 12

Features of Multi-Agent Systems (adapted from d'Inverno & Luck, 2004)

Features	Meaning
Agent modeling	An agent needs a model of its environment that contains other agents.
Multi-agent planning	Multi-agent systems may need plans to coordinate agents' behaviors or to achieve a goal with others.
Social relationships	An agent provides services for another agent or cooperates with other agents.
Interaction	Interaction in multi-agent systems is not pre-defined. Agents need models of each other to decide how to interact with each other.
Communication	Multi-agent systems need communication mechanisms and protocol for agent interaction.

Agent-Oriented Software Design Methodology

Features of Agent-Oriented Software Design

The concept of agents not only provides a new way to develop computing technologies, but also offers a novel perspective on software design for managing the inherent complexity of software systems (Jennings, 2000, 2001; Luck, Ashri, & d'Inverno, 2004). In an agent-oriented system, agents autonomously interact with each other and the environment to pursue common goals or perform tasks. With agents as a means of abstraction, the agent-oriented paradigm provides a natural and intuitive perspective on modeling, designing, and building complex, distributed software systems.

The agent-oriented approach deals with complex software design in three ways: decomposition, abstraction, and organization (Jennings, 2000, 2001). First, a complex system is decomposed into a set of problem-solving components, which are in turn designed as various types of agents. Agents are autonomous and can decide their actions based on local situations. The agent-oriented decomposition reduces the complexity of control and increases the flexibility of interaction, thus supporting distribution and scalability of software systems. Second, a set of agent-oriented concepts and notions is used to abstract and model computing systems. Some basic elements in the set include agent, goal, role, environment, interaction, and the organization of agents (Sterling & Taveter, 2009). The agent-oriented abstraction provides an effective means for modeling complex systems. Third, agents are organized in a multi-agent structure, and an application is modeled as a society of interacting agents. The individual agent or a group of organizational agents can be designed and incrementally added into the system. The agent-oriented organization makes the design more flexible and increases software reusability.

Prometheus: An Agent-Oriented Software Design Method

A variety of agent-oriented analysis and design methodologies have been developed to support developing agent-based systems (Bergenti, Gleizes, & Zambonelli, 2004; Henderson-Sellers & Giorgini, 2005). The widely recognized methods include the Gaia methodology (Wooldridge, Jennings, & Kinny, 2000; Zambonelli, Jennings, & Wooldridge, 2003), Multi-agent Systems Engineering (MaSE) methodology (Deloach, Wood, & Sparkman, 2001), the Tropos methodology (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004), and the Prometheus methodology (Padgham &

Winikoff, 2004). Among these methods, Prometheus is a well-documented and practical software design method aimed at developing intelligent multi-agent systems.

The Prometheus methodology consists of three design phases: system specification, architectural design, and detailed design. The first design phase is system specification, which aims to specify system goals and functionalities, and identify percepts (inputs), actions (outputs), and the data that the system will use and produce. Percepts are incoming information from the environment. Actions are the means by which the system affects its environment.

The development of scenarios is used to identify and refine system interface, data, goals and functionalities. Scenarios describe use cases of the system in operation. A typical scenario includes a sequence of structured steps. A step in a scenario can be a percept, an action, a goal, or a sub-scenario. System specification is usually an iterative process.

The second phase of the Prometheus methodology is architectural design, which determines agents that a system will contain and how they will interact. This stage involves three types of activities: (a) specifying agent types, (b) defining the interactions between agents, and (c) designing the overall system structure. Agent types are determined by grouping functionalities through data coupling. The system's structure is illustrated in a system overview diagram, which displays a general picture of how the system as a whole will function. It encompasses the agents, the communications between agents, percepts, actions and data. It also presents the system's boundary and its environment, including percepts and actions.

The third phase, detailed design, describes the internals of each agent and the ways in which agents will achieve their goals within the overall system. The focus is on defining agent capabilities and designing events, plans, and detailed data structures for each agent capability.

The Use of Software Agents in Learning Assessment

Researchers have made efforts to facilitate learning assessment with software agents. One application of software agents in assessment is automatic generation of assessment questions and evaluation of responses. Tecuci and colleagues (Tecuci, 1998; Tecuci & Keeling, 1999) developed an intelligent learning assessment system based on a learning agent shell that consists of a learning engine and an inference engine. After adding a test generator and a domain specific user interface, the learning agent shell was extended to a learning agent of history education. A history education expert interacted with the learning agent to develop its knowledge base and trained its skills of generating test questions and assessing student responses. With the capability of generating personalized tests and assessing responses, the learning agent became a history learning assessment agent in the field of slavery in America. Since this kind of assessment agent can automatically produce multiple-choice questions, assess responses and provide feedback, it significantly reduces instructors' workload.

In another study, Anohina and Grundspenkis (2006, 2007) developed an intelligent knowledge assessment system to provide instructors with a tool to evaluate students' knowledge and use the evaluation results to adjust and improve instruction. The format of an assessment task is a fixed concept map structure with blank nodes. In the concept map structure, nodes are connected with linking lines without arrowheads. There

are also no linking phrases. Students are asked to select node terms from a list and fill in the blank nodes. The core of the intelligent knowledge assessment system is an assessment module that includes a communication agent, a knowledge evaluation agent, an interaction registering agent, and an agent-expert. These agents work collaboratively to deliver tasks, collect students' responses, evaluate students' responses, and provide feedback.

Software agents have been integrated in e-learning systems to support adaptive assessment that fits learners' characteristics and knowledge levels. With the Bayesian network technique, Giotopoulos et al. (2007) developed a questioner assessment agent to provide personalized assessment in e-learning environments. A set of related multiple-choice questions are organized as a Bayesian network, which is called a questioner (K. C. Giotopoulos, personal communication, September 28, 2008). The Bayesian network provides the questioner assessment agent with a means to determine the next question presented to the student based on his or her answers to previous questions. The questioner assessment agent handles questioners, judges students' responses, and makes decisions on the sequences of question presentation to fit students' knowledge levels.

In an agent-oriented collaborative learning system (Lafifi & Bensebaa, 2007), an assessment agent manages and delivers exercises for formative assessment. The types of exercise in the system are multiple-choice questions, true/false questions, and fill-in-the-blank questions. Eleven parameters, such as model of exercise presentation, number of exercises for a learning objective, and sequence of exercise presentation, are used to configure learning objectives, type and number of exercise, assessment time, as well as rules for judging learners' knowledge level. When a learner requests to take an

assessment, the assessment agent presents a set of exercises based on learners' characteristics and knowledge levels.

Another research effort is incorporating software agents to improve the quality of peer assessment. In the peer assessment process, students may use different criteria and produce different evaluating results on a piece of peer's work. In a computer-based peer assessment system, Lai and Lan (2006) used agent negotiation to reduce unfairness of assessment results. Each student possesses an agent when they conduct peer assessment with the system. After a group of students' agents receive evaluation results, these agents select a negotiation strategy and begin to negotiate with each other until reaching an agreement on an evaluation result. Lai and Lan examined the validity of the agent negotiation. The experiment result indicates that the agent negotiated peer assessment marks and the instructor's assessment marks are significantly and positively related.

Summary of Literature Review

Concept maps provide a means for revealing students' understanding of concepts as well as relationships between concepts, which are not easy to assess with multiple-choice questions or even essay writing. A concept map assessment is composed of a task, a response format, and a scoring system. One viable concept mapping task format for computer-based large-scale assessment is to provide students with node terms and linking phrases, which include some node terms and linking phrases that serve as distracters. Research indicates that comparing student maps with criteria maps is a reliable and valid scoring method.

The existing computer-based tools for assessing concept maps are primarily developed for formative assessment and used to promote learning and improve

instruction. No existing concept map assessment system fully addresses the needs of large-scale assessment based on concept maps. Therefore, it is worthwhile to conduct a study to develop a computer-based system for large-scale assessment based on concept maps.

Software agent technology enriches the means and increases the possibility of developing such a large-scale assessment tool. The concept of agents not only supports developing advanced computing technologies but also provides a novel perspective on software development. The agent-oriented approach offers a software engineering paradigm for modeling, designing, and building complex software systems. As an umbrella term, software agents include a variety of specific agent types that possess various features. Among these features, reactivity, autonomy, collaboration, communication, inferential capability, and temporal continuity should be addressed in the design and development of an agent-oriented system for large-scale assessment.

Software agents have been used to support the academic assessment process. The applications include generating multiple choice tests and judging answers, evaluating concept maps with fixed structure, providing adaptive assessment, and coordinating peer assessment results. The review determined there has been no research on systematically exploring the functionalities and system architecture for building software agent-oriented systems for large-scale assessment. The agent-oriented software design paradigm has a potential in developing a sophisticated system for large-scale assessment based on concept maps.

Chapter Three: System Design, Development, and Evaluation

The Design of the Assessment Agent System

An agent-oriented software design method, named Prometheus (Padgham & Winikoff, 2004), was adopted to guide the system design process. The design process included three phases: system specification, architecture design, and detailed design (see Table 13).

Table 13

The Assessment Agent System Design Process

Phase	Step
System specification	Analyzing system requirements
	Specifying system functionalities
	Developing scenarios
	Specifying interface and data
Architecture design	Specifying agent types
	Describing interaction between agents
	Designing overall system architecture
Detailed design	Designing agent capabilities
	Designing plans for actualizing agents' capabilities

Phase 1: System Specification

Step 1: Analyzing system requirements. An assessment agent is a type of software agent that may autonomously produce items for assessment or assess responses of items on behalf of others. The assessment agent in this study assesses concept maps created and submitted by students. An assessment agent system is composed of assessment agents and other types of software agents. The system is used to create, deliver, assess, and manage assignments and examinations.

There are three types of activities in the process of concept map assessment in this study. First, experts or instructors create assessment tasks and assessment criteria, which are stored in a database. Second, students create concept maps as responses to the assessment task and submit the concept maps for assessment. To strike a balance between effectively revealing students' knowledge structures and efficiently assessing student concept maps with computers, the assessment task format is that students construct concept maps with a given set of node terms and linking phrases. The node terms and linking phrases provided to students include both appropriate and inappropriate terms and phrases; however, inappropriate terms and phrases serve as distracters. Third, the computer assesses student-created concept maps by comparing them with the expert-created assessment criteria.

The first step in the system specification phase involved analyzing the system requirements. This process began with an analysis of the generic requirements of computer-based assessment systems. The generic analysis was followed by an analysis of specific assessment agent system requirements associated with this study.

To take advantage of computers for assessment, researchers have proposed expectations for computer-based assessment systems (Bull & McKenna, 2004; Burkhardt & Pead, 2003). These expectations cover four important aspects of assessment, including assessment delivery and task presentation, student working and response, response evaluating, as well as assessment management and administration. The details of these expectations are summarized and presented in Table 14.

Table 14

Expectations for Computer-based Assessment Systems

Aspect of assessment	Expectation for computer-based assessment systems
Assessment delivery and task presentation	Adaptivity to individual student's ability Communication via networks Performance support Potential for large-scale assessment Potential for diagnostic assessment, self-assessment, formative assessment, or summative assessment Use of multimedia
Student working and response	Intuitive user interface User interactivity Natural and rich working environments
Response evaluating	Automatic assessing Real time feedback
Management and administration	User administration Task authoring and management Collecting and analyzing responses Task quality analysis Reporting Security Large item banks for reuse

The assessment agent system in this study is a special type of computer-based assessment system. Its requirements were specified based on the expectations for computer-based assessment systems and the findings of reviewing concept maps as an assessment tool. The assessment agent system requirements for this study (see Table 15) reflect the most important expectations for computer-based assessment, including task creation and management, task delivery, student working and responding to assessment tasks, automatic assessing, assessment reporting, and assessment security.

Table 15

Assessment Agent System Requirements

Category of system requirements	System requirements
User administration and security	Create, maintain and update user profile User authentication
Task authoring and management	Create, edit, and manage assessment tasks Create, edit, and manage assessment criteria
Assessment delivery	Deliver assessment via the Internet
Task presentation	Use text to describe the task Provide nodes and linking phrases
Working environment	Intuitive user interface Smooth user-system interactivity
Response collection	Recognize concept maps Accept and store responses
Assessing and feedback	Automatically assess concept maps Provide feedback on student maps
Reporting	Report individual student performance Report a group of student's performance

Step 2: Specifying system functionalities. To meet the specified system requirements identified in Step 1, the assessment agent system should possess corresponding functionalities and goals. The functionalities of the assessment agent system are behaviors of the system that allow it to achieve its goals and meet users' requirements. The functionalities that must be addressed by the system in this study include user-system interaction, user management, task authoring and management, assessment delivery, task presentation, response collection, automatic assessing with

feedback, and reporting. The corresponding goals of the functionalities are presented in Table 16.

Table 16

Functionalities and Goals of the Assessment Agent System

System functionality	System goals
User-system interaction	Provide text areas for receiving inputs from users Provide menus and/or buttons for receiving commands Support operations of clicking, dragging, and dropping to build concept maps Display information
User management	Accept, maintain, and update user profiles Identify users
Task authoring and management	Accept and store data in a database Provide course and task lists for selection Provide student name lists for selection
Assessment delivery	Deliver assessment tasks to users via the Internet
Task presentation	Provide course and task lists for selection Present task description in text Provide nodes and linking phrases
Response collection	Recognize concept maps Accept and store responses in a database Notify student map reception
Automatic assessing with feedback	Compare student maps with assessment criteria Send results and feedback to a database
Reporting	Query assessment results in a database Generate report on individual student performance Generate report on a group of student's performance Deliver report on individual student performance Deliver report on a group of student's performance Present report on individual student performance Present report on a group of student's performance

Step 3: Developing scenarios. A scenario illustrates the sequence of steps to achieve goals in a use case of the system in operation (Padgham & Winikoff, 2004). The scenario development technique is used to identify and refine functionalities, goals, percepts, actions and data. A goal refers to the desired state to be achieved through a system operating process.

In this study, the main scenarios in the assessment agent system include: (a) assessment task creation, (b) task assignment, (c) task presentation, (d) student map creation, (e) student map submission, (f) concept map assessment, (g) assessment report request, and (h) student performance report request. These scenarios are described with a format suggested by Padgham and Winikoff (2004). Table 17 presents the student map submission scenario, which illustrates steps responding to a student's request of submitting a student map. The descriptions of other scenarios can be found in Appendix A.

Table 17

Steps in the Student Map Submission Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
2	Percept	Student map submission request	User-system interaction	
3	Goal	Identify users	User management	Uses: Records in user database tables
4	Goal	Recognize concept maps	Response collection	
5	Goal	Accept and store responses in a database	Response collection	Produces: Records in student map database tables
6	Action	Student map submission confirmation	User-system interaction	

Steps 4: Specifying interface and data. The interface between the assessment agent system and its users includes percepts and actions, which are inputs and outputs of the system. Percepts are information received from users by the system. Actions are the system's behaviors that affect its users.

Percepts of the system include: (a) entering text, (b) task submission request, (c) existing task editing request, (d) task assigning request, (e) report request, (f) task presentation request, (g) item selection, (h) selecting and moving a node, (i) drawing a linking line, (j) dragging and dropping a linking phrase, (k) student map submission request, (l) assessment report request, and (m) exit.

Actions of the system include: (a) text display, (b) task submission confirmation, (c) task assigning confirmation, (d) report display, (e) task display, (f) node display, (g) linking line display, (h) linking phrase display, (i) student map submission confirmation, and (j) assessment report display.

Three types of data are involved in the system and stored in a database, including user data, task data, and student map data. User data contains information of user profile, task created or taken, assessing results and feedback. Task data contains information about course title, task title, task description, node terms and linking phrases, assessment criteria, and task creator name. Student map data contains information about submitted concept maps, student names and identification number.

Phase 2: System Architecture Design

The assessment agent system architecture design defined agents as well as their interaction with each other and with the environment to provide the specified system functionalities. The architecture design included three steps: (a) specifying agent types, (b) describing interaction between agents, and (c) designing the overall system structure.

Step 1: Specifying agent types. Through analyzing data coupling and grouping functionalities, five types of agents were designed for the system, including instructor agent, student agent, management agent, assessment agent, and reporting agent (see Figure 10). Each type of agent is described in the following section.

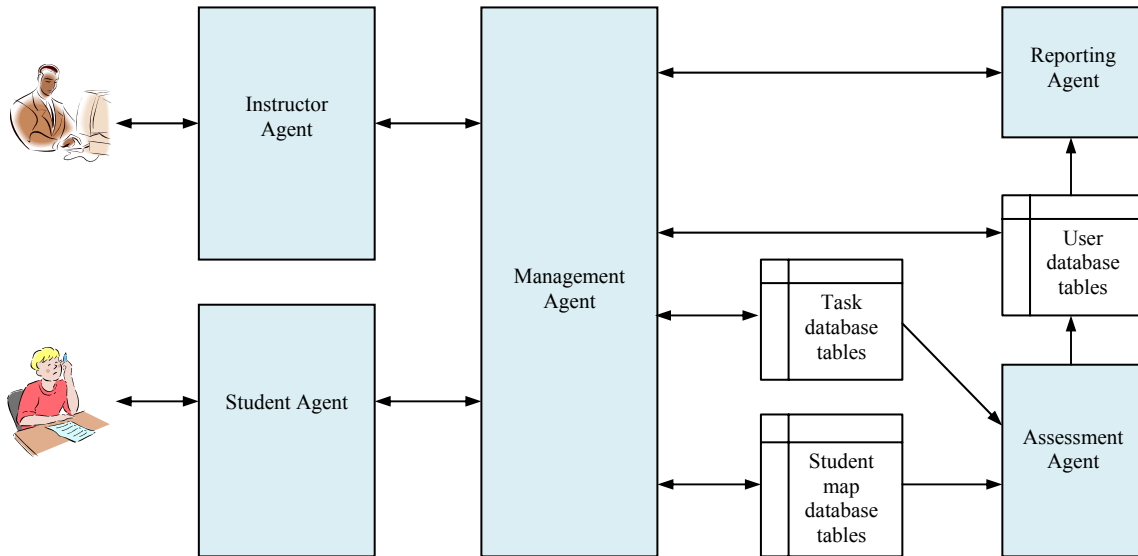


Figure 10. Five types of agents.

Instructor agents interact with instructors who initiate the interaction. The functionalities included in an instructor agent are user-system interaction, task authoring and management, and reporting. With an instructor agent, an instructor can create and edit assessment tasks, assign tasks to students, and obtain reports on student performance.

Student agents provide functionalities of user-system interaction, task presentation, response collection, and reporting. A student can request presentation of assessment tasks, create and submit concept maps, and request assessment reports through interacting with a student agent.

The management agent deals with all requests from instructor agents and student agents, communicates with the assessment agent and reporting agent, and manages data in the database. The functionalities included in the management agent are user management, task authoring and management, assessment delivery, and response collection.

The assessment agent assesses submitted student maps and generates feedback on student maps. The functionality included in the assessment agent is automatic assessing with feedback. The reporting agent queries the database, generates reports on student performance, and delivers reports to the management agent.

Step 2: Describing interaction between agents. The interactions between agents are illustrated and defined with interaction diagrams, messages, and message groups. An interaction diagram illustrates interactions between agents (Padgham & Winikoff, 2004). A message is information sent from an agent to another agent. A message group includes a series of messages exchanged between two agents for responding to a request. The following sections describe the interaction between agents for responding to user requests.

Agent interaction for responding to a task submission request. An instructor may interact with an instructor agent to create assessment tasks. After finishing creation of a task, the instructor makes a task submission request to the instructor agent. The instructor agent communicates with the management agent to store the task in the task database tables. Figure 11 illustrates the interactions and communications among the instructor, the instructor agent, and the management agent during the task submission process.

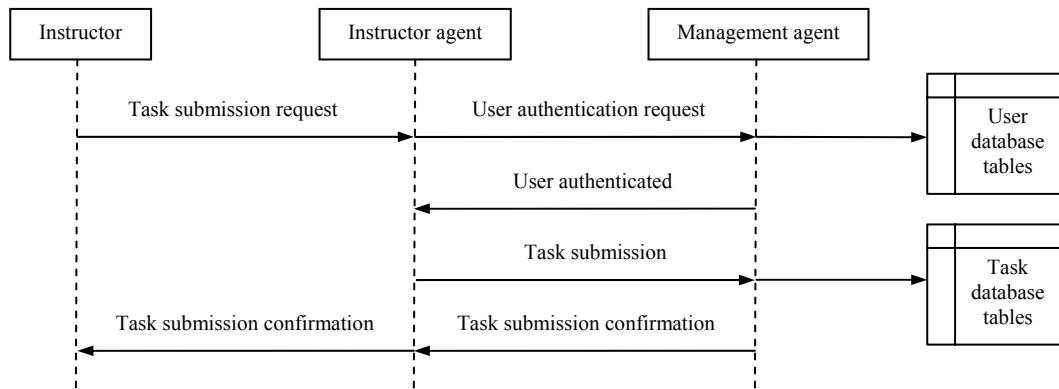


Figure 11. Interaction diagram of assessment task submission.

After receiving the task submission request, the instructor agent and the management agent exchange messages to deal with the task submission request from the instructor. These messages compose the task submission message group, which includes user authentication request message, user authenticated message, task submission message, and task submission confirmation message. The task submission message group defines the communications between an instructor agent and the management agent when an instructor makes a request to submit an assessment task.

Agent interaction for responding to a task assigning request. An instructor may assign an assessment task to students by making a task assigning request to the instructor agent. The instructor agent communicates with the management agent to respond to the instructor's request. Figure 12 illustrates the interactions and communications among the instructor, the instructor agent, and the management agent during the task assigning process.

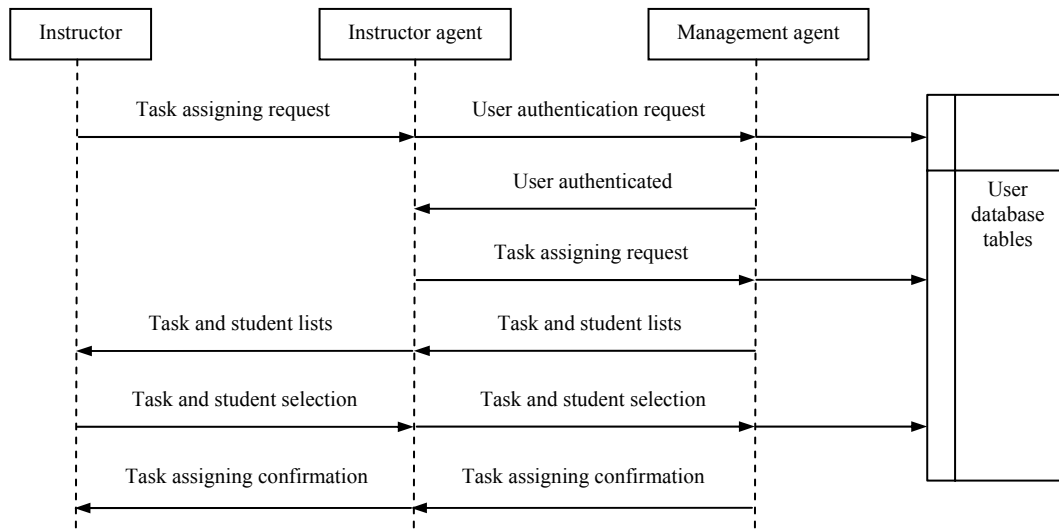


Figure 12. Interaction diagram of assessment task assignment.

After receiving a task assigning request, the instructor agent and the management agent exchange messages to deal with the task assigning request from the instructor. These messages compose the task assigning message group, which includes user authentication request message, user authenticated message, task assigning message, task and student lists message, task and student selection message, and task assigning confirmation message. The task assigning message group defines the communications between an instructor agent and the management agent during assigning an assessment task to students.

Agent interaction for responding to a task presentation request. When taking an assessment, a student makes a task presentation request to his or her student agent. The student agent communicates with the management agent for delivering a task. Figure 13 illustrates the interactions and communications among the student, the student agent, and the management agent during the task presentation process.

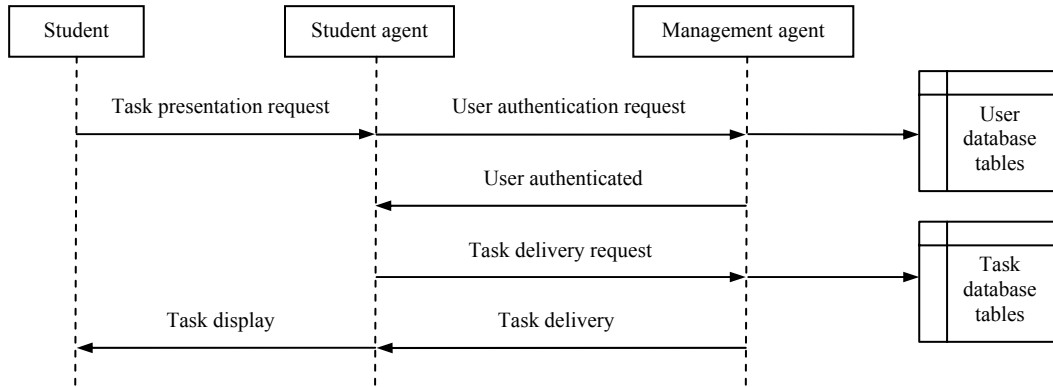


Figure 13. Interaction diagram of task presentation.

After receiving a task presentation request, the student agent and the management agent exchange messages to deal with the task presentation request from the student. These messages compose the task presentation message group, which includes user authentication request message, user authenticated message, task delivery request message, and task delivery message. The task presentation message group defines the communications between a student agent and the management agent when a student makes a task presentation request.

Agent interaction for responding to a student map submission request. After finishing creation of a student map, a student makes a student map submission request to the student agent. The student agent communicates with the management agent to store the student map in the student map database tables. Figure 14 illustrates the interactions and communications among the student, the student agent, and the management agent during the student map submission process.

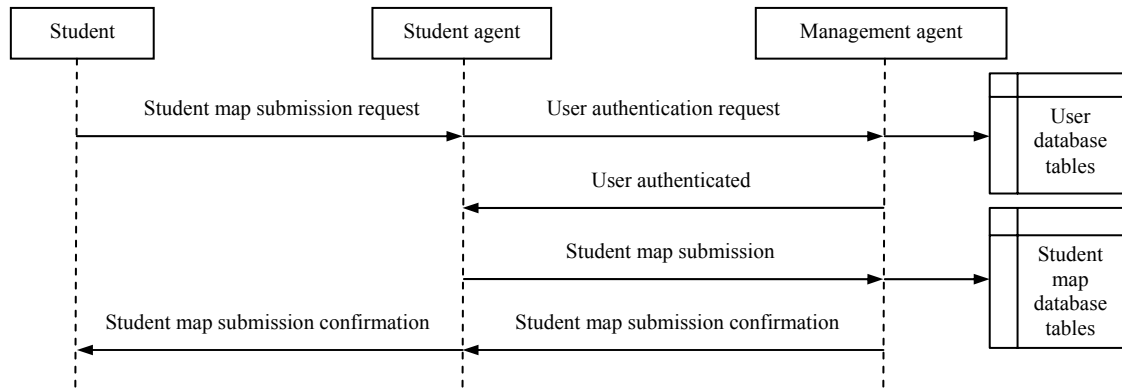


Figure 14. Interaction diagram of student map submission.

After receiving a student map submission request, the student agent and the management agent exchange messages to deal with the student map submission request from the student. These messages compose the student map submission message group, which includes user authentication request message, user authenticated message, student map submission message, and student map submission confirmation message. The student map submission message group defines the communications between a student agent and the management agent when a student makes a request to submit a student map.

Agent interaction for responding to an assessment report request. After receiving an assessment report request, the student agent communicates with the management agent, and the management agent communicates with the reporting agent for generating and delivering the requested assessment reports. Figure 15 illustrates the interactions and communications among the student, the student agent, the management agent, and the reporting agent during the assessment report generating and delivering process.

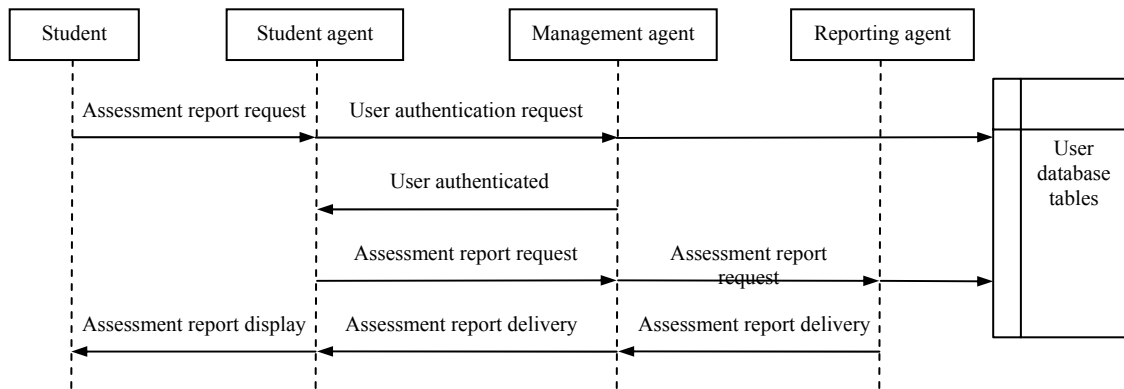


Figure 15. Interaction diagram of assessment report request.

The student agent and the management agent exchange messages to deal with the assessment report request from the student. These messages compose the assessment report request message group, which includes user authentication request message, user authenticated message, assessment report request message, and assessment report delivery message. The assessment report request message group defines the communications between a student agent and the management agent when a student makes an assessment report request.

Agent interaction for responding to a report request. After receiving a report request, the instructor agent communicates with the management agent, and the management agent communicates with the reporting agent for generating and delivering the requested report. Figure 16 illustrates the interactions and communications among the instructor, the instructor agent, the management agent, and the reporting agent during the report generating and delivering process.

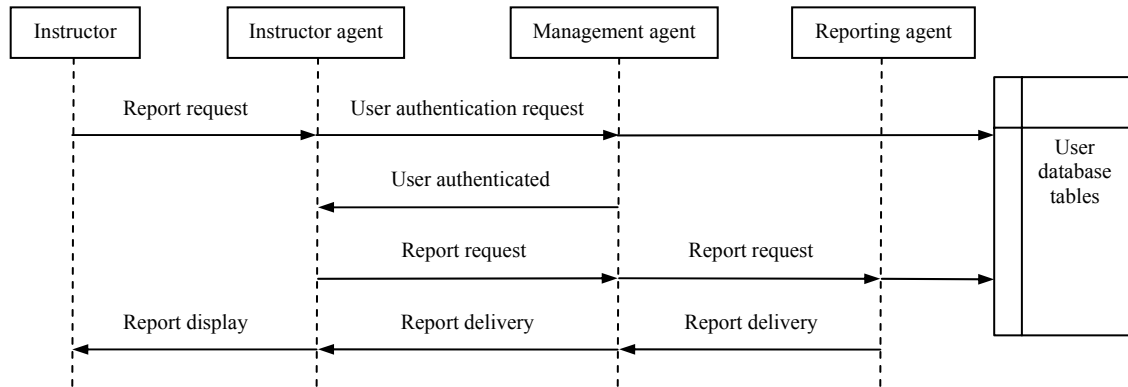


Figure 16. Interaction diagram of report request.

The instructor agent and the management agent exchange messages to deal with the report request from the instructor. These messages compose the report request message group, which includes user authentication request message, user authenticated message, report request message, and report delivery message. The report request message group defines the communications between an instructor agent and the management agent when an instructor makes a report request.

Step 3: Designing the overall system structure. The overall system structure was designed based on the defined agents, messages, message groups, percepts, actions, and data (see Figure 17). The assessment agent system includes one management agent, one assessment agent, one reporting agent, instructor agents and student agents. The management agent, assessment agent, and reporting agent are situated in a server computer. Instructor agents and student agents are situated in instructors' or students' computers. The distribution of instructor agents and student agents in local computers can effectively reduce the communication workload between the users and the server in large-scale assessment.

Users interact with their agents to perform tasks. For instance, when beginning to take an assessment, a student makes a task presentation request to his or her student agent. The student agent communicates with the management agent through the task presentation message group. The management agent queries the task database tables and delivers the task to the student agent. The student agent displays the assessment task to the student.

As for the instructor, when he or she makes a report request to an instructor agent, the instructor agent communicates with the management agent through the report request message group. The management agent sends a report request message to the reporting agent. The reporting agent queries the user database tables, generates a report on student performance and delivers it to the management agent. The management agent sends the report to the instructor agent and the instructor agent displays the report to the instructor.

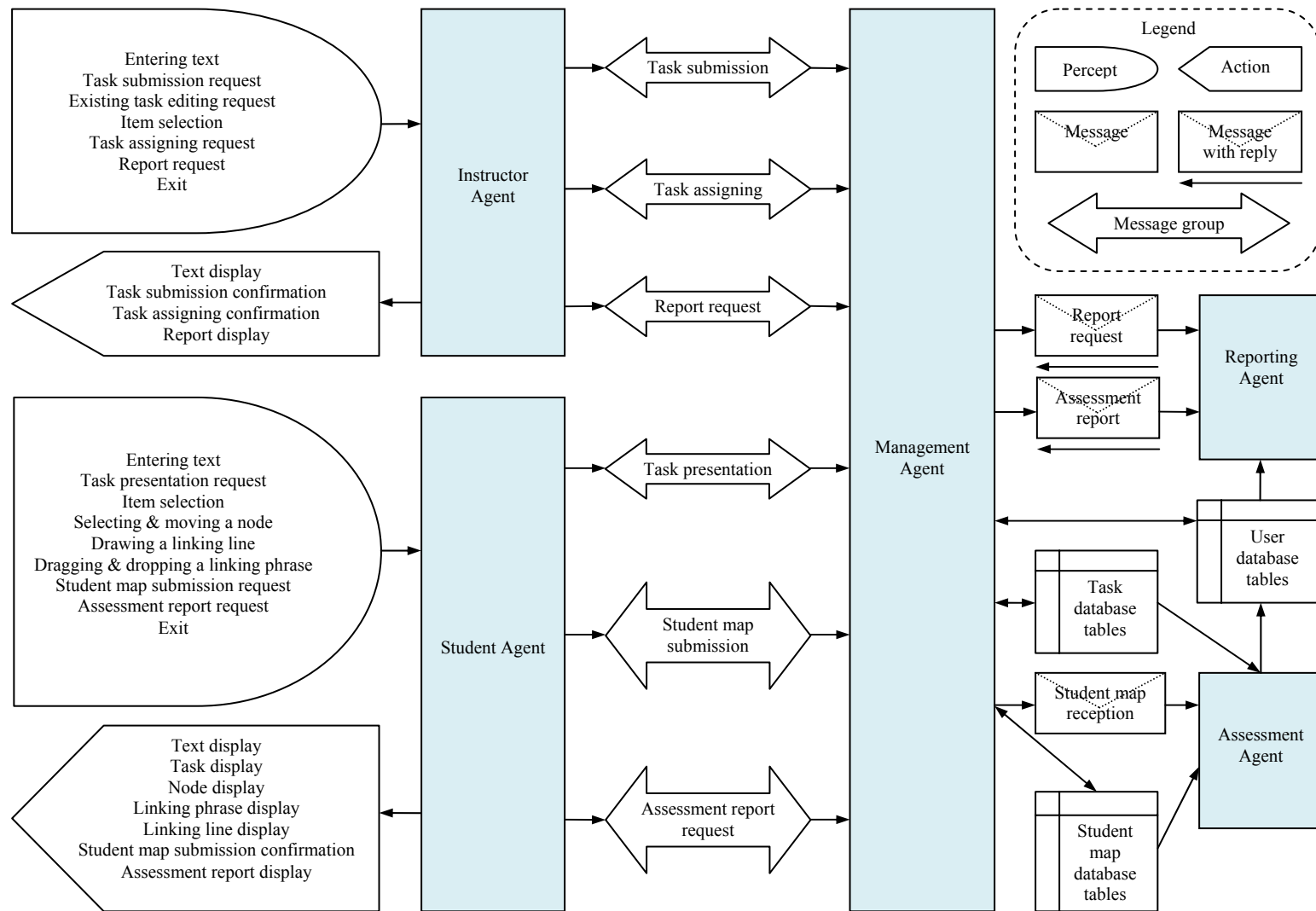


Figure 17. Overview of the assessment agent system structure.

Phase 3: Detailed Design of Agents

The detailed design of agents includes designing agents' capabilities to provide the assigned functionalities, and designing plans for actualizing agents' capabilities.

Detailed design of instructor agents. An instructor agent possesses communication capability and data handling capability. With the communication capability, an instructor agent receives entered text as well as commands, and displays information to interact with an instructor. The data handling capability allows an instructor agent to receive and send data to the management agent. Figure 18 presents the instructor agent overview diagram.

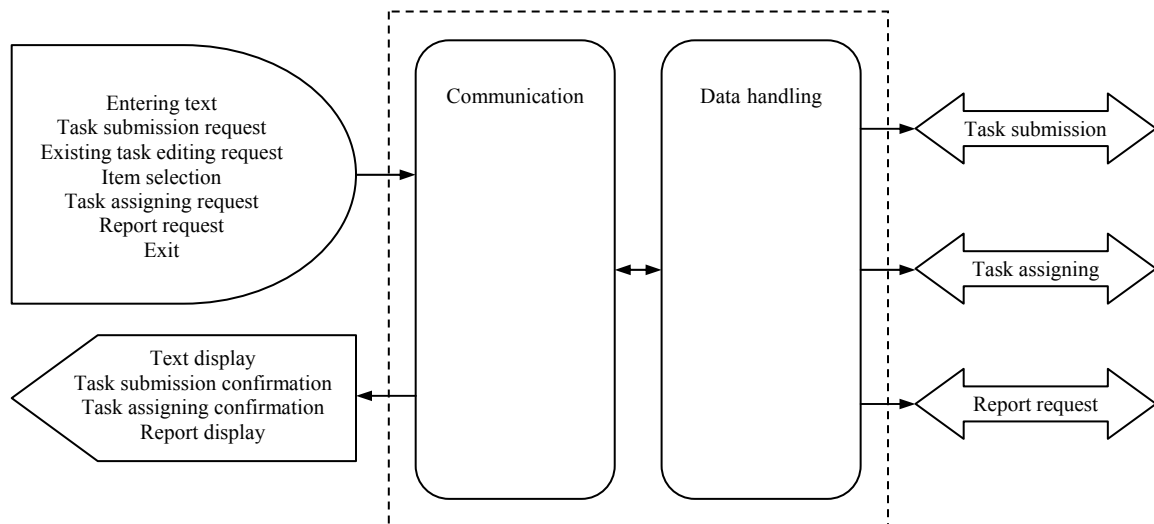


Figure 18. The instructor agent overview diagram.

There are seven types of percept that an instructor agent may receive, including instructor entering text, submitting tasks, editing existing tasks, selecting items, assigning tasks to students, requesting reports of student performance, and exiting an instructor agent. When an instructor agent receives a percept, it will execute a plan to respond to the percept. Table 18 presents the plans of instructor agents.

Table 18

Plans of Instructor Agents

Percept	Plan of instructor agent
Entering text	When an instructor enters text, such as task description, node terms, linking phrases, and assessment criteria, the instructor agent displays the entered texts on the corresponding area of the user interface.
Task submission	After receiving a task submission request, the instructor agent collects the task description, node terms, linking phrases, and assessment criteria, and sends to the management agent for storage. After the task is successfully stored in the database, the screen displays the confirmation information: “Your assessment task has been received and saved.”
Editing an existing task	After receiving a request to edit an existing task, the screen displays course titles and task titles for selection. The selected task contents are displayed for editing.
Task assigning and item selection	After receiving a task assigning request, the screen displays course titles, task titles, and student names for selection. The selected student names are associated with the selected task and sent to the management agent for storing in the database.
Report request	After receiving a report request, the screen displays the student names, course titles, and task titles for selection. The selected student names, course titles, task titles, and assessment reports are displayed on the screen.
Exit	After receiving an exit request, the instructor agent saves all data and closes itself.

Detailed design of student agents. A student agent possesses communication capability, concept map recognizing capability and data handling capability. With the communication capability, a student agent receives commands as well as entered text, and displays information to interact with a student. The concept map recognizing capability is used to extract student-created maps and transform them into a data format that can be stored in the database. The data handling capability allows a student agent to receive and send data to the management agent. Figure 19 presents the student agent overview diagram.

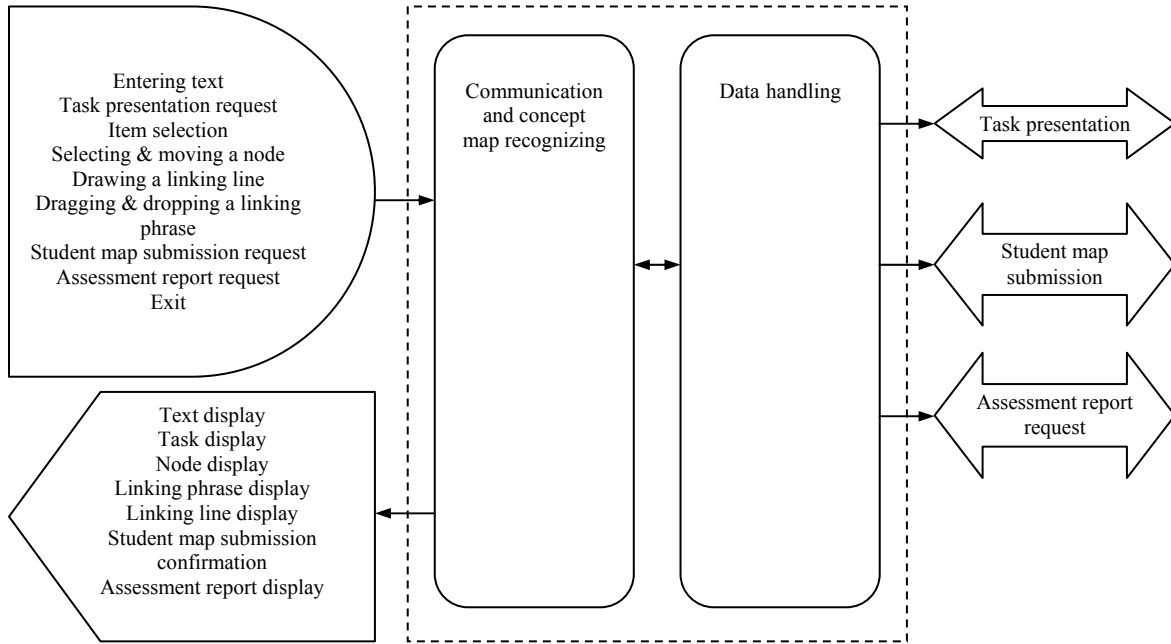


Figure 19. The student agent overview diagram.

There are nine types of percept that a student agent may receive, including student entering text, requesting task presentation, selecting an item, selecting and moving a node, drawing a linking line, dragging and dropping a linking phrase, submitting a concept map, requesting assessment reports, and exiting a student agent. When a student agent receives a percept, it will execute a plan to respond to the percept. Table 19 presents the plans of student agents.

Table 19

Plans of Student Agents

Percept	Plan of student agent
Task presentation request and item selection	After receiving a task presentation request, course titles and task titles are displayed on the screen for selection. After selecting the course title and task title, the nodes and linking phrases appear on the screen.
Selecting and moving a node	When a node is selected, dragged, and dropped on the mapping area, it is displayed on the screen. When a node is moved on the screen, it is displayed in the new place.
Drawing a linking line	When a linking line is drawn between two nodes, it is displayed on the screen.
Dragging and dropping a linking phrase	When a linking phrase is selected, dragged, and dropped into the box of a linking line, it is associated with the linking line and displayed on the screen.
Student map submission	After receiving student map submission request, the student agent extracts node terms and linking phrases for storage. After the student map is successfully stored in the database, the screen displays the confirmation information: "Your concept map has been received and saved."
Assessment report request	After receiving an assessment report request, the screen displays course titles and task titles for selection. After selection, the assessment report is displayed on the screen.
Exit	After receiving an exit request, the student agent saves all data and closes itself.

Detailed design of the management agent. The management agent possesses data handling capability and database managing capability. The data handling capability allows the management agent to receive and send data to other agents. With the database managing capability, the management agent can store, maintain, and query data in the database. Figure 20 presents the management agent overview diagram.

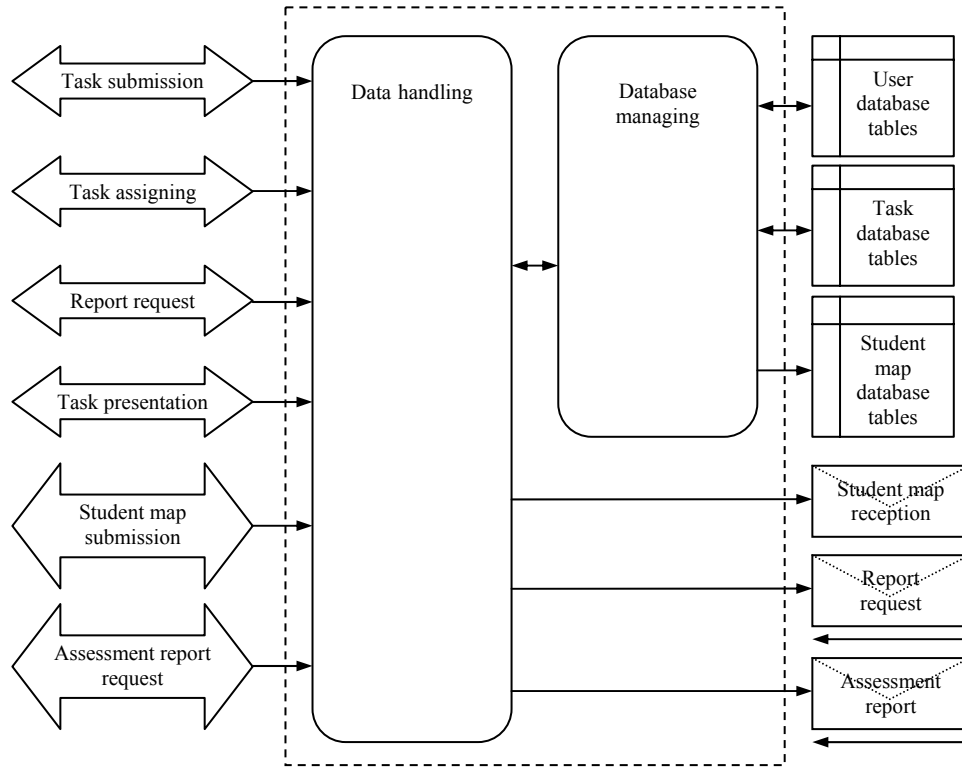


Figure 20. The management agent overview diagram.

The management agent communicates with instructor agents, student agents, the assessment agent, and the reporting agent. There are messages exchanged between the management agent and other agents. When the management agent receives a message from another agent, it executes a plan to respond to the message. These plans are listed in Table 20.

Table 20

Plans of the Management Agent

Message	Plan of the management agent
User authentication request	First, query user tables in the database. Second, reply to the message with verification information or user does not exist.
Task submission	First, insert the task content in the task tables in the database. Second, reply to the instructor agent with confirmation information: "Your assessment task has been received and saved."
Editing an existing task	First, query task tables in the database. Second, deliver course titles and task titles to the instructor agent. Third, after receiving the selected course title and task title, query task tables in the database and deliver selected task content to the instructor agent.
Assign a task	First, query task tables in the database. Second, deliver course titles and task titles to the instructor agent. Third, deliver student names to the instructor agent. Fourth, after receiving selected course title, task title, and student names, add the course title and task title to each of the selected student profiles in the user tables of the database. Fifth, reply to the message with confirmation information: "The task has been assigned to selected students."
Request report	First, send the message to the reporting agent. Second, after receiving a reply from the reporting agent, deliver the report to the instructor agent.
Task delivery request	First, query task tables in the database. Second, deliver the requested task to the student agent.
Student map submission	First, insert the node terms and linking phrases in the student map tables of the database. Second, reply to the student agent with confirmation information: "Your concept map has been received and saved."
Assessment report request	First, send the message to the reporting agent. Second, after receiving a reply from the reporting agent, deliver the report to the student agent.

Detailed design of the assessment agent. The assessment agent possesses capabilities of data handling, database managing, and comparing student maps with

assessment criteria. The data handling capability allows the assessment agent to receive messages from the management agent. With the database managing capability, the assessment agent can store, maintain, and query data in the database. The capability of comparing student maps with assessment criteria is used to assess student maps. When the assessment agent receives a student map reception message, it retrieves the student map in the student map database tables, and compares it with assessment criteria that are located in the task database tables. The assessment report, including the scores and feedback, are stored in the user database tables. Figure 21 presents the assessment agent overview diagram.

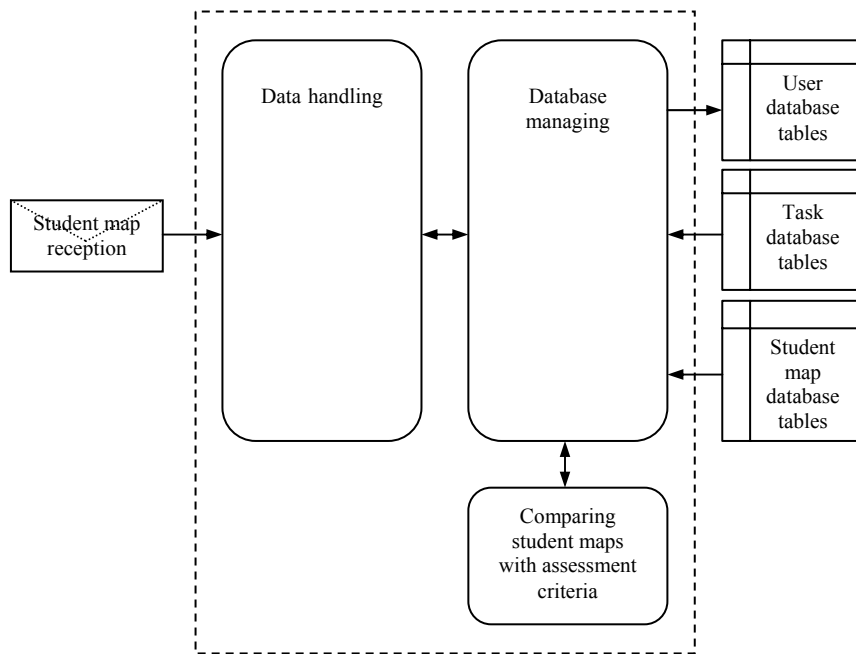


Figure 21. The assessment agent overview diagram.

To balance assessing the understanding of the main ideas and the details, the assessment focuses on measuring accuracy and completeness of student concept maps. In this dissertation, accuracy refers to the correctness of node terms, linking phrases, and

linking line directions in propositions of student concept maps. It is used to measure students' understanding of concepts and relationships between concepts. Completeness refers to the proportion of correct propositions in a student concept map out of all correct propositions in the assessment criteria. It is used to measure students' comprehensive conceptual understanding of an issue, a topic, a problem, or a phenomenon.

The method used to assess accuracy and completeness of student concept maps involves comparison of the student concept maps with node terms, linking phrases, and linking line directions contained in the database provided by experts. The following section provides an example to demonstrate the assessing method. Table 21 and Table 22 are respectively node terms and linking phrases provided to students for constructing their concept maps. These node terms and linking phrases include distracter terms and phrases. Figure 22 is an expert concept map. Table 23 includes propositions in the expert concept map, which serve as the initial criteria for assessing student concept maps.

Table 21

Node Terms (Created by an Expert)

Node term ID #	Node term
1	Problems
2	Research
3	Solutions
4	Theories
5	Real World
6	Phenomena
7	Ignorance
8	Truth

Table 22

Linking Phrases (Created by an Expert)

Linking phrase ID #	Linking phrase
1	Derive from
2	Drive
3	Develops
4	Generates
5	Reflect
6	Influence
7	Explains
8	Reveal

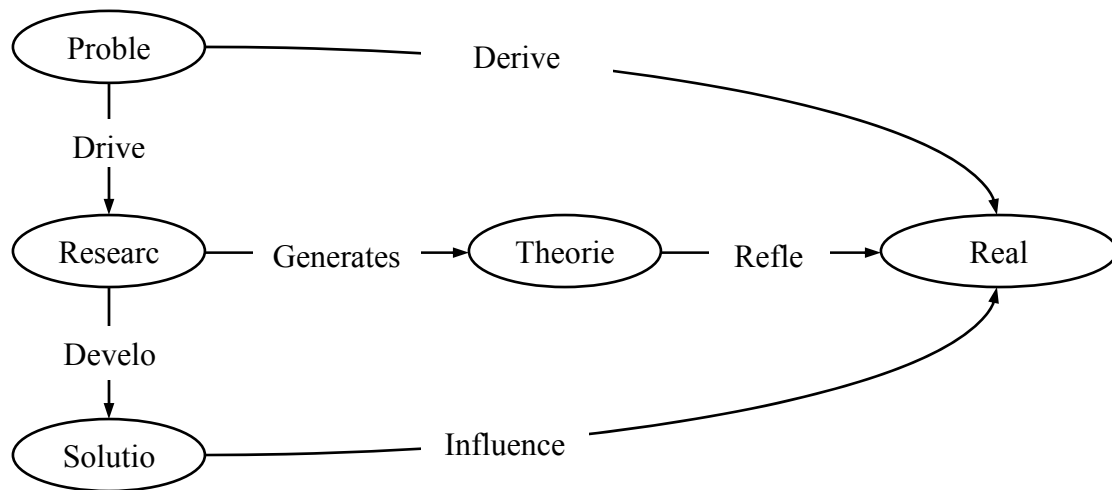


Figure 22. An expert concept map.

Table 23

Propositions in the Expert Concept Map (Created by an Expert)

Proposition ID #	Source node term	Linking phrase	Target node term	Category
1	Problems	Derive from	Real World	Correct
2	Problems	Drive	Research	Correct
3	Research	Develops	Solutions	Correct
4	Solutions	Influence	Real World	Correct
5	Research	Generates	Theories	Correct
6	Theories	Reflect	Real World	Correct

After a student constructs a concept map with a student agent and submits it (see Figure 23), the student concept map is transformed into a proposition table (see Table 24), and stored in the student map database tables. Next, the assessment agent assesses the student concept map by comparing the propositions in the student concept map with expert propositions. The initial assessment results of propositions in the student concept map fall into three categories: correct proposition, different proposition, and missing proposition (see Table 25). Correct propositions are student-created propositions that match correct propositions in the assessment criteria. Different propositions are student-created propositions that do not match any propositions in the assessment criteria. Different propositions may be either incorrect propositions or expert-ignored correct propositions. Missing propositions are correct propositions in the assessment criteria but not found in the student map.

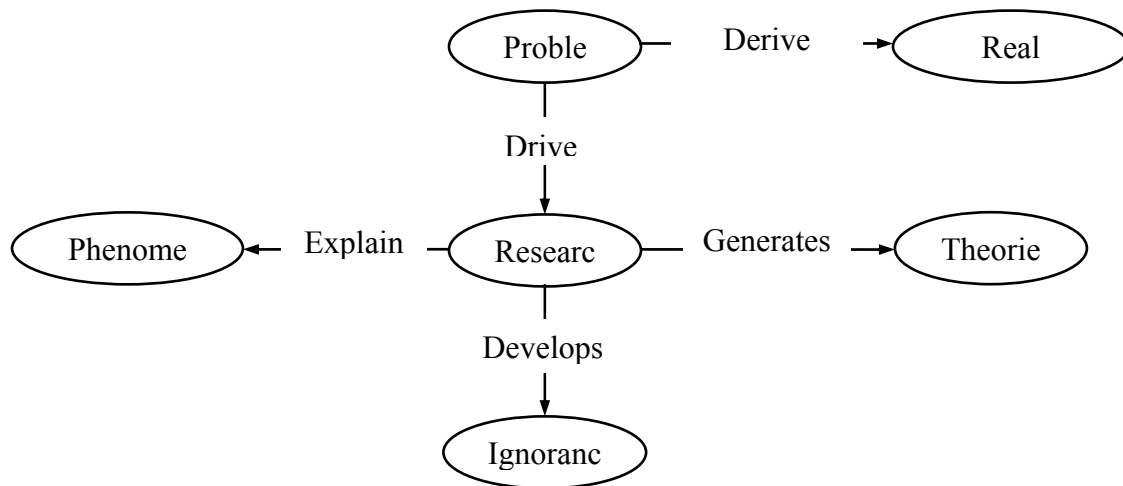


Figure 23. A student concept map.

Table 24

Propositions in the Student Concept Map (Created by Computer Based on the Student Concept Map)

Proposition ID #	Source node term	Linking phrase	Target node term
1	Problems	Derive from	Real World
2	Problems	Drive	Research
3	Research	Generates	Theories
4	Research	Explains	Phenomena
5	Research	Develops	Ignorance

Table 25

Assessment Results of Propositions in the Student Concept Map (Created by Computer Based on the Student Concept Map)

Source node term	Linking phrase	Target node term	Assessment result
Problems	Derive from	Real World	Correct
Problems	Drive	Research	Correct
Research	Generates	Theories	Correct
Research	Explains	Phenomena	Different
Research	Develops	Ignorance	Different
Research	Develops	Solutions	Missing
Solutions	Influence	Real World	Missing
Theories	Reflect	Real World	Missing

After the assessment agent produces the initial assessment result, an expert reviews propositions in the different-proposition category. The proposition Research→Explains→Phenomena is identified as correct, and the proposition Research→Develops→Ignorance is identified as incorrect. Both propositions are added to the assessment criterion propositions pool in the database. Table 26 presents the improved assessment criteria. The final assessment results of the student map are presented in Table 27.

Other student concept maps will be assessed against the improved assessment criteria. Since the assessment criterion propositions pool includes expert-identified incorrect propositions, assessment results generated by the assessment agent will be extended to four categories: correct proposition, incorrect proposition, different proposition, and missing proposition.

Table 26

Improved Assessment Criteria

Proposition ID #	Source node term	Linking phrase	Target node term	Category
1	Problems	Derive from	Real World	Correct
2	Problems	Drive	Research	Correct
3	Research	Develops	Solutions	Correct
4	Solutions	Influence	Real World	Correct
5	Research	Generates	Theories	Correct
6	Theories	Reflect	Real World	Correct
7	Research	Explains	Phenomena	Correct
8	Research	Develops	Ignorance	Incorrect

Table 27

Final Assessment Results of the Student Map

Source node term	Linking phrase	Target node term	Assessment result
Problems	Derive from	Real World	Correct
Problems	Drive	Research	Correct
Research	Generates	Theories	Correct
Research	Explains	Phenomena	Correct
Research	Develops	Ignorance	Incorrect
Research	Develops	Solutions	Missing
Solutions	Influence	Real World	Missing
Theories	Reflect	Real World	Missing

Detailed Design of the Reporting agent. The reporting agent possesses data handling capability and database querying capability. The data handling capability allows the reporting agent to receive messages, generate reports, and send data to the

management agent. With the database querying capability, the reporting agent can query data in the database. Figure 24 presents the reporting agent overview diagram.

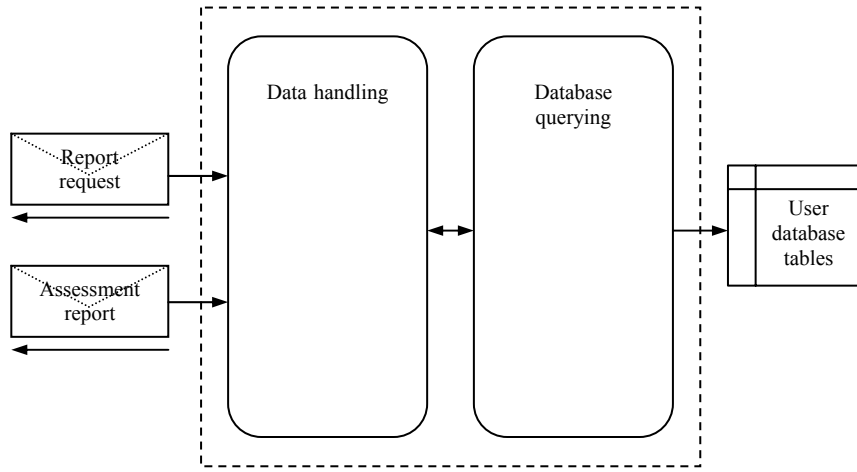


Figure 24. The reporting agent overview diagram.

When the reporting agent receives an assessment report request from the management agent, it queries the user tables in the database, generates reports, and delivers the reports to the management agent. Items in a report include student name, identification number, course title, task title, score of accuracy, score of completeness, and details of proposition assessment. Table 28 presents an example of an assessment report. Details of proposition assessment results are displayed in a format of Table 29.

Table 28

An Example of Assessment Report

Student Name	Smith, Joshua
Student ID	smithj
Course Title	Educational research
Task Title	Understanding research
Number of Correct Proposition	3
Number of Different Proposition	2
Number of Missing Proposition	3
Score of Accuracy	$3 \times 3 = 9$ points
Percent of Completeness	$3/6 = 50\%$

Table 29

Assessment results of the Student Concept Map (Created by Computer)

Assessment of propositions			
Source node term	Linking phrase	Target node term	Assessment of proposition
Problems	Derive from	Real World	Correct
Problems	Drive	Research	Correct
Research	Generates	Theories	Correct
Research	Explains	Phenomena	Different
Research	Develops	Ignorance	Different
Research	Develops	Solutions	Missing
Solutions	Influence	Real World	Missing
Theories	Reflect	Real World	Missing

The Development of the Assessment Agent System

The system was developed with several commercial and open-source tools, including Adobe Dreamweaver, Flash, ColdFusion, and MySQL. Table 30 outlines the system development process.

Table 30

The System Development Process

Develop database
Develop user interface
Develop system functionalities
Test, improve, and finalize system

Database Development

One important part of the system is a database for data storage and management. The database development includes designing a data model by identifying entities and attributes, normalizing tables and fields, as well as defining relationships between tables, and implementing the database with MySQL. The data involved in the system are classified into three types: user data, task data, and student map data. User data are stored in student tables and instructor tables. Task data are stored in course tables, task tables, linking phrase tables, node term tables, and expert proposition tables. Student map data are stored in student proposition tables.

User Interface Development

The student interface. The user interface was designed to be intuitive and easy to understand and operate. The student interface is composed of a series of web pages developed with Adobe Dreamweaver and Flash. The major student interface web page is a concept map creation web page displaying node as well as linking phrases. The bottom area of the page houses buttons that provide functionalities for returning the student menu, viewing task description, submitting concept map, and exiting the system (see Figure 25).

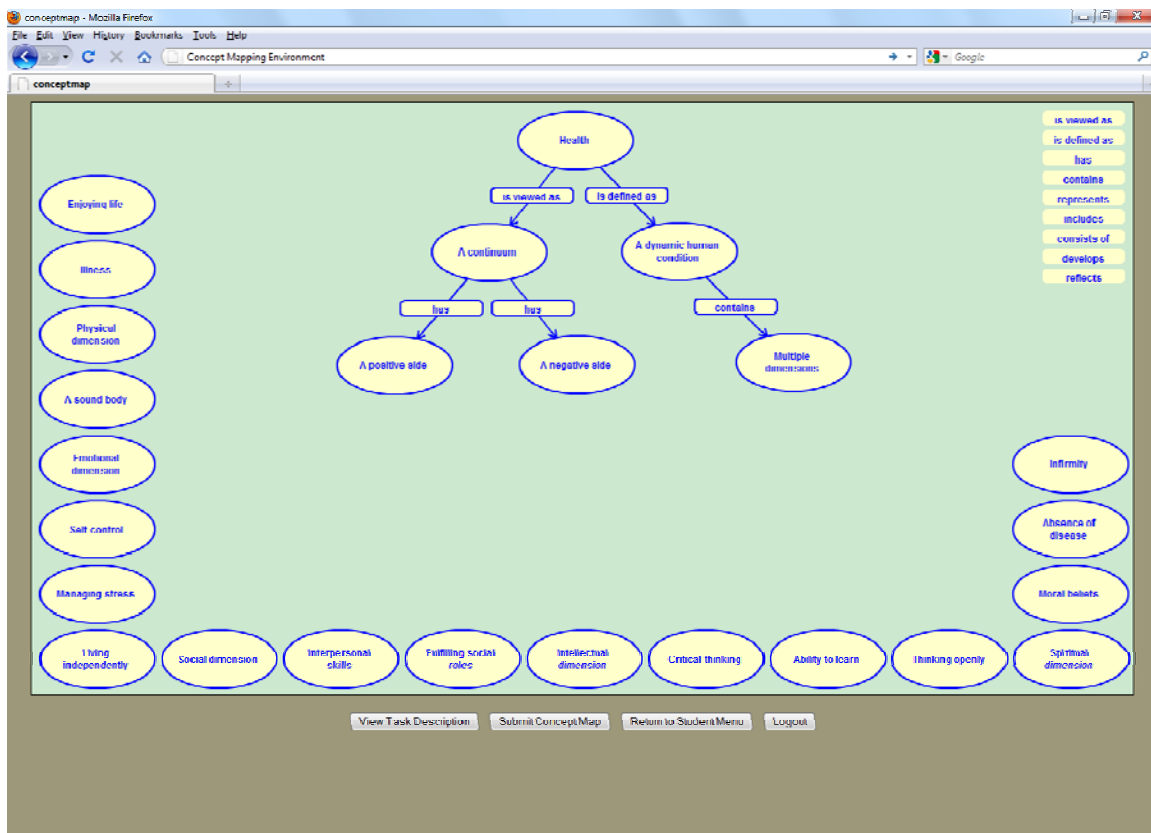


Figure 25. The concept map creation web page.

Instructor input. An instructor uses existing software, including Excel and Navicat, to create tasks, submit tasks, edit existing tasks, assign tasks to students, and obtain reports of student performance. An assessment task can be created with Excel and transferred into the database with Navicat. An instructor may input or edit the task description, node terms, linking phrases, and assessment criterion propositions through interacting with the database with Excel and Navicat.

A programmer coded and implemented the database, user interface, and system functionality. The researcher tested the system on various computers and platforms and communicated with the programmer to fix bugs, and update and improve the system. The system was tested on Windows and Mac OS X with popular Internet browsers, including Firefox, Internet Explorer, and Safari. A faculty member in the Instructional Design and

Technology program at Virginia Tech supervised and coordinated the system development. After the initial version of the Assessment Agent System was developed, formative evaluations were conducted to detect possible problems in the system by users.

The Formative Evaluation of the Assessment Agent System

The formative evaluation in this study was one-to-one evaluation. One-to-one evaluation allows the researcher and participants to have opportunities to talk face-to-face with each other to produce fruitful evaluation information (Tessmer, 1993). The purpose of one-to-one evaluation in this study was twofold. First, it evaluated user interface and system functionality. Second, it tested the materials used in the evaluation. One-to-one evaluation was conducted twice in the study. Results of one-to-one evaluation were analyzed after each evaluation and used to improve the user interface, system functions and materials used in the evaluation.

The First One-To-One Evaluation

Participants and materials. The Participants in the first one-to-one evaluation were two graduate students at Virginia Tech. One was a major in curriculum and instruction; another was a major in mathematics. Materials used in the evaluation included student directions for evaluating the assessment agent system, task description, and reading material.

Procedure. In each one-to-one evaluation session, only one participant was involved. First, the participant was introduced to the research project and the purpose of the evaluation activities. Second, the participant began to read the provided materials and carried out a concept map assessment task by following the directions in the materials. Third, after the participant obtained the assessment report, the researcher explained the

content in the assessment report. Fourth, the participant completed the evaluation questionnaire. The researcher observed the whole evaluation process, communicated with the participant and took notes to record participants' comments and suggestions during the evaluation process. The first session lasted for 1 hour and 40 minutes. The second session lasted for 2 hours.

Data collection. Participants' comments and suggestions in the evaluation process were recorded by the researcher through taking notes. After completing the system tryout process, each of the participants completed a system evaluation questionnaire on his computer. The researcher developed this questionnaire based on the usability checklist for computer-based testing programs created by Harmes and Parshall (2000). The questionnaire includes 28 items, which are divided into four sections: participant background, visual clarity, system functionality, consistency, and error prevention and correction.

Results of the first one-to-one evaluation. In general, the user interface and system functionalities worked well, and the materials were easy to understand and follow. The major problems related to the system were (a) moving nodes on the right edge in the user interface, (b) deleting linking phrases and dropping new ones, and (c) the student menu webpage design. Participants suggested reducing the length of the reading material, and revising some items of the evaluation questionnaire. More details of the first one-to-one evaluation can be found in Appendix B.

After completing the first one-to-one evaluation, the system was thoroughly tested again. The researcher communicated with the faculty advisor and programmer to improve the system. The student directions and task description documents were revised.

The Second One-To-One Evaluation

The participant in the second one-to-one evaluation was one psychology major graduate student at Virginia Tech. Materials used in the evaluation included student directions for evaluating the assessment agent system, task description, and reading material.

The procedure of the second one-to-one evaluation was similar to that of the first one-to-one evaluation. The participant was introduced to the basic knowledge of concept maps and trained how to use the system. The session lasted for 1 hour 15 minutes. The data collection method in the second one-to-one evaluation was the same as that in the first one-to-one evaluation.

The major problems revealed in the second one-to-one evaluation were a problem of deleting links and a problem of reporting function. More details of the second one-to-one evaluation can be found in Appendix B.

After the second one-to-one evaluation, the user interface was redesigned and redeveloped. The reading material was condensed to one page. Node terms and linking phrases of the concept map assessment task, as well as criterion propositions, were refined. Some items in the questionnaire were revised. The student directions document was also revised based on the improvement of the system.

Chapter Four: Results

This study addressed three research questions. Responses to those questions were derived partially from a review of relevant literature. This review was followed by the design, development, and evaluation of the Assessment Agent System based on information obtained from the literature review. The results of this process are presented below.

Research Question 1. What functionalities are necessary for large-scale assessment based on concept maps in a software agent-oriented system?

Through the literature review, eight categories of functionality were identified. These functionalities are user-system interaction, user management, task authoring and management, assessment delivery, task presentation, response collection, automatic assessing with feedback, and reporting. Details of these functionalities can be found in Table 16 in chapter three. These functionalities were developed and carried out by five types of agents. Table 31 presents the developed functionalities of the Assessment Agent System.

Table 31

The Developed Functionalities of the Assessment Agent System

Category of functionality	Developed system functionality
User-system interaction	Through the student interface, the system allows students to input user name and password, select course and assessment task, create and submit concept maps, and request assessment reports. Assessment reports can be displayed on computer screens. Through the instructor interface, the system allows instructors to input user name and password, student account information, and assessment task, and request assessment reports. Assessment reports can be displayed on computer screens.
User management	The system accepts, maintains, and updates user data in to the database. The system allows authorized users to log in the system.
Task authoring and management	The system accepts and stores student account data and assessment task data in the database.
Assessment delivery	The system delivers assessment tasks to students' computers via the Internet.
Task presentation	After the assessment task is delivered to a student's computer, course and task lists can be displayed on the student's computer screen for selection. The task description, nodes and linking phrases can be displayed on the student's screen through the interaction between the student and the student interface.
Response collection	After a student creates a concept map and clicks the "Submit Concept Map" button on the student interface, the system extracts node terms and linking phrases of the student map, and inserts the node terms and linking phrases in the student map tables of the database.
Automatic assessing with feedback	The system compares student node terms and linking phrases with expert-created node terms and linking phrases, and stores assessment results in the database.
Reporting	The system queries assessment results in the database and generates and delivers reports on student performances.

Research Question 2. What system architecture could be appropriate for large-scale assessment based on concept maps in a software agent-oriented system?

The literature review suggested multi-agent structures would be appropriate to create sophisticated, distributed computing systems. The Assessment Agent System in this study is composed of five types of agents. They are instructor agent, student agent, management agent, assessment agent, and reporting agent. Each of these agents was designed to possess different capabilities. Agents work together to provide the necessary functions through communication and cooperation.

The operability of the five types of agents was examined as part of a group evaluation and all agents functioned as designed. The results of the group evaluation and the observations regarding the agents appears later in this chapter.

Research Question 3. What design could provide a valid assessing method for large-scale assessment based on concept maps in a software agent-oriented system?

The literature review suggested that it should be possible to automate the process of assessing student concept maps when node terms and linking phrases are provided if student responses can be compared with assessment criteria on a proposition by proposition basis. However, the usefulness of the proposition-comparing method depends heavily on the accuracy and thoroughness of the criterion propositions. In practice, students may create some propositions that do not match criterion propositions, but that may be considered correct.

This study resulted in the development of an agent-oriented assessment tool based on concept maps. Students using the tool are presented with node terms and linking

phrases and asked to create a set of propositions demonstrating their understanding of a particular topic.

The remainder of this chapter presents the procedural results associated with the development of this tool. The procedural results were derived from a summative evaluation of the Assessment Agent System. The summative evaluation method employed in this study was group evaluation. Group evaluation involves more participants and may reveal more diverse opinions.

The Summative Evaluation of the Assessment Agent System

The purpose of the summative evaluation was for students to evaluate the improved user interface as well as system functionality and identify their concerns and opinions.

Participants

Participants in the group evaluation were 34 students at Virginia Tech, 28 graduate students and 6 undergraduate students. They were recruited through email and face-to-face invitation. Their study disciplines were highly varied and included science, technology, engineering, education, business, social sciences and humanities. Twenty-nine participants had experience taking computer-based assessments. Fifteen participants of the thirty-four had experience evaluating computer-based educational products.

Materials and Settings

Materials used in the group evaluation included student directions for evaluating the assessment agent system, task description, and reading material (see Appendix D, Appendix E, and Appendix F). The student directions document included fourteen steps that guided participants to perform the whole process of evaluation activities, from

electronically signing the consent form, to learning basic knowledge of concept maps, to carrying out a concept map assessment task, to completing an evaluation questionnaire. The reading material was about the concept of health and was adapted from a college textbook. The task description explained the concept map assessment task and listed twenty-five node terms and nine linking phrases. It was developed by the researcher based on the content of the reading material. The evaluation was conducted in labs. Evaluation settings were Windows XP or Windows 7, Firefox 3.6, Adobe Flash player 10, and screen resolutions of 1440 by 900 or higher.

Data Collection

After finishing the system tryout process, each of the participants completed an evaluation questionnaire on his or her computer. The researcher developed this questionnaire based on the usability checklist for computer-based testing programs created by Harmes and Parshall (2000). The questionnaire included 29 items, which were divided into four sections: participant background, visual clarity, system functionality, consistency, and error prevention and correction (see Appendix G).

Procedure

First, participants were provided with materials used in the evaluation, introduced to the research project, and told that they were invited to evaluate the Assessment Agent System from the perspective of students. Second, the researcher explained the evaluation procedure and the basic knowledge of concept maps and demonstrated how to use the system. Third, participants began to read the provided materials and carried out a concept map assessment task by following the directions in the materials. Fourth, after

participants obtained their assessment reports, the researcher explained the content in the assessment reports. Fifth, participants completed the evaluation questionnaire.

Each session, including briefing, training, carrying out the concept map assessment task, and completing the system evaluation questionnaire, lasted between 50 minutes and 1 hour and 20 minutes.

Results of the Group Evaluation

Visual clarity. Most participants rated “Always” or “Most of the time” for the items related to visual clarity (see Appendix H). Suggestions for improving the user interface focused on the concept map creation space and color options. The concept map assessment task contained 25 nodes. As more and more propositions were created, the area for building the concept map on the screen became more and more crowded. Participants suggested enlarging the concept map working space or making the size of nodes smaller.

System functionality. Most participants rated “Always” or “Most of the time” for the items related to button functioning and concept map creation (see Appendix H). All buttons worked well, and operations for constructing concept maps were intuitive and easy. Participants encountered a problem when they deleted links. Sometimes, the system did not respond to participants’ clicking the linking lines for deleting links. Another issue that concerned most participants was the content of their assessment report, especially those propositions that were assessed as “different proposition.” They thought that some of those “different” propositions were correct.

Consistency. All participants rated “Always” or “Most of the time” for the items related to consistency, including icons, color arrangement, placement of information, and system responds to user actions (see Appendix H).

Error prevention and correction. Most participants rated “Always” or “Most of the time” for the items related to error prevention and correction, including mistake correction, viewing the entire concept map on the screen, and checking their concept map before submitting (see Appendix H). Their concerns focused on the small space for creating concept maps. In such an environment, participants complained that nodes and links overlapped each other when more and more propositions were created. One participant suggested that when a student submitted a concept map, a warning box be displayed to ask if the student was sure he or she was ready to submit the concept map.

More details of the group evaluation can be found in Appendix C.

Chapter Five: Discussions

Discussion of Research Results

System Functionality

Through the literature review, eight categories of functionality were identified. They are user-system interaction, user management, task authoring and management, assessment delivery, task presentation, response collection, automatic assessing with feedback, and reporting. After analyzing data coupling, these functionalities were grouped and assigned to five types of agents of the Assessment Agent System, including instructor agent, student agent, management agent, assessment agent, and reporting agent. Software agents in the system cooperatively provided the eight categories of functionality. The Assessment Agent System was developed and tested, and it possessed the designed functionalities.

To evaluate system functionalities, a concept map assessment task and assessment criteria were developed and input into the system, student accounts were created in the system, and one-to-one evaluation as well as group evaluation were conducted. With the system, users could be managed and assessment tasks could be created and updated. Student users were able to interact with the system to request that an assessment task be presented on the computer screen, to create and submit concept maps, and to obtain assessment reports. The results of evaluation indicate that the functionalities designed and developed for the Assessment Agent System are necessary and meet users' needs for large-scale assessment based on concept maps.

Existing computer-based tools for assessing concept maps, such as Mind Matrix (Dennis et al., 2002), Human Performance Knowledge Mapping Tool (Chung et al.,

2006), and Concept Connector Tools (Luckie et al., 2003), possess a variety of functions. These functionalities fall into the eight categories of functionality identified in this study. The findings about necessary functionalities of this study are in agreement with those of other studies on computer-based concept map assessment.

Students who evaluated the Assessment Agent System suggested more functions for the system. Most of the suggestions related to user-system interaction, automatic assessing, and feedback. For example, they hoped that the system could provide more advanced functions that allow them to create and modify concept maps more easily. They were also concerned about the concept map assessment and wanted more personal feedback on their concept maps. In the future, the Assessment Agent System can be improved on the functions of user-system interaction, automatic assessing, and feedback. However, this is beyond the scope of this study.

System Architecture

The Assessment Agent System is composed of five types of agents. They are instructor agent, student agent, management agent, assessment agent, and reporting agent. Each of these agents was designed to possess different capabilities. Agents work together to provide the necessary functions through communication and cooperation.

The architecture of a computer system for large-scale assessment affects its performance, maintenance, and usability (Higgins, Hegazy, Symeonidis, & Tsintsifas, 2003). Architectures of existing computer systems for assessing concept maps were designed as either client-server architecture or modeled as a set of modules (e.g., Chung et al., 2006; Dennis et al., 2002; Gouli et al., 2004; Luckie et al., 2003). Compared to these structures, the assessment agent system architecture was modeled as a group of

software agents, which is a higher level abstraction and a more reusable structure (Bergenti, 2003). Although the assessment agent system architecture was designed based on assessing concept maps, the five-types-of-agents architecture may serve as reference architecture for designing other computer-based assessment systems.

Concept Map Assessment Method

When node terms and linking phrases are provided, assessing student concept maps can be automated by comparing student concept maps with assessment criteria, proposition by proposition. However, the validity of the proposition-comparing method depends on the criterion propositions. In practice, students may create some propositions that do not match criterion propositions, but that may be considered correct.

To overcome this problem, the assessment process may include two phases. The assessment begins with expert-created propositions as the assessment criteria. In the first phase, the assessment agent assesses the first set of student-created propositions, and the assessment result includes three categories: correct-proposition, different-proposition, and missing-proposition. However, the assessment result is temporary.

In the second phase, an expert examines the “different” propositions of the student concept map. The “different” propositions are identified as either correct or incorrect. In this way, the final assessment results are produced, partly done by the assessment agent and partly done by an expert.

Next, the expert adds the identified correct and incorrect propositions (assessed as “different” by the assessment agent) of the student concept map to the assessment criterion propositions pool. The assessment criteria are improved and include expert-identified correct and incorrect propositions.

Subsequent student concept maps will be assessed against the improved assessment criteria, and assessment results generated by the assessment agent will include four categories: correct-proposition, incorrect-proposition, different-proposition, and missing-proposition. The expert continually reviews the “different” propositions of the student concept map, produces final assessment results and improves the assessment criteria.

As the process continues, the pool of correct and incorrect propositions in assessment criteria becomes larger and larger and covers more and more correct propositions, either created by experts or students, and incorrect propositions created by students. Although the assessment process relies on an expert to improve the assessment criteria at the beginning, once the proposition pool in the database encompasses all possible propositions, the assessment agent can automatically produce valid assessment results.

For incorrect propositions in student concept maps, assessment results may further indicate the nature of the incorrectness of nodes and linking phrases. Table 32 presents seven patterns of incorrect propositions in student concept maps compared to expert propositions.

Table 32

Patterns of Incorrect Proposition in Student Concept Maps

Pattern Number	Source node	Target node	Linking phrase	Assessment result
1	√	√	×	Incorrect linking phrase
2	√	×	√	Incorrect target node
3	√	×	×	Incorrect target node Incorrect linking phrase
4	×	√	√	Incorrect source node
5	×	√	×	Incorrect source node Incorrect linking phrase
6	×	×	√	Incorrect source node Incorrect target node
7	×	×	×	Incorrect source node Incorrect target node Incorrect linking phrase

Summary of Significant Contributions

The dissertation research has made several contributions to the area of educational technology studies. First, this study yields a multi-agent system architecture for designing computer-based, large-scale assessment systems. This system architecture can be used as reference architecture for developing other assessment systems. Second, this research project demonstrated an approach to designing complex educational software applications with agent-oriented methods. Third, the assessment agent system is a domain independent tool for assessing comprehensive understanding based on concept maps.

References

- Anohina, A., Graudina, V., & Grundspenkis, J. (2007). Using concept maps in adaptive knowledge assessment. In G. Magyar, G. Knapp, W. Wojtkowski, W. G. Wojtkowski, & J. Zupančič (Eds.), *Advances in information systems development: New methods and practice for the networked society* (Vol. 1, pp. 469-479). New York: Springer-Verlag.
- Anohina, A., & Grundspenkis, J. (2006). Prototype of multiagent knowledge assessment system for support of process oriented learning. In O. Vasilecas, J. Eder, & A. Caplinskas (Eds.), *Proceedings of the 2006 seventh international Baltic conference on databases and information systems* (pp. 211-219). Piscataway, NJ: IEEE.
- Anohina, A., & Grundspenkis, J. (2007). A concept map based intelligent system for adaptive knowledge assessment. In O. Vasilecas, J. Eder, & A. Caplinskas (Eds.), *Databases and information systems IV: Selected papers from the seventh international Baltic conference DB&IS'2006* (pp. 263-276). Amsterdam: IOS Press.
- Baylor, A. L. (2002). Expanding preservice teachers' metacognitive awareness of instructional planning through pedagogical agents. *Educational Technology Research and Development*, 50(2), 5-22.
- Baylor, A. L. & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education*, 15(2), 95-115.

- Bergenti, F. (2003). A discussion of two major benefits of using agents in software development. In P. Petta, R. Tolksdorf & F. Zambonelli (Eds.), *Engineering societies in the agents world III* (pp. 1-12). Berlin, Germany: Springer-Verlag.
- Bergenti, F., Gleizes, M.-P., & Zambonelli, F. (Eds.). (2004). *Methodologies and software engineering for agent systems: The agent-oriented software engineering handbook*. Boston: Kluwer Academic Publishers.
- Bradshaw, J. M. (1997). An introduction to software agents. In J. M. Bradshaw (Ed.), *Software agents* (pp. 3-46). Cambridge, MA: MIT Press.
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8, 203-236.
- Bull, J., & McKenna, C. (2004). *Blueprint for computer-assisted assessment*. New York: RoutledgeFalmer.
- Burkhardt, H., & Pead, D. (2003). Computer-based assessment: A platform for better tests? In C. Richardson (Ed.), *Whither assessment?* (pp. 133-148). London: Qualifications and Curriculum Authority.
- Cañas, A. J., Coffey, J. W., Carnot, M. J., Feltovich, P., Hoffman, R. R., Feltovich, J., & Novak, J. D. (2003). *A summary of literature pertaining to the use of concept mapping techniques and technologies for education and performance support*. Retrieved from Institute for Human and Machine Cognition website: <http://www.ihmc.us/users/acanas/Publications/ConceptMapLitReview/IHMC%20Literature%20Review%20on%20Concept%20Mapping.pdf>

- Chang, K. E., Sung, Y. T., & Chen, S. F. (2001). Learning through computer-based concept mapping with scaffolding aid. *Journal of Computer Assisted Learning*, 17, 21-33.
- Chung, G. K. W. K., & Baker, E. L. (1997). *Year 1 technology studies: Implications for technology in assessment* (CSE Tech. Rep. No. 459). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Chung, G. K. W. K., Baker, E. L., Brill, D. G., Sinha, R., Saadat, F., & Bewley, W. L. (2006). *Automated assessment of domain knowledge with online knowledge mapping* (CSE Tech. Rep. No. 692). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Chung, G. K. W. K., Baker, E. L., & Cheak, A. M. (2002). *Knowledge Mapper authoring system prototype* (CSE Tech. Rep. No. 575). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Clarke, J. H. (1990). *Patterns of thinking: Integrating learning skills in content teaching*. Boston: Allyn and Bacon.
- Collins, A. M., & Quillian, M. R. (1972). How to make a language user. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 309-351). New York: Academic Press.
- Cronin, H., Meadows, D., & Sinatra, R. (1990). Integrating computers, reading, and writing across the curriculum. *Educational Leadership*, 48(1), 57-62.

- Dansereau, D. F., & Holley, C. D. (1982). Development and evaluation of a text mapping strategy. In A. Flammer & W. Kintsch (Eds.), *Discourse processing* (pp. 536-554). Amsterdam: North Holland Publishing.
- Deloach, S. A., Wood, M. F., & Sparkman, C. H. (2001). Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 231-258.
- Dennis, R. A., Stamboltsian, V., & Huang, K. (2002). The Mind-Matrix: A learning, instruction, and assessment system. *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*. Retrieved from http://www.asee.org/acPapers/2002-993_Final.pdf
- d'Inverno, M., & Luck, M. (2004). *Understanding agent systems* (2nd ed.). Berlin, Germany: Springer-Verlag.
- Edelson, D. C. (2002). Design research: What we learn when we engage in design. *The Journal of the Learning Sciences*, 11, 105-121.
- Etzioni, O., & Weld, D. S. (1995). Intelligent agents on the Internet: Fact, fiction, and forecast. *IEEE Expert*, 10(4), 44-49.
- Fisher, K. M. (1990). Semantic networking: The new kid on the block. *Journal of Research in Science Teaching*, 27, 1001-1018.
- Fisher, K. M. (2000). SemNet software as an assessment tool. In J. J. Mintzes, J. H. Wandersee, & J. D. Novak (Eds.), *Assessing science understanding: A human constructivist view* (pp. 197-221). San Diego, CA: Academic Press.

- Fisher, K. M., Faletti, J., Patterson, H., Thornton, R., Lipson, J., & Spring, C. (1990). Computer-based concept mapping. *Journal of College Science Teaching*, 19, 347-352.
- Foner, L. (1993). *What's an agent, anyway? A sociological case study* (Agents Memo 93-01). Retrieved from Massachusetts Institute of Technology Media Lab website: <http://foner.www.media.mit.edu/people/foner/Reports/Julia/Agents--Julia.pdf>
- Franklin, S., & Graesser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. In J. P. Müller, M. J. Wooldridge, & N. R. Jennings (Eds.), *Intelligent agents III: Agent theories, architectures, and languages* (pp. 21-35). Berlin, Germany: Springer-Verlag.
- Giotopoulos, K. C., Alexakos, C. E., Beligiannis, G. N., & Likothanassis, S. D. (2007). Integrating computational intelligence techniques and assessment agents in e-learning environments. *International Journal of Computational Intelligence*, 3(4), 328-337. Retrieved from <http://www.waset.org/ijci/v3/v3-4-39.pdf>
- Gouli, E., Gogoulou, A., Papanikolaou, K., & Grigoriadou, M. (2004). COMPASS: An adaptive web-based concept map assessment tool. In A. J. Cañas, J. D. Novak, & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*. Retrieved from <http://cmc.ihmc.us/papers/cmc2004-128.pdf>
- Harmes, J. C., & Parshall, C. G. (2000, November). *An iterative process for computerized test development: Integrating usability methods*. Paper presented at the annual meeting of the Florida Educational Research Association, Tallahassee, FL.

- Henderson-Sellers, B., & Giorgini, P. (Eds.). (2005). *Agent-oriented methodologies*. Hershey, PA: Idea Group Publishing.
- Herl, H. E., Baker, E. L., & Niemi, D. (1996). Construct validation of an approach to modeling cognitive structure of U.S. history knowledge. *The Journal of Educational Research*, 89, 206-218.
- Herl, H. E., O'Neil, H. F., Chung, G. K. W. K., & Schacter, J. (1999). Reliability and validity of a computer-based knowledge mapping system to measure content understanding. *Computers in Human Behavior*, 15, 315-333.
- Higgins, C., Hegazy, T., Symeonidis, P., & Tsintsifas, A. (2003). The CourseMarker CBA system: Improvements over Ceilidh. *Education and Information Technologies*, 8(3), 287-304.
- Hoeft, R. M., Jentsch, F. G., Harper, M. E., Evans, A. W., III, Bowers, C. A., & Salas, E. (2003). TPL-KATS - concept map: A computerized knowledge assessment tool. *Computers in Human Behavior*, 19, 653-657.
- Hyerle, D. (1996). *Visual tools for constructing knowledge*. Alexandria, VA: Association for Supervision and Curriculum Development.
- Hyerle, D. (2000). *A field guide to using visual tools*. Alexandria, VA: Association for Supervision and Curriculum Development.
- Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117, 277-296.
- Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4), 35-41.

- Jennings, N. R., & Wooldridge, M. (1998a). Applications of intelligent agents. In N. R. Jennings & M. Wooldridge (Eds.), *Agent technology: Foundations, applications, and markets* (pp. 3-28). Berlin, Germany: Springer-Verlag.
- Jennings, N. R., & Wooldridge, M. J. (Eds.). (1998b). *Agent technology: Foundations, applications, and markets*. Berlin, Germany: Springer-Verlag.
- Johnson, W. L., Rickel, J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11, 47-78.
- Kay, A. (1984, September). Computer software. *Scientific American*. 251(3), 53-59.
- Kim, S., Herzog, O., Lockemann, P., & Spaniol, O. (Eds.). (2006). *Multiagent engineering: Theory and applications in enterprises*. Berlin, Germany: Springer.
- Klein, D. C. D., Chung, G. K. W. K., Osmundson, E., Herl, H. E., & O'Neil, H. F., Jr. (2002). *The validity of knowledge mapping as a measure of elementary students' scientific understanding* (CSE Tech. Rep. No. 557). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Klusch, M. (2001). Information agent technology for the Internet: A survey. *Data & Knowledge Engineering*, 36(3), 337-372.
- Lafifi, Y., & Bensebaa, T. (2007). Learners' assessment in a collaborative learning system. *Asian Journal of Information Technology* 6(2), 145-153. Retrieved from <http://www.medwelljournals.com/fulltext/ajit/2007/145-153.pdf>
- Lai, K. R., & Lan, C. H. (2006). Modeling peer assessment as agent negotiation in a computer supported collaborative learning environment. *Journal of Educational*

- Technology & Society*, 9(3), 16-26. Retrieved from
http://www.ifets.info/journals/9_3/3.pdf
- Lambiotte, J. G., Dansereau, D. F., Cross, D. R., & Reynolds, S. B. (1989).
 Multirelational semantic maps. *Educational Psychology Review*, 1, 331-367.
- Laurel, B. (1997). Interface agents: Metaphors with character. In J. M. Bradshaw (Ed.),
Software agents (pp. 67-77). Cambridge, MA: MIT Press.
- Lin, F. O. (Ed.). (2005). *Designing distributed learning environments with intelligent
 software agents*. Hershey, PA: Information Science Publishing.
- Lin, F., Esmahi, L., & Poon, L. (2005). Integrating agents and web services into adaptive
 distributed learning environments. In F. O. Lin (Ed.), *Designing distributed
 learning environments with intelligent software agents* (pp. 184-217). Hershey,
 PA: Information Science Publishing.
- Lockemann, P. C. (2006). Agents. In S. Kirn, O. Herzog, P. Lockemann, & O. Spaniol
 (Eds.), *Multiagent engineering: Theory and applications in enterprises* (pp. 17-
 33). Berlin, Germany: Springer.
- Luck, M., Ashri, R., & d'Inverno, M. (2004). *Agent-based software development*. Boston:
 Artech House.
- Luckie, D. B., Batzli, J. M., Harrison, S., & Ebert-May, D. (2003). C-TOOLS: Concept-
 connector tools for online learning in science. *International Journal of Learning*,
 10, 1051-1067.
- Merriam-Webster's collegiate dictionary* (11th ed.). (2004). Springfield, MA: Merriam-
 Webster.

- McClure, J. R., Sonak, B., & Suen, H. K. (1999). Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Journal of Research in Science Teaching*, 36, 475-492.
- Mind Matrix editor authoring manual*. (2002). Retrieved from <http://www.mind-matrix.org/pdf/AuthorManual.pdf>
- Müller, J. P. (1996). *The design of intelligent agents: A layered approach*. Berlin, Germany: Springer-Verlag.
- Müller, J. P. (1998). Architectures and applications of intelligent agents: A survey. *The Knowledge Engineering Review*, 13(4), 353-380.
- Nesbit, J. C., & Adesope, O. O. (2006). Learning with concept and knowledge maps: A meta-analysis. *Review of Educational Research*, 76, 413-448.
- Novak, J. D. (1990). Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching*, 27, 937-949.
- Novak, J. D. (1995). Concept mapping: A strategy for organizing knowledge. In S. M. Glynn & R. Duit (Eds.), *Learning science in the schools: Research reforming practice* (pp. 229-245). Mahwah, NJ: Lawrence Erlbaum Associates.
- Novak, J. D. (1998). *Learning, creating and using knowledge: Concept maps as facilitative tools in schools and corporations*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Novak, J. D., & Gowin, D. R. (1984). *Learning how to learn*. New York: Cambridge University Press.
- Nwana, H. S. (1996). Software agents: An overview. *The Knowledge Engineering Review*, 11(3), 205-244.

- O'Donnell, A. M., Dansereau, D. F., & Hall, R. H. (2002). Knowledge maps as scaffolds for cognitive processing. *Educational Psychology Review*, 14, 71-86.
- Padgham, L., & Winikoff, M. (2004). *Developing intelligent agent systems: A practical guide*. Chichester, West Sussex, England: John Wiley & Sons.
- Richey, R. C., & Klein, J. D. (2007). *Design and development research: Methods, strategies, and issues*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Richey, R. C., Klein, J. D., & Nelson, W. A. (2004). Developmental research: Studies of instructional design and development. In D. H. Jonassen (Ed.), *Handbook of research on educational communications and technology* (2nd ed., pp. 1099-1130). Mahwah, NJ: Lawrence Erlbaum Associates.
- Ruiz-Primo, M. A. (2004). Examining concept maps as an assessment tool. In A. J. Cañas, J. D. Novak, & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*. Retrieved from <http://cmc.ihmc.us/papers/cmc2004-036.pdf>
- Ruiz-Primo, M. A., Schultz, S. E., & Shavelson, R. J. (1997). *Concept map-based assessment in science: Two exploratory studies* (CSE Tech. Rep. No. 436). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Ruiz-Primo, M. A., Schultz, S. E., Li, M., & Shavelson, R. J. (2001). Comparison of the reliability and validity of scores from two concept-mapping techniques. *Journal of Research in Science Teaching*, 38, 260-278.

- Ruiz-Primo, M. A., & Shavelson, R. J. (1996). Problems and issues in the use of concept maps in science assessment. *Journal of Research in Science Teaching*, 33, 569-600.
- Ruiz-Primo, M. A., Shavelson, R. J., Li, M., & Schultz, S. E. (2001). On the validity of cognitive interpretations of scores from alternative concept-mapping techniques. *Educational Assessment*, 7, 99-141.
- Ruiz-Primo, M. A., Shavelson, R. J. & Schultz, S. E. (1997). *On the validity of concept map-based assessment interpretations: An experiment testing the assumption of hierarchical concept maps in science* (CSE Tech. Rep. No. 455). Los Angeles: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).
- Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Upper Saddle River, NJ: Prentice Hall.
- Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
- Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
- Shepard, L.A. (1989). Why we need better assessments. *Educational Leadership*, 46(7), 4-9.
- Sterling, L. S., & Taveter, K. (2009). *The art of agent-oriented modeling*. Cambridge, MA: MIT Press.
- Sycara, K. P. (1998, Summer). Multiagent systems. *AI Magazine*, 19(2), 79-92.

- Tecuci, G. (1998). *Building intelligent agents: An apprenticeship multistrategy learning theory, methodology, tool and case studies*. San Diego, CA: Academic Press.
- Tecuci, G., & Keeling, H. (1999). Developing an intelligent educational agent with disciple. *International Journal of Artificial Intelligence in Education*, 10, 221-237.
- Tessmer, M. (1993). *Planning and conducting formative evaluations: Improving the quality of education and training*. London: Kogan Page.
- Tindal, G., & Haladyna, T. M. (Eds.). (2002). *Large-scale assessment programs for all students: Validity, technical adequacy, and implementation*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Tsai, C. C., Lin, S. S. J., & Yuan, S. M. (2001). Students' use of web-based concept map testing and strategies for learning. *Journal of Computer Assisted Learning*, 17, 72-84.
- van den Akker, J., Cravemeijer, K., McKenney, S., & Nieveen, N. (Eds.). (2006). *Educational design research*. New York: Routledge.
- Weiss, G. (Ed.). (1999). *Multiagent systems: A modern approach to distributed artificial intelligence*. Cambridge, MA: MIT Press.
- Wooldridge, M. (2002). *An introduction to multiagent systems*. Chichester, West Sussex, England: John Wiley & Sons.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152.

Wooldridge, M., Jennings, N. R., & Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3, 285-312.

Yin, Y., Vanides, J., Ruiz-Primo, M. A., Ayala, C. C., & Shavelson, R. J. (2005). Comparison of two concept-mapping techniques: Implications for scoring, interpretation, and use. *Journal of Research in Science Teaching*, 42, 166-184.

Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2003). Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3), 317-370.

Appendix A: Scenarios

Assessment Task Creation Scenario

The assessment task creation scenario illustrates steps responding to an instructor creating an assessment task. The steps in the scenario are presented in Table A1.

Table A1

Steps in the Assessment Task Creation Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide text areas for receiving inputs from users	User-system interaction	
2	Percept	Entering text	User-system interaction	
3	Action	Text display	User-system interaction	
4	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
5	Percept	Task submission request	User-system interaction	
6	Goal	Identify users	User management	Uses: Records in user database tables
7	Goal	Accept and store data in a database	Task authoring and management	Produces: Task description, node terms, linking phrases, and assessment criteria
8	Action	Task submission confirmation	Task authoring and management	

Task Assignment Scenario

The task assignment scenario illustrates steps responding to an instructor assigning an assessment task to students. The steps in the scenario are presented in Table A2.

Table A2

Steps in the Task Assignment Scenario

Number	Type	Name	Functionality	Data
1	Percept	Task assigning request	User-system interaction	
2	Goal	Identify users	User management	Uses: Records in user database tables
3	Goal	Provide course and task lists for selection	Task authoring and management	Uses: Records in task database tables
4	Percept	Item selection	User-system interaction	
5	Goal	Provide student name lists for selection	Task authoring and management	Uses: Records in user database tables
6	Percept	Item selection	User-system interaction	
7	Goal	Accept and store data in a database	User management	Produces: Records in user database tables
8	Action	Task assigning confirmation	Task authoring and management	

Task Presentation Scenario

The task presentation scenario illustrates steps responding to a student's task presentation request. The steps in the scenario are presented in Table A3.

Table A3

Steps in the Task Presentation Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
2	Percept	Task presentation request	User-system interaction	
3	Goal	Identify users	User management	Uses: Records in user database tables
4	Goal	Provide course and task lists for selection	Task presentation	Uses: Records in task database tables
5	Percept	Item selection	User-system interaction	
6	Goal	Deliver assessment tasks to users via the Internet	Assessment delivery	Uses: Records in task database tables
7	Action	Task display	Task presentation	Uses: Task description, node terms and linking phrases

Student Map Creation Scenario

The student map creation scenario illustrates steps responding to a student creating a concept map. The steps in the scenario are presented in Table A4.

Table A4

Steps in the Student Map Creation Scenario

Number	Type	Name	Functionality	Data
1	Goal	Support operations of clicking, dragging, and dropping to build concept maps	User-system interaction	
2	Percept	Selecting and moving a node	User-system interaction	
3	Action	Node display	User-system interaction	
4	Percept	Selecting and moving a node	User-system interaction	
5	Action	Node display	User-system interaction	
6	Percept	Drawing a linking line	User-system interaction	
7	Action	Linking line display	User-system interaction	
8	Percept	Dragging and dropping a linking phrase	User-system interaction	
9	Action	Linking phrase display	User-system interaction	

Student Map Submission Scenario

The student map submission scenario illustrates steps responding to a student's request of submitting a student map. The steps in the scenario are presented in Table A5.

Table A5

Steps in the Student Map Submission Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
2	Percept	Student map submission request	User-system interaction	
3	Goal	Identify users	User management	Uses: Records in user database tables
4	Goal	Recognize concept maps	Response collection	
5	Goal	Accept and store responses in a database	Response collection	Produces: Records in student map database tables
6	Action	Student map submission confirmation	User-system interaction	

Concept Map Assessment Scenario

The concept map assessment scenario illustrates the steps of assessing a student concept map. The steps in the scenario are presented in Table A6.

Table A6

Steps in the Concept Map Assessment Scenario

Number	Type	Name	Functionality	Data
1	Goal	Notify student map reception	Response collection	
2	Goal	Compare student maps with assessment criteria	Automatic assessing with feedback	Uses: Records in student map database tables, records in task database tables Produces: assessing result and feedback
3	Goal	Send results and feedback to a database	Automatic assessing with feedback	Produces: Records in user database tables

Assessment report Request Scenario

The assessment report request scenario illustrates steps responding to a student's request for assessment reports. The steps in the scenario are presented in Table A7.

Table A7

Steps in the Assessment report Request Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
2	Percept	Assessment report request	User-system interaction	
3	Goal	Identify users	User management	Uses: Records in user database tables
4	Goal	Query assessment results	Reporting	Uses: Records in user database tables
5	Goal	Generate report on individual student performance	Reporting	Produces: Report
6	Goal	Deliver report on individual student performance	Reporting	
7	Action	Assessment report display	User-system interaction	

Student Performance Report Request Scenario

The student performance report request scenario illustrates steps responding to an instructor's request for a report on a group of student's performance. The steps in the scenario are presented in Table A8.

Table A8

Steps in the Student Performance Report Request Scenario

Number	Type	Name	Functionality	Data
1	Goal	Provide menus and/or buttons for receiving commands	User-system interaction	
2	Percept	Report request	User-system interaction	
3	Goal	Identify users	User management	Uses: Records in user database tables
4	Goal	Query assessment results	Reporting	Uses: Records in user database tables
5	Goal	Generate report on a group of student's performance	Reporting	Produces: Report
6	Goal	Deliver report on a group of student's performance	Reporting	
7	Action	Report display	User-system interaction	

Appendix B: One-To-One Evaluation

The First One-To-One Evaluation

Participants

Participants were two graduate students at Virginia Tech. One was a major in curriculum and instruction; another was a major in mathematics.

Materials and Settings

Materials used in the first one-to-one evaluation included student directions for evaluating the assessment agent system, task description, and reading material.

Participant A used a laptop computer with Windows XP, IE 8.0, Adobe Flash player 10, and screen resolution of 1024 by 768. Only electronic materials in DOC format were provided. Participant B used a laptop computer with Windows XP, Firefox 3.6, Adobe Flash player 10, and screen resolution of 1680 by 1050. Both electronic materials and printed materials were provided.

Procedure

In each one-to-one evaluation session, only one participant was involved. First, the participant was provided with materials used in the evaluation, including student directions for evaluating the assessment agent system, task description, and reading material. Second, the participant was introduced to the research project. Third, the participant was told that the purpose of the activities was to evaluate the Assessment Agent System from the perspective of a student, and to test the materials used in the evaluation. Fourth, the participant began to read the provided materials and carried out a concept map assessment task by following the directions in the materials. The major activities included logging into the system, selecting course and task, reading the task

description and reading material, constructing concept maps, submitting concept maps, and obtaining assessment reports. Fifth, after the participant obtained the assessment report, the researcher explained the content in the assessment report. Sixth, the participant completed the evaluation questionnaire. In order to test the student directions material, the two participants in the first one-to-one evaluation were not trained in the basic knowledge of concept maps and how to use the system.

The researcher observed the whole evaluation process, communicated with the participant and took notes to record participants' comments and suggestions during the evaluation process. After the participant finished all steps in the Student Directions for Evaluating the Assessment Agent System, the researcher asked some questions. These questions are:

- (a) Did you have any problems completing the evaluation process with the provided materials?
- (b) Were questions in the questionnaire easy to understand?
- (c) Do you have any suggestions about the materials used for evaluating the system?
- (d) Do you have any suggestions about the Assessment Agent System?

The first session lasted for 1 hour and 40 minutes. The second session lasted for 2 hours.

Data Collection

Participants' comments and suggestions in the evaluation process were recorded by the researcher through taking notes. After completing the system tryout process, each of the participants completed a system evaluation questionnaire on his computer. The

researcher developed this questionnaire based on the usability checklist for computer-based testing programs created by Harmes and Parshall (2000). The questionnaire includes 28 items, which are divided into four sections: participant background, visual clarity, system functionality, consistency, and error prevention and correction.

Results of the First One-To-One Evaluation

In general, the system functioned well, and the materials were easy to understand and follow. Participants were able to use the system without face-to-face training. One participant said that the system and reading material were interesting. The functions and interface are friendly; it looked like a digital game, and attracted him to complete the concept map. Colors are well designed. He learned many things by participating in the evaluation. Another participant said that he enjoyed the evaluation. However, the evaluation revealed some problems of the system, and participants provided suggestions to improve the system and the materials used in the evaluation.

Problems and Suggestions Related To the System

(a) Nodes moving problems. On the “Map Creation” webpage, if nodes on the left edge and bottom edge were not moved, some nodes on the right edge could not be moved.

(b) The problem of deleting a linking phrase and adding another one. After a link between two nodes was deleted and a new link was recreated between the same two nodes, a link phrase could not be dropped into the relationship box. More seriously, once the above problem occurred, other boxes either could not drop into a linking phrase or the dropped linking phrase filled in other boxes.

(c) A suggestion about the Student Menu webpage. One participant suggested separating “Taking Assessment” and “Request Assessment Result” into two different web pages. When the participant entered the Student Menu webpage for the first time, he tried to selected items under the “Request Assessment Result” section.

(d) One participant suggested that the system provide a zoom view function so that the participant could view the whole map with low resolution.

Problems and Suggestions Related To the Materials

(a) The researcher observed some problems related to the materials during the evaluation process. One participant ignored to sign the consent form and to read the reading material at the beginning. Another participant did not know what should to do after reading the reading material. Both participants ignored that there were inappropriate nodes and phrases in the provided nodes and phrases. The pop up window was blocked. The participant cannot view task description by clicking the View Task Description button.

(b) One participant suggested reducing the length of reading material. This will allow participants to focus on evaluating the system. The participant said that he was unwilling to spend time on reading long material during evaluating the system.

(c) One participant who was provided with only electronic materials suggested that the system include online reading material and student directions.

(d) Problems of the questionnaire. One participant said that item 7 and item 21 were not easy to understand. Another participant suggested revising item 2 and item 3.

The Second One-To-One Evaluation

Participant

The participant was one graduate student in the Department of Psychology at Virginia Tech.

Materials and Settings

Materials used in the second one-to-one evaluation included student directions for evaluating the assessment agent system, task description, and reading material. The participant used a laptop computer with Windows XP, Firefox 3.6, Adobe Flash player 10, and screen resolution of 1440 by 900 in an educational technology lab.

Procedure

First, the participant was provided with printed and electronic versions of materials used in the evaluation, including student directions for evaluating the assessment agent system, task description, and reading material. Second, the participant was introduced to the research project. Third, the participant was told that the purpose of the activities was to evaluate the Assessment Agent System from the perspective of a student, and to test the materials used in the evaluation. Any comments and suggestions were very welcome. Fourth, the researcher explained the evaluation procedure and the basic knowledge of concept maps and demonstrated how to use the system. Fifth, the participant began to read the provided materials and carried out a concept map assessment task by following the directions in the materials. The major activities included logging into the system, selecting course and task, reading the task description and reading material, constructing a concept map, submitting the concept map, and obtaining the assessment report. Sixth, after the participant obtained the assessment report, the

researcher explained the content in the assessment report. Seventh, the participant completed the evaluation questionnaire.

The researcher observed the whole evaluation process, communicated with the participant and took notes to record participants' comments and suggestions during the evaluation process. After the participant finished all the steps in the Student Directions for Evaluating the Assessment Agent System, the researcher asked some questions. These questions are:

- (a) Was the reading material appropriate for evaluating the system?
- (b) Was the concept map task appropriate for evaluating the system?
- (c) Do you have any suggestions about the materials used for evaluating the system?
- (d) Do you have any suggestions about the Assessment Agent System?

The session lasted for 1 hour 15 minutes. The data collection method in the second one-to-one evaluation was the same as that in the first one-to-one evaluation.

Results of the Second One-To-One Evaluation

Overall, the system functioned well, and the evaluation materials were easy to understand and follow. There was no problem in the evaluation materials. The reading material and concept map task were appropriate.

The participant said that the user interface was friendly; especially the nodes could be moved around on the screen to organize the structure of the concept map. He suggested using a float panel to display linking phrases. This would allow more words displayed on the screen. The participant further suggested providing synonyms of linking phrases that allow users with various vocabulary abilities to select appropriate words.

Problems and Possibility of Improvement

(a) A problem of deleting links. When deleting a link, a red line was displayed and persistent on the screen. After clicking the red line, a box displayed and indicated that there was a programming error.

(b) A problem of reporting function. If a relationship box was blank in a concept map and the concept map was submitted for assessing, the reporting function would not work.

(c) The participant suggested using different background colors on the assessment report webpage.

(d) With the screen resolution 1440 by 900, the entire concept map cannot be displayed on the screen. The map creation working area needed to be adjusted to fit the screen by increasing the horizontal length and decreasing the vertical length.

Appendix C: Group Evaluation

Participants

Participants in the group evaluation were 34 students at Virginia Tech, 28 graduate students and 6 undergraduate students. They were recruited through email and face-to-face invitation. Their study disciplines were highly varied and included science, technology, engineering, education, business, social sciences and humanities. Twenty-nine participants had experience taking computer-based assessments. Fifteen participants of the thirty-four had experience evaluating computer-based educational products.

Materials and Settings

Materials used in the group evaluation included student directions for evaluating the assessment agent system, task description, and reading material (see Appendix D, Appendix E, and Appendix F). The student directions document included fourteen steps that guided participants to perform the whole process of evaluation activities, from electronically signing the consent form, to learning basic knowledge of concept maps, to carrying out a concept map assessment task, to completing an evaluation questionnaire. The reading material was about the concept of health and was adapted from a college textbook. The task description explained the concept map assessment task and listed twenty-five node terms and nine linking phrases. It was developed by the researcher based on the content of the reading material. The evaluation was conducted in labs. Evaluation settings were Windows XP or Windows 7, Firefox 3.6, Adobe Flash player 10, and screen resolutions of 1440 by 900 or higher.

Data Collection

After finishing the system tryout process, each of the participants completed an evaluation questionnaire on his or her computer. The researcher developed this questionnaire based on the usability checklist for computer-based testing programs created by Harmes and Parshall (2000). The questionnaire included 29 items, which were divided into four sections: participant background, visual clarity, system functionality, consistency, and error prevention and correction (see Appendix G).

Procedure

First, participants were provided with printed and electronic version of materials used in the evaluation, including student directions for evaluating the assessment agent system, task description, and reading material. Second, participants were introduced to the research project, and were told that they were invited to evaluate the Assessment Agent System from the perspective of students. Third, the researcher explained the evaluation procedure and the basic knowledge of concept maps and demonstrated how to use the system. Fourth, participants began to read the provided materials and carried out a concept map assessment task by following the directions in the materials. The major activities included logging into the system, selecting course and task, reading the task description and reading material, constructing concept maps, submitting concept maps, and obtaining assessment reports. Fifth, after participants obtained their assessment reports, the researcher explained the content in the assessment reports. Sixth, participants completed the evaluation questionnaire.

Each session, including briefing, training, carrying out the concept map assessment task, and completing the system evaluation questionnaire, lasted between 50 minutes and 1 hour and 20 minutes.

Results of the Group Evaluation

Visual clarity. Most participants rated “Always” or “Most of the time” for the items related to visual clarity (see Appendix H). One participant commented that the user interface was “very user friendly”. Another participant stated, “Very comfortable handling of the colors, using warm and soft colors makes the interface pleasing to the eyes and easy to work with for extended periods of time.”

Suggestions for improving the user interface focused on the concept map creation space and color options. The concept map assessment task contained 25 nodes. As more and more propositions were created, the area for building the concept map on the screen became more and more crowded. Participants suggested enlarging the concept map working space or making the size of nodes smaller.

Another possibility for improving the user interface suggested by some participants is providing ways for changing node shape and changing colors of nodes, linking phrases and linking lines. One participant stated, “I wish there were different color options ("a coloring tool") so I could color nodes, arrows or linking phrases that were related or that were at the same organizational level in the concept map.”

System functionality. Most participants rated “Always” or “Most of the time” for the items related to button functioning and concept map creation (see Appendix H). All buttons worked well, and operations for constructing concept maps were intuitive and easy. One participant wrote, “The program was very easy to use once I became familiar with all the operations. Dragging nodes, creating links, adding link phrases were all very easy to do.”

Participants encountered a problem when they deleted links. Sometimes, the system did not respond to participants' clicking the linking lines for deleting links. Another issue that concerned most participants was the content of their assessment report, especially those propositions that were assessed as "different proposition." They thought that some of those "different" propositions were correct.

Participants made some suggestions for improving the system functionality. These suggestions include (a) providing synonyms of linking phrases, (b) saving images of student concept maps and allowing students to retrieve their concept maps graphically after submitting them, (c) graphically displaying correct and incorrect propositions on the student concept maps, (d) increasing the concept map creating space and providing zoom in and zoom out function that allow students to view their entire concept maps, (e) providing more personalized comments on student concept maps, and (f) providing online instruction for using the system.

Consistency. All participants rated "Always" or "Most of the time" for the items related to consistency, including icons, color arrangement, placement of information, and system responds to user actions (see Appendix H). One participant commented, "I think this is the strongest suit of the assessment tool. The colors and the node symbols were all consistent in arrangement and placement. I can't think of much that could help improve the consistency."

Error prevention and correction. Most participants rated "Always" or "Most of the time" for the items related to error prevention and correction, including mistake correction, viewing the entire concept map on the screen, and checking their concept map before submitting (see Appendix H). Their concerns focused on the small space for

creating concept maps. In such an environment, participants complained that nodes and links overlapped each other when more and more propositions were created. One participant suggested that when a student submitted a concept map, a warning box be displayed to ask if the student was sure he or she was ready to submit the concept map.

Appendix D: Student Directions for Evaluating the Assessment Agent System

Welcome to the evaluation of the Assessment Agent System. The evaluation process includes 14 steps. The entire process can be completed in approximately 1 hour and 30 minutes.

Step 1: Electronically sign the consent form on your computer

To view and sign the consent form (see Figure D1), please copy and paste the following link into your web browser. The password is **xxxxxxx**.

<https://survey.vt.edu/survey/entry.jsp?id=1270674285666>

If you have any questions or concerns, please contact the researcher Jianhua Liu, jhliu@vt.edu.

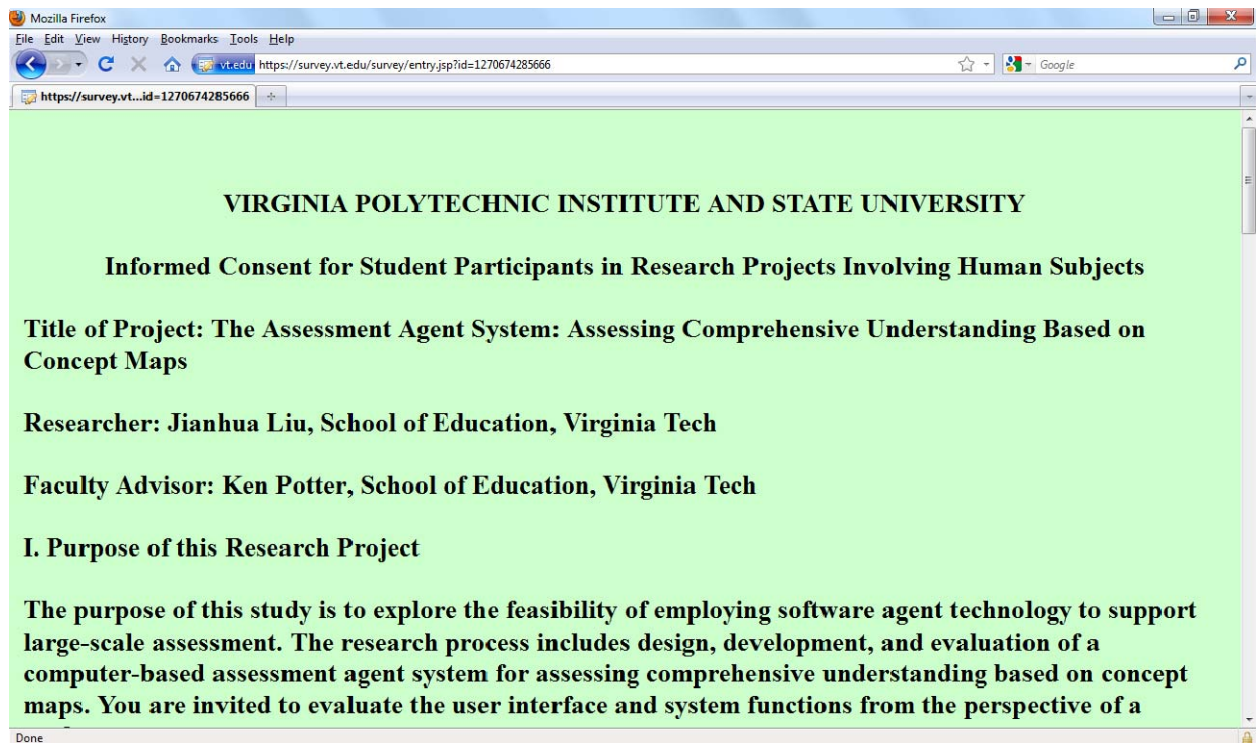


Figure D1. Student consent form.

Step 2: Read basic knowledge of concept maps

Definition of Concept Maps

A concept map is a diagram in which nodes represent concepts, and labeled directional links connecting nodes represent relationships between concepts.

An Example of Concept Map

Figure D2 is an example of concept map.

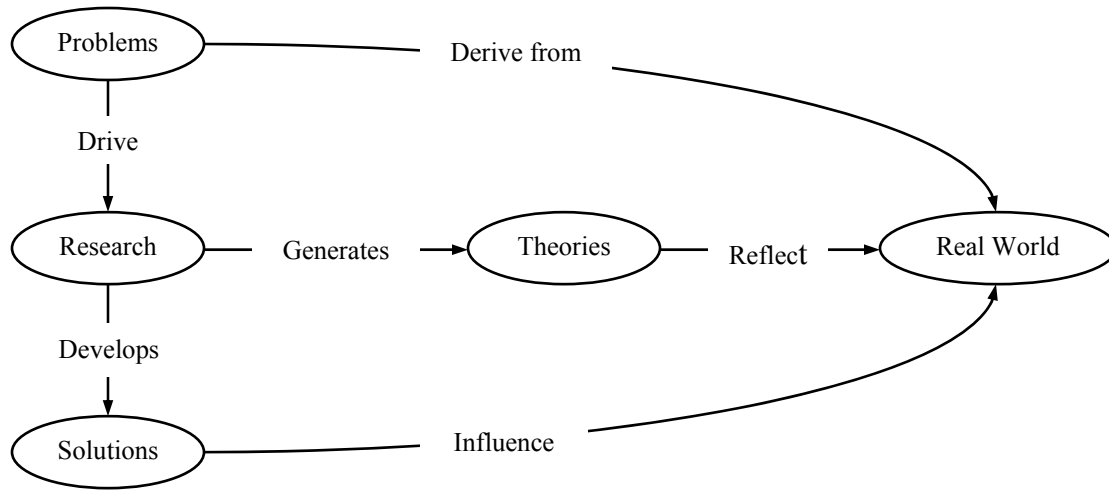


Figure D2. An example of concept map.

Components of Concept Maps

The basic components in a concept map are nodes and links.

A node is composed of a node symbol and a node term. A node symbol is a graphic shape, such as a circle, oval, or rectangle. A node term labels a node. The combination of a node symbol and a node term represents a concept.

In Figure D2, the node “Research” is composed of an oval and the term “Research”. The node “Research” represents the concept of research.

A link includes a linking line and a linking phrase. A linking line connects two nodes to indicate that there is a relationship between the two concepts. The arrowhead of the linking line indicates the direction of the relationship between the two concepts. A linking phrase specifies the relationship.

In Figure D2, “Generates” is a linking phrase, which is associated with a linking line to represent the relationship between the concepts “Research” and “Theories”.

A node-link-node combination is called a proposition, which is the basic unit of a meaningful statement in a concept map. In Figure D2, “Research - Generates - Theories” is a proposition.

Step 3: Read Task Description

Please read the provided Task Description material.

Step 4: Read the material “The Concept of Health”

You will be asked to create a concept map based on the content of the material.

Step 5: Log into the system

Please visit <http://www.citsie.net/jianhua/student/login.cfm> and log into the system with the provided user name and password. See Figure D3.

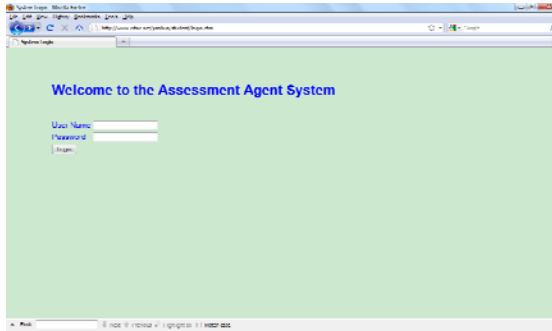


Figure D3. Login web page.

Step 6: Select “Take Assessment”

After logging in, the student menu is displayed. Select “Take Assessment”. Click the Submit button. See Figure D4.

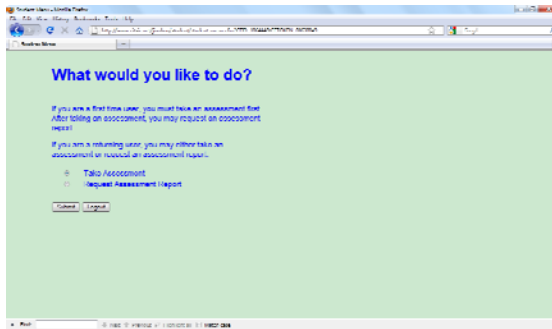


Figure D4. Student menu web page.

Step 7: Select course and task

Select the course “Health Education” and the task “Concept of Health”. Click the Submit button. See Figure D5.

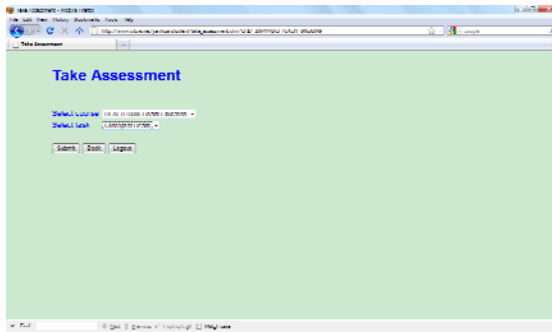


Figure D5. Select the course “Health Education” and the task “Concept of Health”.

Step 8: Learn the concept map creating environment

On the concept map creating environment web page (See Figure D6), nodes are displayed along the edges. Nodes are oval shaped. The linking phrases are displayed along the right edge. Linking phrases are rectangle shaped. On the bottom of the web page, there are four buttons: View Task Description, Submit Concept Map, Return to Student Menu, and Logout. Please use a high screen resolution so that the concept map creating environment can be appropriately displayed on your computer.

If your computer cannot display the concept map creating environment web page, a Flash player needs to be installed on your computer. If your computer does not have a Flash player, please visit the following link and download a free Adobe Flash Player.
<http://get.adobe.com/flashplayer/>

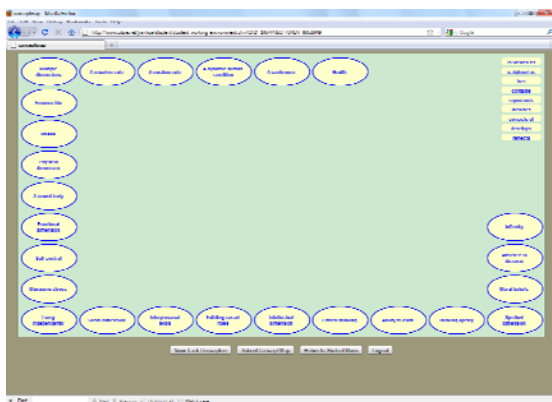


Figure D6. Concept map creating environment web page.

Step 9: Learn the operation for constructing a concept map

The construction of a concept map with the provided nodes and linking phrases involves moving nodes, connecting nodes, dragging and dropping linking phrases, and deleting linking lines or linking phrases.

- **Moving nodes:** Move your mouse over the node that you want to move, hold the mouse button down, and then drag the node into the place where you want on the screen.
- **Connecting nodes:** Connect two nodes by drawing a linking line. Move your mouse over the starting node, then while holding the **shift** key and clicking your left mouse button down, drag your mouse from the starting node to the terminal node, and release your mouse button. A line between the two nodes and an empty relationship box should appear.
- **Dragging and dropping linking phrases into relationship boxes.** Add a linking phrase to a relationship box by clicking your mouse on the linking phrase and dragging it into the empty relationship box. It should appear in the box when you release your mouse.
- **Changing linking phrases.** Replace a linking phrase that is already in a relationship box with a new one by clicking your mouse on the new linking phrase and dragging it into the relationship box. It should appear in the box when you release your mouse. The old linking phrase is replaced by the new linking phrase.
- **Deleting a linking line or linking phrase.** Click the linking line, a dialog box appears. If you click “Yes”, the linking line and linking phrase are deleted from your concept map. See Figure D7.

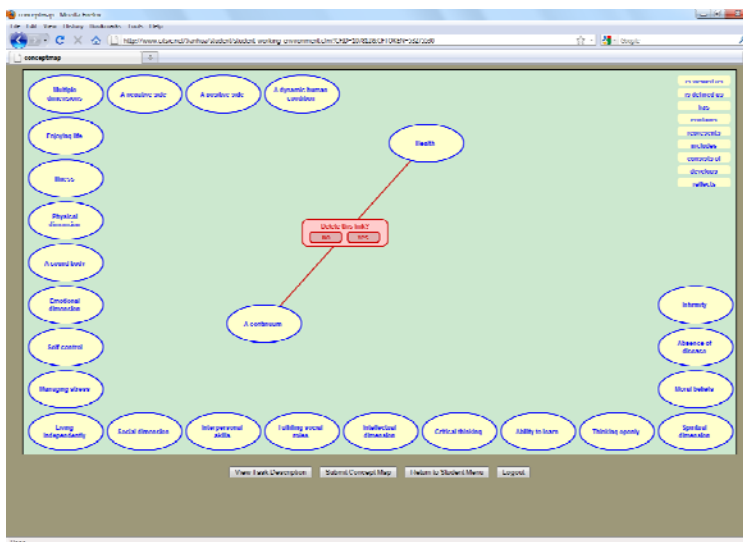


Figure D7. Deleting a linking line, a linking phrase or a link.

Step 10: Read the online task description

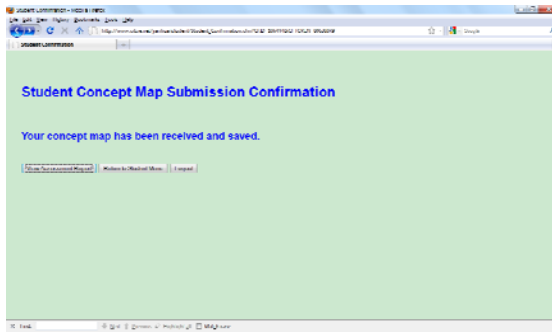


Figure D10. After clicking the “Submit Concept Map” button and the system received your concept map, a student map submission confirmation is displayed.

Step 13: View the assessment report

Click the “Return to Student Menu” button on the Student Concept Map Submission Confirmation web page. After the Student Menu web page is displayed, select “Request Assessment Report” and click the Submit button. See Figure D11.

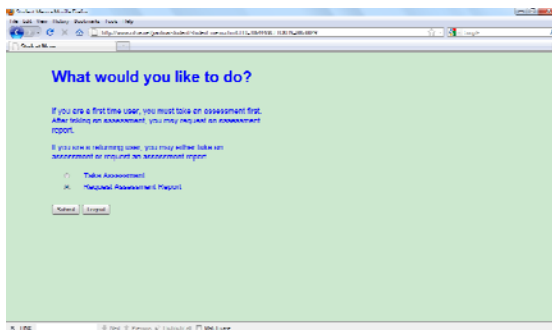


Figure D11. Select “Request Assessment Report”.

After the “Request Assessment Report” web page is displayed, select the course “Health Education” and the task “Concept of Health”. Click the Submit button. See Figure D12.

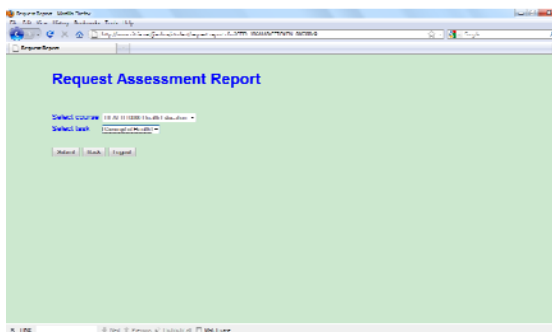


Figure D12. Select the course “Health Education” and the task “Concept of Health”.

The assessment report is displayed. See Figure D13.

Concept Map Assessment Report

Student Name	JANA PILOT
Student ID	10001
Course Title	Health Education
Task Title	Concept of Health
Number of Correct Propositions	10
Number of Followed Propositions	0
Number of Missing Propositions	10
Score of Accuracy	100%
Percent of Completion	100%

Assessment of propositions

Source node term	Linking phrase	Target node term	Assessment of proposition
HEALTH	is related to	A continuum	Correct
A continuum	has	A positive side	Correct
A continuum	has	A negative side	Correct
HEALTH	is defined as	A dynamic human condition	Correct
A positive side	includes	Adopting etc.	Missing
A negative side	includes	Illness	Missing
A dynamic human condition	includes	Multiple determinants	Missing
Multiple determinants	includes	Physical determinants	Missing
Physical determinants	includes	A social one	Missing
Multiple determinants	includes	Behavioral determinants	Missing
Behavioral determinants	includes	Self control	Missing
Behavioral determinants	includes	Managing stress	Missing
Behavioral determinants	includes	Living responsibly	Missing

Figure D13. Concept map assessment report.

Step 14: Complete a questionnaire

To view the questionnaire (see Figure D14), please copy and paste the following link into your web browser. The password is **xxxxxxx**.

<https://survey.vt.edu/survey/entry.jsp?id=1276181855531>

Please complete the questionnaire to present your evaluation, comments and suggestions about the system.

**System Evaluation Questionnaire
(Student Perspective)**

Section 1: Participant Background

1. User name

2. Do you have any experience taking computer-based assessments?
☐ Yes
☐ No

3. Have you ever evaluated computer-based educational products (e.g., educational software, websites)?
☐ Yes
☐ No

Section 2: Evaluate Visual Clarity

4. Was the information on the screen easy to read?
☐ ALWAYS

Figure D14. Evaluation questionnaire.

Thank you for participating in the evaluation of the assessment agent system.

Appendix E: Task Description

Course Title: Health Education

Task Title: Concept of Health

You are asked to create a concept map to present your understanding of the concept of health based on the reading material in step 4. You should create the concept map with the given node terms and linking phrases. These terms and phrases may include **inappropriate** ones. You need to identify which ones are appropriate and use them in your concept map.

Table E1

Node Terms and Linking Phrases

Node Terms	Linking Phrases
Health	is viewed as
A continuum	is defined as
A dynamic human condition	has
A positive side	includes
A negative side	represents
Multiple dimensions	contains
Enjoying life	consists of
Illness	develops
Physical dimension	reflects
A sound body	
Emotional dimension	
Self-control	
Managing stress	
Living independently	
Social dimension	
Interpersonal skills	
Fulfilling social roles	
Intellectual dimension	
Critical thinking	
Ability to learn	
Thinking openly	
Spiritual dimension	
Moral beliefs	
Absence of disease	
Infirmity	

Please continue to Step 4 in the Student Directions for Evaluating the Assessment Agent System.

Appendix F: Reading Material

The Concept of Health

Health can be viewed as a continuum with positive and negative poles. Positive health is associated with a capacity to enjoy life and to withstand challenges. Negative health is associated with illness and disease and, in the extreme, with premature death.

Health is a dynamic human condition with multiple but intertwined dimensions - physical, emotional, social, intellectual, and spiritual. A brief description of each dimension follows.

Physical Dimension of Health

The physical dimension of health encompasses the functional operation and soundness of the body. How well are you able to perform activities of daily living such as dressing, cooking, and walking? Can you perform routine tasks efficiently without becoming overly fatigued? Physical health, although the predominant dimension at times, is recognized as only one of five dimensions that underlie our health.

Emotional Dimension of Health

The emotional dimension is associated with the ability to deal with personal feelings in a positive and constructive manner, to cope with stress, and to live independently. A strong emotional dimension is a key to greater life satisfaction.

Social Dimension of Health

Closely tied to emotional health, the social dimension is the ability to interact effectively with other people, to develop satisfying interpersonal relationships, and to fulfill social roles. Hallmarks of social health are the capacity for intimacy, meaningful relationships with family and friends, and participation in community activities.

Intellectual Dimension of Health

The intellectual dimension is reflected in our ability to question and evaluate information, to think and learn from a variety of experiences, and to be open to new ideas. Critical thinking and problem solving are fundamental lifelong pursuits that enable us to continue learning new material and master new skills.

Spiritual Dimension of Health

The essential component of spiritual health is a commitment to a set of values or principles that guide our actions. This commitment is usually but not always based on divine beliefs. The basis for leading a moral and ethical life is rooted in spiritual health.

The dimensions of health do not exist as separate elements. Rather, each dimension relates to all others, and contributes to the overall balance, quality, and strength of the whole.

Adapted from Sparling, P. B., & Redican, K. J. (2009). Personal health: A concise guide. McGraw-Hill. pp. 3-8.

Please continue to Step 5 in the Student Directions for Evaluating the Assessment Agent System.

Appendix G: System Evaluation Questionnaire

Table G1

System Evaluation Questionnaire

Participant Background				
User name:				
Do you have any experience taking computer-based assessments?	Yes/No			
Have you ever evaluated computer-based educational products (e.g., educational software, websites)?	Yes/No			
Visual Clarity				
Was the information on the screen easy to read?	Always	Most of the time	Some of the time	Never
Did the color arrangement of the screen help make the displays clear?	Always	Most of the time	Some of the time	Never
Was it easy to find the needed information on the screen?	Always	Most of the time	Some of the time	Never
Were the different information settings clearly separated from one another?	Always	Most of the time	Some of the time	Never
Comments and suggestions on Visual Clarity:				

Table G1 (continued)

System Evaluation Questionnaire

System Functionality				
Did all the buttons and menus function correctly?	Always	Most of the time	Some of the time	Never
Were you able to arrange nodes to create your concept map?	Always	Most of the time	Some of the time	Never
Were you able to remove nodes from your concept map?	Always	Most of the time	Some of the time	Never
Were you able to create linking lines?	Always	Most of the time	Some of the time	Never
Were you able to delete linking lines?	Always	Most of the time	Some of the time	Never
Were you able to drag and drop linking phrases into relationship boxes?	Always	Most of the time	Some of the time	Never
Were you able to change linking phrases that were already in relationship boxes?	Always	Most of the time	Some of the time	Never
Were you able to delete linking phrases?	Always	Most of the time	Some of the time	Never
Were you able to delete links?	Always	Most of the time	Some of the time	Never
Were you able to rearrange and organize nodes in your concept map?	Always	Most of the time	Some of the time	Never
Did the assessment report provide beneficial information about your concept map?	Provided all information needed	Provided most information needed	Provided some information needed	Provided none of information needed
Comments and suggestions on Functionality:				

Table G1 (continued)

System Evaluation Questionnaire

Consistency				
Were icons used consistently in the student interface?	Always	Most of the time	Some of the time	Never
Was the color arrangement consistent in the student interface?	Always	Most of the time	Some of the time	Never
Did the same type of information (e.g. navigation) have consistent placement in the student interface?	Always	Most of the time	Some of the time	Never
Was the assessment agent system consistent in responding to a particular user action?	Always	Most of the time	Some of the time	Never
Comments and suggestions on Consistency:				
Error Prevention and Correction				
Was it easy for you to correct a mistake?	Always	Most of the time	Some of the time	Never
Could you view the entire concept map on the screen?	Always	Most of the time	Some of the time	Never
Were you able to check your concept map before it was submitted?	Always	Most of the time	Some of the time	Never
Comments and suggestions on Error Prevention and Correction:				

Appendix H: Participant Responses to Items in the Questionnaire

Table H1

Participant Responses to Items Related to Visual Clarity

Visual Clarity	Always	Most of the time	Some of the time	Never	Total
Was the information on the screen easy to read?	20	12	2	0	34
Did the color arrangement of the screen help make the displays clear?	26	6	2	0	34
Was it easy to find the needed information on the screen?	22	10	2	0	34
Were the different information settings clearly separated from one another?	21	10	3	0	34

Table H2

Participant Responses to Items Related to System Functionality

System Functionality	Always	Most of the time	Some of the time	Never	Total
Did all the buttons and menus function correctly?	26	8	0	0	34
Were you able to arrange nodes to create your concept map?	28	6	0	0	34
Were you able to remove nodes from your concept map?	28	4	1	1	34
Were you able to create linking lines?	31	3	0	0	34
Were you able to delete linking lines?	24	8	2	0	34
Were you able to drag and drop linking phrases into relationship boxes?	29	5	0	0	34
Were you able to change linking phrases that were already in relationship boxes?	32	1	0	1	34
Were you able to delete linking phrases?	27	3	1	3	34
Were you able to delete links?	27	6	1	0	34
Were you able to rearrange and organize nodes in your concept map?	30	4	0	0	34
Did the assessment report provide beneficial information about your concept map?	Provided all information needed 7	Provided most information needed 19	Provided some information needed 8	Provided none of information needed 0	34

Table H3

Participant Responses to Items Related to Consistency

Consistency	Always	Most of the time	Some of the time	Never	Total
Were icons used consistently in the student interface?	29	5	0	0	34
Was the color arrangement consistent in the student interface?	33	1	0	0	34
Did the same type of information (e.g. navigation) have consistent placement in the student interface?	31	3	0	0	34
Was the assessment agent system consistent in responding to a particular user action?	31	3	0	0	34

Table H4

Participant Responses to Items Related to Error Prevention and Correction

Error Prevention and Correction	Always	Most of the time	Some of the time	Never	Total
Was it easy for you to correct a mistake?	20	12	2	0	34
Could you view the entire concept map on the screen?	23	6	5	0	34
Were you able to check your concept map before it was submitted?	29	4	0	1	34

Appendix I: Institutional Review Board Approval Memorandums



Office of Research Compliance
Institutional Review Board
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, Virginia 24061
540/231-4991 Fax 540/231-0959
e-mail moored@vt.edu
www.irb.vt.edu

FWA00000572 (expires 1/20/2010)
IRB # is IRB00000687

DATE: April 22, 2009

MEMORANDUM

TO: Kenneth R. Potter
Jianhua Liu

FROM: David M. Moore

Approval date: 4/22/2009
Continuing Review Due Date: 4/7/2010
Expiration Date: 4/21/2010

SUBJECT: **IRB Expedited Approval:** "The Development of an Assessment Agent System for Assessing Comprehensive Understanding Based on Concept Maps", IRB # 09-432

This memo is regarding the above-mentioned protocol. The proposed research is eligible for expedited review according to the specifications authorized by 45 CFR 46.110 and 21 CFR 56.110. As Chair of the Virginia Tech Institutional Review Board, I have granted approval to the study for a period of 12 months, effective April 22, 2009.

As an investigator of human subjects, your responsibilities include the following:

1. Report promptly proposed changes in previously approved human subject research activities to the IRB, including changes to your study forms, procedures and investigators, regardless of how minor. The proposed changes must not be initiated without IRB review and approval, except where necessary to eliminate apparent immediate hazards to the subjects.
2. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.
3. Report promptly to the IRB of the study's closing (i.e., data collecting and data analysis complete at Virginia Tech). If the study is to continue past the expiration date (listed above), investigators must submit a request for continuing review prior to the continuing review due date (listed above). It is the researcher's responsibility to obtain re-approval from the IRB before the study's expiration date.
4. If re-approval is not obtained (unless the study has been reported to the IRB as closed) prior to the expiration date, all activities involving human subjects and data analysis must cease immediately, except where necessary to eliminate apparent immediate hazards to the subjects.

Important:

If you are conducting **federally funded non-exempt research**, please send the applicable OSP/grant proposal to the IRB office, once available. OSP funds may not be released until the IRB has compared and found consistent the proposal and related IRB application.

cc: File

Invent the Future

VIRGINIA POLYTECHNIC INSTITUTE UNIVERSITY AND STATE UNIVERSITY
An equal opportunity, affirmative action institution



MEMORANDUM

DATE: April 1, 2010

TO: Kenneth R. Potter, Jianhua Liu

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires June 13, 2011)

PROTOCOL TITLE: The Assessment Agent System: Assessing Comprehensive Understanding Based on Concept Maps

IRB NUMBER: 09-432

As of April 22, 2010, the Virginia Tech IRB Chair, Dr. David M. Moore, approved the continuation request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at <http://www.irb.vt.edu/pages/responsibilities.htm> (please review before the commencement of your research).

PROTOCOL INFORMATION:

Approved as: **Expedited, under 45 CFR 46.110 category(ies) 7**

Protocol Approval Date: **4/22/2010**

Protocol Expiration Date: **4/21/2011**

Continuing Review Due Date*: **4/7/2011**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals / work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Invent the Future

Date*	OSP Number	Sponsor	Grant Comparison Conducted?

*Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

cc: File

Appendix J: Informed Consent Form

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Informed Consent for Participants in Research Projects Involving Human Subjects

Title of Project: The Assessment Agent System: Assessing Comprehensive Understanding Based on Concept Maps

Researcher: Jianhua Liu, School of Education, Virginia Tech

Faculty Advisor: Ken Potter, School of Education, Virginia Tech

I. Purpose of this Research Project

The purpose of this study is to explore the feasibility of employing software agent technology to support large-scale assessment. The research process includes design, development, and evaluation of a computer-based assessment agent system for assessing comprehensive understanding based on concept maps. You are invited to evaluate the user interface and system functions from the perspective of a student.

II. Procedures

The evaluation process includes three sessions: (1) training, (2) trying out the system, and (3) completing a questionnaire. In the first session, you will learn the basic knowledge of concept maps. In the second session, you will read material and use the system to create a concept map based on the content of the material. In the third session, you will complete a questionnaire to present your evaluation, comments, and suggestions about the system.

The entire process could be done in approximately 1 hour and 30 minutes. The evaluation will be conducted either in a lab or online.

III. Risks

The risks associated with participation in this study are minimal. The researchers will work to ensure all responses collected through this study are stored securely and remain confidential.

IV. Benefits

Your evaluation provides information that will be used to improve the assessment agent system. No guarantee of benefits has been made to encourage you to participate in this study.

V. Extent of Anonymity and Confidentiality

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a pseudonym will identify you during analyses and any written reports of the research. It is possible that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. Data will be kept for approximately one year after the conclusion of the study.

VI. Compensation

Your participation is voluntary and unpaid.

VII. Freedom to Withdraw

You may withdraw from the research project at any time and for any reason. You are free to not answer any questions without penalty. To withdraw, please inform the researchers listed at the bottom of this form.

VIII. Subject's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

I will follow the provided "Student Directions for Evaluating the Assessment Agent System" to participate in the study.

IX. Subject's Permission

I have read the Consent Form and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent.

Should I have any pertinent questions about this research or its conduct, and research subjects' rights, and whom to contact in the event of a research-related injury to the subject, I may contact:

Jianhua Liu
Researcher
Telephone: 540-xxx-xxxx
E-mail: jhliu@vt.edu

Ken Potter
Faculty Advisor
Telephone: 540-231-9694
E-mail: kpotter@vt.edu

David M. Moore
Chair
Virginia Tech Institutional Review Board for the Protection of Human Subjects
Office of Research Compliance
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, VA 24060
Telephone: 540-231-4991
E-mail: moored@vt.edu