Integrating Concept Mapping Techniques into Object-Oriented Analysis and Design Curriculum

Ven Yu Sien, HELP University College, Malaysia

Abstract: Information technology (IT) students at higher educational institutions have in general found difficulty in grasping object-oriented (OO) concepts. They particularly find difficulty in performing abstractions of real-world problems within the context of object-oriented analysis and design (OOAD). They are unable to effectively build models from the problem domain because they essentially do not know 'what' to model. Concept mapping is a popular tool used in education for facilitating learning, comprehension and the development of knowledge. Within the context of OOAD, we successfully used concept maps to help students understand and master the technique of abstraction in order that they can improve their OO modelling skills and produce more appropriate OOAD models. We present a set of guidelines for teaching OO modelling with concept maps. These guidelines (consisting of four teaching modules) could be integrated into existing OOAD courses at undergraduate or postgraduate level, and OOAD workshops to help software engineering educators resolve some of the difficulties they face when teaching OOAD. This paper contains results of our evaluation of the effectiveness of integrating the concept mapping techniques in an introductory OOAD course.

Keywords: Object-Oriented Analysis and Design, Software Engineering Education, Concept Maps, Object-Oriented Modelling, Teaching Guidelines

Introduction

MOST EDUCATION INSTITUTIONS teach object-oriented analysis and design (OOAD) courses in the context of a software development life cycle e.g., iterative and incremental, Rational Unified Process or Extreme Programming. While development processes differ, they all share some common phases i.e., the requirements specification, analysis, design and implementation phases. Each phase produces deliverables that are required by other phases in the life cycle. A deliverable may be a document, a software artefact or a system test plan [1].

A software system is usually complex. It is therefore essential to break the software system into manageable components in order to fully understand and manage its complexity. The overall system and its components may be represented as models – abstractions that illustrate important characteristics from various perspectives. During the OOAD phases, models are produced to show the type of information processing that is required of the new system. Modelling a system, however, requires the representation of different perspectives or views of the system and therefore there are different types of diagrams for modelling each of these views. Models used for OO development can take the form of graphics, narratives, or formulae. It is however difficult to abstract concepts from real-world problems [2-4].

Despite some of the claims made by advocates of the OO approach [5, 6], novices in OO techniques in general have difficulties understanding OO concepts. OOAD is not easy to learn, and a particular challenge for students is developing the ability to abstract real-world

The International Journal of Learning
Volume 17, Number 2, 2010, http://www.Learning-Journal.com, ISSN 1447-9494

problems. Students seem unable to effectively build OO models from the problem domain because they essentially do not know '*what*' to model.

The Concept Mapping Approach

Students at higher education institutions have in general experienced considerable difficulties understanding OO concepts and acquiring the necessary skills in OOAD [7-9]. In order to resolve some of these issues, we presented a concept mapping approach [10, 11] to introduce concept mapping as a tool to help students with OO modelling. Concept mapping was developed by Joseph Novak [12] in 1972 at Cornell University and is commonly used for visualising relationships between concepts. Concept maps are two-dimensional, hierarchical diagrams that represent the structure of knowledge within a particular domain as nodes (or concepts) and connecting links. A concept is '*an idea or notion that we apply to the things, or objects, in our awareness*' [6]. Concepts are related to each other by a link, and each link has a word or word-phrase describing the relationship between the concepts. Concept mapping is a technique for representing the structure of information visually. A simple example of a concept map is illustrated in Fig. 1.

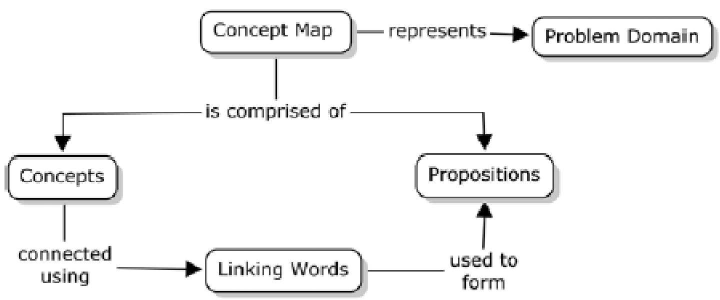

Figure 1: Concept Map Showing the Key Features of Concept Maps – Adapted From [13]

Within the context of OO modelling, we used concept maps as an initial abstraction of the problem domain. The structural and behavioural aspects of the problem domain are represented respectively by static and dynamic concept maps. In order to effectively produce the static and dynamic concept maps, a set of processes are provided to assist students abstract the necessary concepts from use cases (detailed description of processes). Students can apply transformation rules to convert a static concept map to an analysis class diagram, and a dynamic concept map to a sequence diagram. Details of the processes and transformation rules are reported in [10, 11]. The Unified Modelling Language (UML) class and sequence diagrams are selected for this exercise because they represent the essential static and behavioural aspects of a problem domain. The class diagram is fundamental to the OO modelling process and provides a static description of system components and structures in terms of classes and their relationships. The sequence diagram is selected because it is identified as the '*major UML diagram that captures the detailed behaviour of objects in the system'* [14] and it is one of the most widely used dynamic diagrams in UML [15].

Fig. 2 illustrates an example of how a static concept map is transformed to a class diagram.

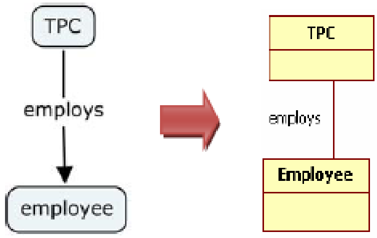

Figure 2: Static Concept Map Converted to a Class Diagram

Similarly, Fig. 3 illustrates an example of how a dynamic concept map is transformed to a sequence diagram.

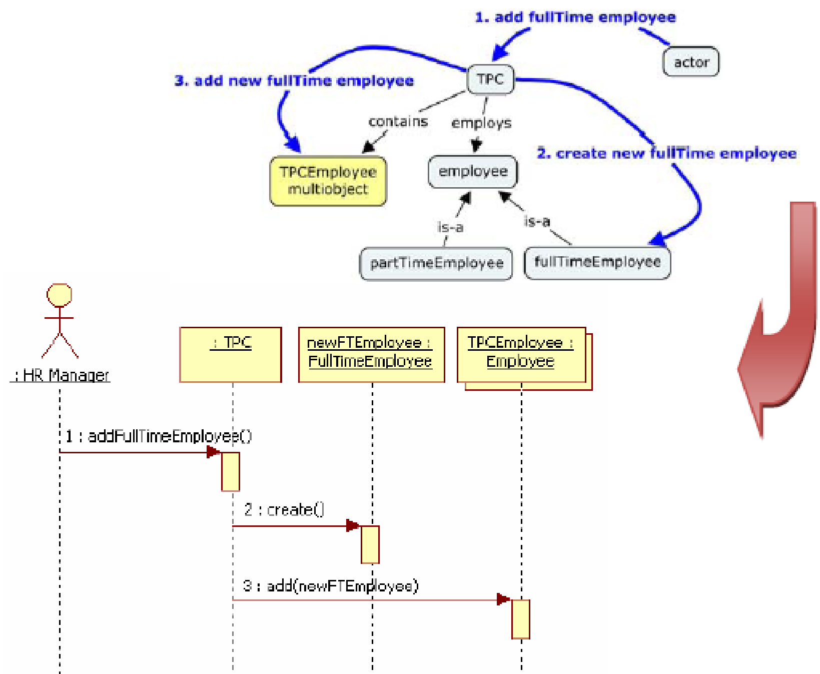

Figure 3: Dynamic Concept Map Converted to a Sequence Diagram

We conducted four experiments to validate the effectiveness of the proposed concept mapping approach using a pre-test and post-test experimental design. The aim of these experiments was to determine if participants who have been taught the concept mapping technique and used it as a basis for developing UML diagrams produced more appropriate UML diagrams. *More appropriate* in the context of these experiments was determined by a comparison of class and sequence diagram scores produced in the pre-test and post-test sessions. All the participants were expected to attend a pre-test session, a workshop on concept mapping techniques and a post-test session. In the pre-test, each participant was given a case study and he/she was required to produce a class and sequence diagram from the case study. After the pre-test, the participants were taught concept mapping techniques (how to produce concept maps and transform them to class and sequence diagrams) at the workshop. The workshop was followed by the post-test where each participant was given another case study (different to the one that he/she had worked on previously) and required to construct static and dynamic concept maps and derive a class and sequence diagram from the respective concept maps.

We used one-way paired t-tests and a significance level of 0.05 to compare the means of the pre-test and post-test diagram scores. When the data from all four experiments was combined, we concluded that the use of concept maps resulted in a significant improvement in class diagrams ($t_{89} = -8.2$, p-value = 0.001) and sequence diagrams ($t_{84} = -10.3$, p-value = 0.001).

Design of Teaching Modules for the Concept Mapping Approach

A set of four teaching modules for the concept mapping approach was used in the workshops on concept-mapping techniques. The teaching modules have the following advantages:

- Each module focuses on one specific aspect of the concept mapping techniques and has clearly defined learning objectives. Each module provides a context for understanding and applying specific concepts. The modules are designed to be taught in sequence as each module depends on the knowledge gained from the preceding modules.
- By breaking the learning content into small modules, students are not overwhelmed by the scope and complexity of the subject matter.
- The modules can be integrated into other courses.
- The modules can be reused, and techniques in each module can be independently enhanced.

An overview of the modules follows:

- Module 1: Static Concept Map
  This module introduces concept maps and how they can be used to represent the problem domain:
  - What is a concept map?
  - Identifying concepts and their relationships from expanded use cases. These processes are reported in [10].
  - Hands-on exercises. Students are expected to work on a case study to produce a static concept map representing the problem domain.

- Module 2: Static Concept Map → Analysis Class Diagram
  This module covers the following topics:
  - Transforming a static concept map to a class diagram. These rules are reported in [10].
  - Hands-on exercises to produce a class diagram. Students continue with the case study that they worked on in Module 1.

- Module 3: Dynamic Concept Maps
  This module presents the concepts and techniques necessary to produce dynamic concept maps from expanded use cases. The following topics are covered:
  - Identifying concepts to represent multi-objects in a dynamic concept map.

- Identifying responsibilities from an expanded use case.
- Assigning responsibilities to concepts using appropriate design patterns.
- Producing a dynamic concept map to 'realise' a use case. These processes are reported in [11].
- Hands-on exercises to produce dynamic concept maps. Students continue with the case study that they worked on in Module 1.

- Module 4: Dynamic Concept Map → Sequence Diagram
  This module covers the following topics:
  - Converting a dynamic concept map to a sequence diagram. These rules are reported in [11].
  - Hands-on exercises to produce sequence diagrams. Students continue with the case study that they worked on in Module 1.

At the end of each module, the students are expected to work on hands-on exercises. The exercises are designed to help students apply theory to practice, and subsequently assist them to improve their abstraction skills. The students are expected to draw the models using pen/pencil and paper. However, the lecturer of the course or facilitator of the workshop may prefer that the students use software applications to draw the models.

During the workshops that were conducted as part of the experiments, we received feedback from lecturers, participants, detailed questionnaires completed by participants and open discussions. We learned that participants in general found the workshops useful and liked the idea of using concept maps before producing class and sequence diagrams. They found the concept maps helpful for identifying appropriate components for the class and sequence diagrams.

Consequently, based on experience gained from conducting the workshops and feedback received from the participants, we made some changes to the teaching modules to make them more effective:

- Increase the time duration for each module.
- Modify the guided practice sessions.
  - The facilitator of the workshop will select and discuss some of the concept maps and UML diagrams produced by the participants. This will help explain commonly committed errors and conceptual misunderstandings.
  - There are three guided practice sessions in each module. Within each module, the facilitator will demonstrate a sample solution (e.g., on a whiteboard) during the first guided practice session. The second and third guided practice sessions will be attempted by the participants.
  - Sample solutions will be provided to the students after each practice session. The students can work from these sample solutions for the subsequent exercise.
  - Students are encouraged to work in pairs.

Integrating the Concept Mapping Approach into an Existing OOAD Course

In a typical introductory OOAD course, many OO concepts and definitions (e.g., objects, classes, inheritance, encapsulation, polymorphism, UML notation) are presented before the various types of UML analysis and design models are taught. Our proposed framework consists of the following sequence of topics:

1. Requirements analysis and use cases
2. Module 1: Static Concept Map
3. Analysis class diagram (e.g., classes, attributes, associations, generalisation-specialisation hierarchy, whole-part association, etc.) and UML notation for the OO approach
4. Remaining topics in object-oriented analysis (OOA) (e.g., system sequence diagrams, contracts, etc.)
5. Module 2: Static Concept Map → Analysis Class diagram
6. Module 3: Dynamic Concept Map
7. Module 4: Dynamic Concept Map → Sequence Diagram
8. Remaining topics in object-oriented design (OOD) (e.g., design class diagram, etc.)
9. Remaining topics in OOAD (e.g., object-oriented programming language code examples showing domain classes and relationships, OO development, etc.)

This process is illustrated in Fig. 4.

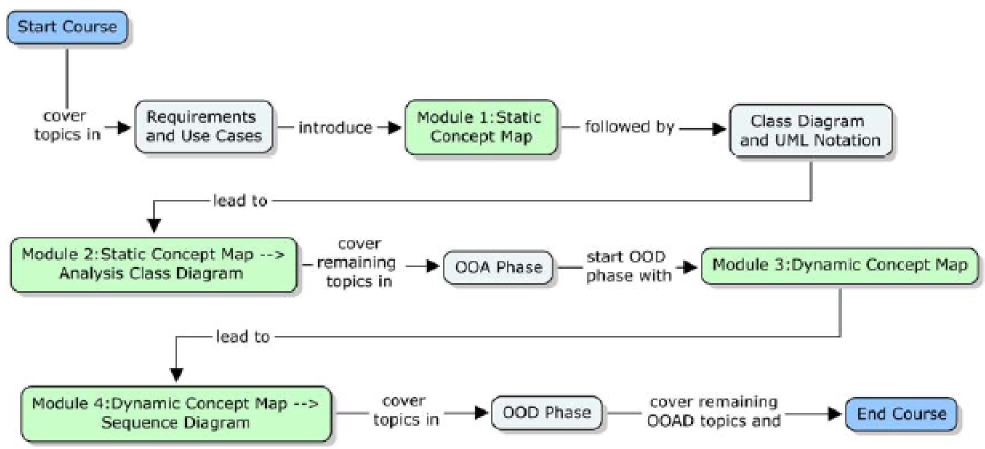

Figure 4: Process for Integrating the Concept-Driven Approach into Existing OOAD Curriculum

Since the evaluation results from the experiments conducted on the effectiveness of using concept maps as a stepping stone to produce class and sequence diagrams provided strong evidence that the concept mapping approach can be used successfully, we implemented the teaching modules in an introductory OOAD course in the February 2009 semester at HELP University College, Malaysia.

The original structure of the OOAD course (before inclusion of the concept mapping modules) is shown in Table 1.

Table 1: Topics in an OOAD Course

Object-Oriented Analysis and Design What is analysis and design? The Unified Modelling Language
Introduction to a Development Process The UML and development processes Iterative development
Requirements and Use Cases Use case diagram and writing use cases Ranking use cases
Developing the Conceptual Model Part 1 Introducing basic OO concepts: object, class, attributes, and operations The Unified Modelling Language What is a domain model? How to create a domain model? Associations and inheritance relationships
Developing the Conceptual Model Part 2 Associative types, association role names Identifying and applying aggregation
System Operations and Contracts Identifying system operations from use cases Specifying system behaviour from system operations
UML Interaction Diagrams Common UML interaction diagram notation Basic sequence diagram notation
GRASP: Designing Objects with Responsibilities GRASP: A methodical approach to basic OO design What's the connection between responsibilities, GRASP, and UML diagrams? What are patterns?
Designing for Visibility What is visibility?
Design Class Diagrams How to make a design class diagram Conceptual model vs design class diagrams
Mapping Designs to Code Programming and iterative, evolutionary development Mapping designs to code

The revised structure of the course (Table 2) shows how the modules were incorporated.

Table 2: Amended Topics in an OOAD Course

Object-Oriented Analysis and Design
Introduction to a Development Process
Requirements and Use Cases
Module 1: Static Concept Maps Using concept maps as an initial abstraction of the problem domain
Developing the Conceptual Model Part 1 **Module 2: Static Concept Map → Analysis Class Diagram** Creating an analysis class diagram from a static concept map
Developing the Conceptual Model Part 2
System Operations and Contracts
Module 3: Dynamic Concept Maps Identifying responsibilities from an expanded use case
Module 4: Dynamic Concept Map → Sequence Diagram Converting a dynamic concept map to a sequence diagram
Designing for Visibility
Design Class Diagrams
Mapping Designs to Code

Evaluating the Integration of the Concept-Driven Approach

To investigate the effectiveness of integrating the concept mapping techniques in an introductory OOAD course, we compared the results of two studies. The participants in the first study, Study 1, were not exposed to concept mapping techniques while the participants in the second study, Study 2, were taught concept mapping techniques as part of their OOAD course. While we are aware that we are comparing results from two different sets of students, we can nonetheless use the outcome as a basis for considering the implications of incorporating concept mapping techniques in an OOAD course.

Problem Statement

It has been observed that students frequently produce class diagrams that are incomplete, with many concepts at inconsistent abstraction levels; and sequence diagrams with missing responsibilities and objects [7, 16]. Consequently, we developed some concept mapping techniques to help novices in OOAD produce more appropriate class and sequence diagrams. When these techniques are implemented in introductory OOAD courses, we expect students in these courses to produce more appropriate UML diagrams compared to students who have not been exposed to the concept mapping techniques.

Research Objectives

This study aims to explore the effectiveness of integrating concept mapping techniques in an OOAD course.

The research question for this study is:

Are there any improvements in the class and sequence diagrams produced by students as a consequence of using our concept mapping techniques?

We use the Goal/Question/Metric (GQM) template [17, 18] to help us define the goal of this study.

Table 3: Goal According to GQM

Analyse	the integration of concept mapping techniques in an OOAD course
For the purpose of	minimising the types of errors in class and sequence diagrams, and evaluating whether the concept mapping techniques improve the appropriateness of class and sequence diagrams produced by OOAD students
With respect to	the students' OO modelling skills
From the point of view of	software engineering educators
In the context of	teaching students how to develop UML class and sequence diagrams during the analysis and design phases of a software development project

Subjects

Study 1 consisted of fifty-one Year 2 IT undergraduate students and Study 2 consisted of twenty-one Year 2 IT undergraduate students. All the participants were volunteers and were not paid to take part in the study. The students in Study 1 completed an OOAD course (based on topics listed in Table 1) and were not exposed to any concept mapping techniques whilst the students in Study 2 were taught the concept mapping techniques as part of their OOAD course (based on topics listed in Table 2). Information captured on the background of the participants and their experience with OOAD is summarised in Table 4. The OO experience reported for these participants is based on the experience they gained from their university courses that include OO concepts.

Table 4: Background Information on Participants from Studies 1 and 2

	Study 1 (N=51)	**Study 2 (N=21)**
Average age (years)	22	22
Age range (years)	20-27	19-28

Gender • male • female	 61% 39%	 90% 10%
OO experience • < 1 year • 1- 2 years • 3-5 years • > 5 years	 35% 49% 16% 0%	 48% 52% 0% 0%

Study Material

The participants from both studies were asked to work on a case study containing four expanded use cases that describe the functional requirements of the system.

We used rubrics to define the assessment criteria for evaluating the appropriateness of the participants' diagrams.

Hypothesis

The following hypotheses were formulated

H_1: *More appropriate class diagrams are produced as a consequence of integrating concept mapping techniques into the OOAD course.*

H_2: *More appropriate sequence diagrams are produced as a consequence of integrating concept mapping techniques into the OOAD course.*

Procedure

The participants in Study 1 were given a case study to work on after they had completed the topic on GRASP patterns (see Table 1). In Study 2, Modules 1, 2, 3 and 4 were integrated into the existing OOAD topics (see Table 2). After completing Module 4, the participants in Study 2 were given the same case study to work on.

The participants in Study 1 were given 1 hour to work on the case study as they were required to produce a class and sequence diagram. The participants in Study 2, however were given 1 hour 30 minutes to work on the case study as they were required to produce a static concept map, a class diagram, a dynamic concept map and a sequence diagram.

Descriptive Statistics

Table 5 shows a significant difference in the composite scores for Studies 1 and 2 for both class and sequence diagrams. The class and sequence diagram scores from Study 2 are significantly higher than the corresponding scores from Study 1 – suggesting that concept maps have helped participants develop more appropriate class diagrams.

Table 5: Descriptive Statistics for the Composite Scores

Study	Diagram	N	Mean	Median	Std Dev	Minimum Scores	Maximum Scores
1	Class	51	50.4	53	16.8	8	87
2	Class	21	76.2	80	14.5	46	94
1	Sequence[1]	31	17.7	20	13.3	0	43
2	Sequence[2]	17	59.2	64	26.5	20	100

[1] Twenty students in Study 1 did not produce any sequence diagrams – possible explanations for this may be insufficient time or lack of knowledge in producing sequence diagrams.
[2] Four students in Study 2 did not produce any sequence diagrams – possible explanations for this may be insufficient time or lack of knowledge in producing sequence diagrams.

Hypothesis Testing

In what follows, we will approximate the (population) average scores for the Study 1 participants by the sample means. That is, we will assume that the average scores for class and sequence diagrams for the population of people who have not learnt about concept mapping are 50.4 and 17.7, respectively.

If we let μ_C and μ_S denote the true (population) mean scores for Study 2 for class and sequence diagrams respectively, we can express the null and alternative hypotheses as follows.

$H_{10}: \mu_C = 50.4$ *(There is no improvement in class diagrams as a consequence of using the concept mapping techniques to identify initial concepts and associations from the problem domain.)*

$H_{1A}: \mu_C > 50.4$ *(There is an improvement in class diagrams as a consequence of using the concept mapping techniques to identify initial concepts and associations from the problem domain.)*

$H_{20}: \mu_S = 17.7$ *(There is no improvement in sequence diagrams as a consequence of using the concept mapping techniques to identify appropriate concepts, responsibilities and collaborations to support a use case.)*

$H_{2A}: \mu_S > 17.7$ *(There is an improvement in sequence diagrams as a consequence of using the concept mapping techniques to identify appropriate concepts, responsibilities and collaborations to support a use case.)*

We used a one sample t-test and a significance level of 0.05 to decide between H_{10} and H_{1A}. From the summary of results reported in Table 6, we can conclude that the use of concept maps resulted in a significant improvement in class diagrams ($t_{20} = 8.17$, *p*-value = 0.001). That is, we reject H_{10}. We are 95% confident that the average score for participants using concept maps is between 69.6 and 82.8. This range does not include 50.4, the average score achieved by the group which did not use concept maps. Thus, we conclude that concept mapping significantly improves results. We therefore believe that there is an improvement in class diagrams from using the concept mapping techniques to identify initial concepts and associations from the problem domain.

Similarly, we used a one sample t-test and a significance level of 0.05 to decide between H_{20} and H_{2A}. From the summary of results reported in Table 6, we can conclude that the use of concept maps resulted in a significant improvement in sequence diagrams ($t_{16} = 6.46$, p-value = 0.001). That is, we reject H_{20}. We are 95% confident that the average score for participants using concept maps is between 45.6 and 72.9. This range does not include 17.7, the sample average score achieved by the group which did not use concept maps. Thus we conclude that concept mapping significantly improves the sequence diagram results. We therefore believe that there is an improvement in sequence diagrams from using the concept mapping techniques to identify appropriate concepts, responsibilities and collaborations to support a use case.

Table 6: One-sample t-test

	Null Hypothesis	Mean	Std. Dev	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig (1- tailed test)
					Lower	Upper			
Study 2: Class Diagram	$\mu_C = 50.4$	76.2	14.5	3.16	69.6	82.8	8.17	20	0.001
Study 2: Sequence Diagram	$\mu_S = 17.7$	59.2	26.5	6.43	45.6	72.9	6.46	16	0.001

Discussion of Results

It is evident from the results that the quality of class and sequence diagrams is better in Study 2. At the end of Study 2, we conducted an open discussion with the students and they responded positively to the use of concept maps. This is consistent with previous research [19-22] where concept maps were successfully used as strategies to teach students.

Threats to Validity

This section discusses some threats to validity that may affect this study.

Internal Validity. Internal validity refers to the extent to which we can correctly state that the introduction of concept mapping techniques caused the participants to produce more appropriate class and sequence diagrams. One of the threats to internal validity consists of the expectations of a particular result by the researcher. In this context, the researcher was responsible for marking the class and sequence diagrams produced by the two groups of participants, and she is fully aware that the scores for the diagrams may bias the hypotheses.

To this end, the marking scheme was strictly adhered to. However, in order to eliminate this threat, we should consider using independent assessors to mark the pre-test and post-test diagrams. We have not 'employed' the services of independent assessors due to logistical problems e.g., lack of funding and time constraints. Note: the case study, marking scheme and a sample of the marked diagrams were moderated by independent assessors.

Construct Validity. Construct validity refers to whether the study was actually evaluating what it was trying to evaluate. Study subjects are likely to be anxious about being evaluated and this apprehension may influence the results. We did not detect any evaluation apprehension as the students were assured that the results of the exercises would not contribute towards their overall course scores.

External Validity. The main threat to external validity is generalising our results as our sample may not be representative of all IT undergraduate students.

Conclusion

A set of four teaching modules were developed to help teachers of introductory OOAD courses and workshops integrate the concept mapping approach into their existing syllabi. The first module introduces a procedure for producing static concept maps from textual use cases that describe the functional requirements of a system. The second module presents a set of transformation rules for converting the static concept map to a corresponding class diagram. The third module introduces a procedure for producing a dynamic concept map from a use case. The fourth module presents a set of transformation rules for converting a dynamic concept map to a corresponding sequence diagram.

These modules can be used as a stand-alone workshop or integrated into existing OOAD courses or workshops. There are several clear benefits to adopting these modules:

- The modules provide a good foundation for identifying essential concepts to represent the problem domain.
- The modules present techniques to help students produce appropriate abstractions of real-world problems to be represented by UML diagrams.
- Students are explicitly taught how to produce concept maps representing the static and dynamic aspects of the problem domain.
- Students are explicitly taught how to convert the static and dynamic concept maps to respective class and sequence diagrams.
- The modules include graduated exercises for students to practise with.

The modules were integrated into an introductory OOAD course and initial evaluation results have confirmed the teaching modules to be highly effective in helping students produce more appropriate UML diagrams.

References

[1] Shelly, G. B., Cashman, T. J., and Rosenblatt, H. J., *Systems Analysis and Design*: San Val, 2003.

[2] Liu, D., Subramaniam, K., Far, B. H., and Eberlein, A., "Automating Transition from Use-Cases to Class Model," presented at The Canadian Conference on Electrical and Computer Engineering, 2003.

[3] Sharp, H. C. and Newton, M. A., "Object-Oriented Analysis and Design: A Question of Approach," presented at the 1st International Conference on Software Engineering in Higher Education, 1994.

[4] Ryan, C. and Al-Qaimar, G., "A Cognitive Perspective on Teaching Object Oriented Analysis and Design," in *RMIT Computer Science Postgraduate Conference*, 1996.

[5] Booch, G., *Object Solutions Managing the Object-Oriented Project*. Menlo Park, CA: Addison-Wesley Publishing Company, Inc., 1996.

[6] Martin, J. and Odell, J., *Object-Oriented Methods: A Foundation*. Eaglewood Cliffs, New Jersey: Prentice-Hall International Inc, 1995.

[7] Svetinovic, D., Berry, D. M., and Godfrey, M. W., "Increasing Quality of Conceptual Models: Is Object-Oriented Analysis That Simple?," presented at Proceedings of the 2006 International Workshop on Role of Abstraction in Software Engineering, Shanghai, China, 2006.

[8] Holmboe, C., "The Linguistics of Object-Oriented Design: Implications for Teaching," in *Proceedings of the 10th annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. Caparica, Portugal: ACM, 2005.

[9] Frosch-Wilke, D., "Using UML in Software Requirements Analysis - Experiences from Practical Student Project Work," presented at Informing Science & Information Technology Education Joint Conference, Pori, Finland, 2003.

[10] Sien, V. Y. and Carrington, D., "A Concepts-First Approach to Object-Oriented Modelling," presented at Proceedings of the 3rd IASTED International Conf. on Advances in Computer Science and Technology, Phuket, Thailand, 2007.

[11] Sien, V. Y. and Carrington, D., "Using Concept Maps to Produce Sequence Diagrams," presented at Proceedings of the IASTED International Conference on Software Engineering, Innsbruck, Austria, 2008.

[12] Novak, J. D. and Cañas, A. J., "The Theory Underlying Concept Maps and How to Construct and Use Them," in *Technical Report IHMC CmapTools 2006-01 Rev 01-2008*. Pensacola, Florida, 2008.

[13] Brilliant, S. S. and Knight, J. C., "Empirical Research in Software Engineering: A Workshop," *ACM SIGSOFT Software Engineering Notes*, vol. 24, pp. 44-52, 1999.

[14] George, J. F., Batra, D., Valacich, J. S., and Hoffer, J. A., *Object-Oriented Systems Analysis and Design*, 2nd ed: Pearson Higher Education, 2007.

[15] Dobing, B. and Parsons, J., "How UML Is Used," *Commun. ACM*, vol. 49, pp. 109-113, 2006.

[16] Bolloju, N. and Leung, F., "Assisting Novice Analysts in Developing Quality Conceptual Models with UML," *Communications of the ACM*, vol. 49, pp. 108-112, 2006.

[17] Oivo, M. and Basili, V. R., "Representing Software Engineering Models: The Tame Goal Oriented Approach," *IEEE Transactions on Software Engineering*, vol. 18, pp. 886-898, 1992.

[18] Briand, L. C., Morasca, S., and Basili, V. R., "An Operational Process for Goal-Driven Definition of Measures," *Software Engineering, IEEE Transactions on*, vol. 28, pp. 1106-1125, 2002.

[19] Chiou, C. C., "Effects of Concept Mapping Strategy on Learning Performance in Business and Economics Statistics," *Teaching in Higher Education*, vol. 14, pp. 55-56, 2009.

[20] Puntambekar, S., Stylianou, A., and Hübscher, R., "Improving Navigation and Learning in Hypertext Environments with Navigable Concept Maps," *Human-Computer Interaction*, vol. 18, pp. 395-428, 2003.

[21] Markow, P. G. and Lonning, R. A., "Usefulness of Concept Maps in College Chemistry Laboratories: Students' Perceptions and Effects on Achievement," *Journal of Research in Science Teaching*, vol. 35, pp. 1015-1029, 1998.

[22] Roy, D., "Using Concept Maps for Information Conceptualization and Schematization in Technical Reading and Writing Courses: A Case Study for Computer Science Majors in Japan," presented at Proceedings of IEEE International Professional Communication Conference, Montreal, QC, 2008.

About the Author

Dr Ven Yu Sien

Ven Yu Sien is a PhD student in Computer Science at the School of Information Technology and Electrical Engineering, University of Queensland, Australia and a lecturer in the Department of Information Technology at HELP University College, Malaysia. Prior to her appointment to lectureship, she worked extensively in the commercial software development field, particularly in consultation, design and development of application software for multi-national companies in London, England. Ven Yu's research interests include object-oriented analysis and design and software engineering education.