Assessing Object-oriented Programming Skills in the Core Education of Computer Science and Information Technology: Introducing New Possible Approach

[1]NORAZLINA KHAMIS, [2]SUFIAN IDRIS, [3]RODINA AHMAD & [4]NORISMA IDRIS
[1,3]Department of Software Engineering,
Faculty of Computer Science & Information Technology, University of Malaya
50603 Kuala Lumpur, Malaysia.
azlina@um.edu.my
[2]Department of Computer Science
Faculty of Technology & Information Science, Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia
[4]Department of Artificial Intelligence
Faculty of Computer Science & Information Technology, University of Malaya
50603 Kuala Lumpur, Malaysia

Abstract: - Deciding on how to evaluate each students programming skills is one of the largest challenges facing educators who teach object-oriented courses. Traditional assessment of programming education is with Grade. Quite often students get good grades in programming but still facing great challenges or have difficulties to take on real programming jobs. This research focus on how we addressed this challenge in object oriented programming course by proposing a new assessment method to assess students' object-oriented programming skills. The process begins by identifying generic object-oriented skills that students should acquired. In this paper we discuss the issues on object-oriented programming assessment and our proposed solution for a new assessment model. This followed by an approach taken in the process of identifying the object-oriented skills using Delphi technique. Delphi technique is a structured multi-step process that uses a group of experts to achieve a consensus opinion. We present the methodology of three Delphi processes to identify object-oriented programming skills. The identified skills will be used to guide both the coverage of student learning assessments and can be used by instructors to identify what topics merit emphasis.

Key-Words: - Object-oriented programming, assessment, programming skills, Delphi technique, goal questions metrics

1 Introduction

According to the American Association of Higher Education [1], assessment is a systematic process of looking at student achievement within and across courses by gathering, interpreting and using information about student learning for educational improvement. The term "assessment" is often used in different contexts and means different things to different people. The result for assessment is used for continuous improvement of teaching and learning. Duse and Duse [14] in their research, highlight that assessment was always a controversial operations and the importance of assessments is indisputable. Assessments disclose and close the gap between curricular goals and student outcomes. What are the purposes of conducting an assessment? The main idea of assessment should be to improve student learning. Other purposes of assessment include:

- To identify problems within a particular curriculum and establish an emphasis of particular skills areas within the curriculum. Such assessment can indicate the degree of success of a program or to foster continuous improvement.
- To provide information to students about how well they have learned a particular topic and where they are having difficulty.
- Students learning studies can be used to communicate learning achievement for specified outcomes. Students can determine their overall strengths and weaknesses in learning the course.
- To provide information to educators about individual students' understanding on particular topic

Regardless of the specific purpose of an assessment, incorporating an assessment program in classes offers a

way to reflect about what we are doing and to find out what is really happen in classes. It provides a systematic way to gather and evaluate information to be use to improve our knowledge on students in particular courses. By using assessment, we can help students at assessing their own skills and knowledge. In this paper we will discuss issues and possible solution in enhancing the assessment and measurement by specifically focusing on object-oriented programming course assessment. We begin our discussion with the brief introduction on the concept of assessment in Section 2. Followed by the discussion on issues of the current assessment in CS and IT Programming Education in Section 3. We present our solution for enhancing the assessment in Section 4 and we conclude with discussion on our further work.

2 Concept of Assessment in Education

Assessment can be divided into formative assessment and summative assessment. These two categories of assessment are distinguished by the point at which the assessment occurs in a program. Formative assessment is usually being carried out at the beginning or during a program. It provides the immediate proof for student learning in certain course or certain point in a program.

One of the examples for formative assessment techniques is classroom assessment. This technique is used to collect feedback on how well the students are learning what they are being taught. The purpose of classroom assessment is to provide faculty and students with information and insights needed to improve teaching effectiveness and learning quality.

Another type of assessment is known as summative assessment. Summative assessment is used to check the level of learning. It is usually being carried out at the end of the program in order to generate a grade that reflects the student's performance. The most common techniques of summative assessment are the final written examination. Students will be given a grade based on the correctness of their answers.

Approaches to assessment can also be categorized into norm-referenced assessment and criterion-referenced assessment [2]. Norm-referenced assessment "remains the dominant… (approach) within higher education and 'naturally' preferred by most markers" [3]. Biggs [4] acknowledges that one of the main reasons for implementing norm-referenced assessment is for administrative convenience, but asserts that there is "no *educational* justification for grading on a curve" (p. 69).

A criterion-referenced model is an assessment model where the purpose is to assess the extent to which a student has achieved the goals of a course. In this context, the assessment is carried out aligned with specified criteria. Results are expressed in terms of how well a given student's performance matches set criteria. This model is usually independent of any other student result. The standards are set before teaching takes place. A grade is assigned on the basis of the standard the student has achieved on each of the criteria.

2.1 Concept of reliability and validity of assessment

Assessment is an indirect measurement. When we assess a student, we are making inference about an abstract quality of the student in terms of their capabilities, knowledge, motivation or attitude. Measurement means we involve quality concept. When we talked about quality of assessment, there are two issues will rise; validity and reliability.

The traditional definition of validity is the extent to which a test measures what is was designed to measure. Validity is the most important consideration in choosing assessment. Most recent discussion on validity derives from Messick[5] works. According to Messick, if the test does not measure what it supposed to measure, then the use of it is misleading.

We can divide validity into four types; predictive validity, content validity, constructs validity and concurrent validity. In predictive validity, it concerned whether the test predicts accurately the future performance, while concurrent validity is concerned about whether the test gives significantly the same results as another test of the same skill. Construct validity is concerned about whether the test is an adequate measure of the construct, that is the underlying skill being assessed. Lastly content validity is more straightforward where it concerns whether the test covers the skills necessary for good performance.

Another issue that relates to quality of assessment is reliability. Reliability is concerned with the accuracy with which the test measures the skill it is designed to measure. Would an assessment produce the same or similar score on two particular time or given by two assessors? Reliability therefore relates to consistency of students performance and consistency in assessing the performance. There are many reasons that assessment may turn out to be unreliable. Reliability can also be categorized into interrater reliability, stability and alternate-form reliability. In interrater reliability, it refers to the consistency among the assessors. If the test has only one assessor, interrater reliability is not an issue. If more than one assessor, then a consensus is needed on the standards marking scheme. While in stability, it refers to consistency over time. A test is said to be stable if we deliver the test again to the same group of students would get the same marks. Another reliability category is alternate-form reliability. It refers to consistency across forms. If different items that test the same

concept get similar marks, the test has alternate-form reliability[6].

3. Methods and Issues with Current Assessment in Programming Education

One of the biggest challenges for educators who teach programming courses is how to evaluate students programming skills [7]. There are numerous researches focuses on teaching and learning object-oriented programming [15][16] but few on evaluation. Thus, for this research, we focus on the aim to improve the assessment of one specific course in CS and IT Education, which is object-oriented programming (OOP). Object oriented is one of the most important programming paradigms. Virtually all universities that have CS and IT degree will include OOP as one of their core courses. We conducted a survey to identify issues with current OOP assessment amongst Malaysian universities [8][9].

3.1 Assessment Methods for Programming Courses

From our survey, we found that respondents conducts assessment for students programming skill using these approaches; written examination, practical examination, programming assignments and viva.

a) Written examination
   This is a traditional written examination covering the concepts of object oriented programming. The examinations include multiple choices, true/false or structured questions and it is done under comparatively short timed conditions. Major criticism for this type of assessment is because of the relatively short time allowed, answers may unavoidably be apparent and not all the learning outcomes may be assessed.
b) Practical Examination
   In practical exam, students are required to write applications in a controlled environment and they must work individually. This approach requires students to be able to applied concept and technique in object oriented that they have learned during the class. Thus, through practical examination, the actual programming skills can be determined.
c) Programming assignment
   Programming assignment is type of assessment where it gives the opportunity to the students to implement techniques of programming explained in class. It also provides the opportunity for the students to practice their programming skills. Assignments may be done individually or in groups. Students is also allowed to refer to external outsource. The major problem with this technique is that it is subject to plagiarism.
d) Viva
   Other technique that is being used is viva. Viva is the techniques of assessment where the students must answered the questions orally. In a comparatively short of time, it is possible to ascertain both what the student knows and the depth of understanding. But the problems with this type of assessment is that it requires one-to-one meeting with the students and for programming courses it is quite difficult to evaluate their programming style or programming skills thru answering the questions orally.

3.2 Issues in assessment of OOP course

Regardless any approaches used, students need to answer a set of questions or problems by applying OO concepts that they have learned. Grade will be given based on the correctness of their answers. Most of the respondents highlight the problem with this approach is that grades do not reflect the actual programming skills that the students have acquired.

Different educators have different expectation when marking the student's work. So it is quite difficult to justify if student's who get good grades in OOP at one university have the same OO skills with another students who also get a good grades in other university. Grades only represent the extent to which students have successfully met the university requirements and it is impossible to make inferences about what they know by looking at their grades.

Another issue that arise from our survey is does the assessment is design to assess student's OO skill? What are the guidelines when educators design the questions for OOP assessment? Based from the survey, we found that only three out of five universities apply Bloom's taxonomy concept when designing the questions for their OOP assessment. Bloom's taxonomy is a hierarchical classification of cognitive skills and capabilities. The taxonomy is popular and useful as a tool to facilitate appropriate questioning [10].

One of the drawbacks of using this model is that it is only measures the cognitive skills and capabilities. What about OO skills? Can we identify OO skills based on this model? When these questions were highlighted to the educators, most of them who involved in this survey agree that there is a need to have a model that specifically serves as a benchmark to conduct OOP assessment. From the model, educators can easily identify either their student's have acquired the OO skills. Based on the results from the survey we develop a

model that can serve as a benchmark for assessing OO skills.

Thus we need to develop a new way of thinking about assessment to deal with the issues that are emerging as assessment takes on this broader definition and purpose. We have identified the issues related with current assessment method. We proposed a new assessment model that can be used to support the assessment process specifically for OOP course. The model is based on the criterion-referenced assessment model that has been discussed earlier. It provides a better approach in assessment compared to the traditional approach, which is norm-referenced assessment model.

Figure 1 shows our assessment model in a diagrammatic form. We apply Goal Questions Metric (GQM) approach when developing the model. In the following section we will describe the approach that we used to develop the model.

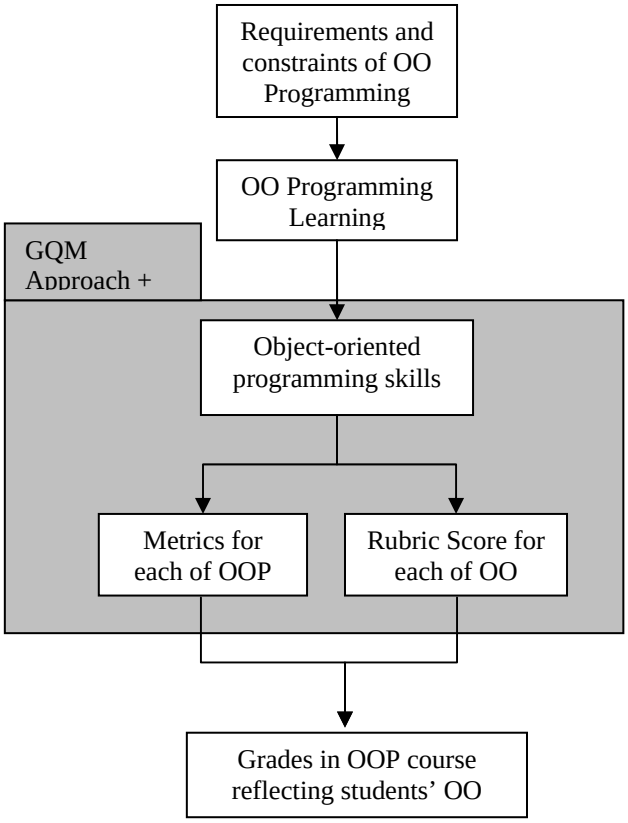

Fig. 1. Assessment Model

4. Implementation of the GQM approach

Goal Questions Metrics (GQM) is a paradigm for developing metrics program to support software development and maintenance. However, the basic concept of GQM is applicable whenever effective metrics are needed to assess satisfaction of goals. In our research context, the approach is applicable because we need to measure how far students achieve the goals in learning OOP skills by establishing an appropriate metrics to measure it.

The GQM approach was initially developed by Victor Basilli and his colleagues in the 80s [11]. This approach is characterized by two processes which are; top down refinement of measurements goals into questions and then into metrics, and bottom up analysis and the interpretation of the collected data.

In our research work, we adapt this approach when developing a model for assessing OO programming skills. The next paragraph will describe in details how we applied the GQM approach in developing the model. We begin by identifying the goals (G) of learning OO concepts that should be achieved by the students. This is then followed by the questions (Q) and the metrics (M).

4.1 Defining Goals

To help us formulate the model using GQM approach, we conduct an interview with a few experts in OO programming. The process of identifying OO skills also will be done based on the reviewing learning objectives for OO programming courses. The learning objectives are taken from several Computer Science and Information Technologies faculties in Malaysian universities. Learning objective from ACM Computing curricular will also serves as a guideline when identifying OO skills.

4.1.1 Mapping Object-oriented Concept with Learning Objectives

One of the most influential programming paradigms today is object oriented programming and it is widely used in education and industry. It also presents unique challenges with regard to teaching, learning and assessing student progress. Many people claims that learning OO is difficult [17, 18, 19]. One possible reason that learning OO is so difficult may be due to reason of not comprehensively understands the fundamental concepts of OO approach. To date, not much review and literature has been done on identifying the fundamental concept of OO, except one study by Amstrong [12]. She conducted a thorough review and literature on OO and come out with taxonomy of OO (refer Figure 2.). The next paragraph will briefly discuss the fundamental concept that has been described by Amstrong.

These eight fundamental concepts of OO must be skillfully applied by those who develop OO program. Therefore, these concepts must be incorporated in learning objectives for OOP courses. An analysis has been done to propose comprehensive learning objectives for OOP courses that can be adopt by any faculties offering OOP courses. Our analysis were based on the learning objectives for OOP as proposed by ACM

Computing Curricula 2001[20], learning objectives for OOP from several universities in Malaysia and a revised learning objectives as suggested by Decker [21].

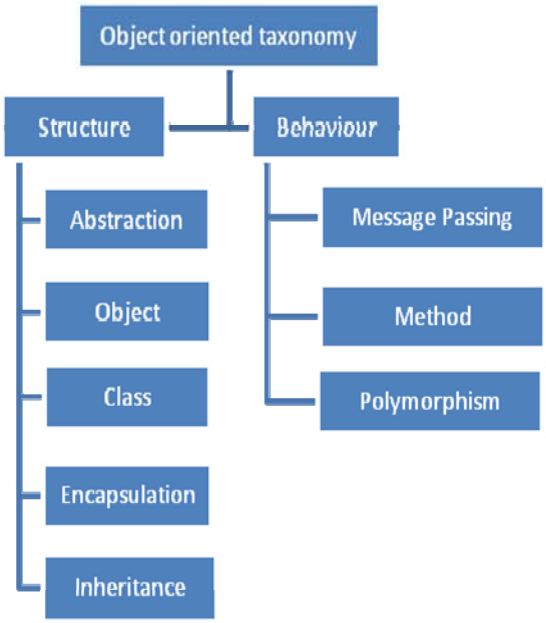

Fig. 2. Taxanomy of object-oriented Concepts

We also incorporated Delphi techniques in the process of defining the object-oriented programming skills.

4.1.2 Delphi Technique

The Delphi was used to consult a body of expert, gather information and formulate a group consensus while limiting the complications and disadvantages of face-to-face group interaction. In our case, the Delphi is used to derive generic OO competencies that students should acquire when they learn OOP course.

The Delphi technique begins with the initial development of a questionnaire focusing on the identified problem by the initiator. Next step, an appropriate respondent group will be selected and the earlier prepared questionnaire will be mailed to each of them. The respondent will answer the questionnaire independently and they will return it to the initiator. The initiators of the questionnaire will summarize responses received, develop a feedback summary and a second questionnaire for the same respondent group. After reviewing the feedback summary, respondents will continue rate priority ideas included in the second questionnaire, then mail back the responses. The process is repeated until investigators reach an agreement on a topic that being discussed. A final summary report is the will be issued to the respondent group.

4.1.3 Methodology

The methodological approach that will be used combines theory and practice. In the theoretical part, the fundamental aspects OO programming, as well as its definition, main features, and learning objectives will be reviewed. Additionally, the existing literature will be reviewed to identify the source that can be used as a guideline for identifying the OO skills. During the second stage, an empirical application of the technique will be formulated in order to obtain opinions from OO experts on the measuring of the OO skills. Further stages will be:

a) Experts Selection:
Experts who will participate in this investigation of the Delphi method will belong to one of these three groups:
- Educators from several universities who have experience in teaching OO programming for more than 5 years.
- Experts/Educators or professors who conduct an extensive research on OO domain.
- Representative from several software development companies who involved in object oriented software development.

b) Questionnaire design:
The questionnaire will focus primarily on the following aspects:
- Clarity and comprehension of the object oriented key concept
- Clarity and comprehension of Learning Objectives (LO) for OO programming courses. This LOs are derived from the analysis of LO's as proposed in ACM CC2001, a revised version of ACM CC2001 and LO's from OOP courses among Malaysian universities. This LO's will serve as a references or starting point for OO expert to define a details OO programming skills related to particular LO's. Experts are then will identify the specific OO skills based on the LO's and OO concepts.

c) Delphi applications:
Once the questionnaire is ready, the whole process described above will be implemented (questionnaire sending, reception, feedback, etc). As soon as the consensus is reached, the process will be complete

The output from this process will be the goals that need to be achieved for those who learn OO programming, which is the first level in QGM approach (refer Table 1).

Table 1. Learning Objectives for OOP Course

LO Code Identifier	Learning Objectives (LO)	Object oriented key concept
LO1	Design, implement, test and debug simple programs in an object oriented programming language	Abstraction, object, class, inheritance, encapsulation, message passing, method, polymorphism
LO2	Describe the concept of object interaction	Object
LO3	Describe how the class mechanism supports encapsulations and information hiding	Encapsulation, class
LO4	Design, implement and test the implementation of 'is-a' relationship among classes using a class hierarchy and inheritance. Distinguish between the superclass and the subclasses in these relationship	Class, inheritance
LO5	Compare and contrast the notions of overloading and overriding methods in an object-oriented language	Method
LO6	Explain the relationship between the static structure of the class and dynamic structure of the instances of the class, especially in the context of how dynamic dispatch is involved in subtype polymorphism	Class, Polymorphism
LO7	Describe how iterators access the elements of the collection	Method, message passing
LO8	Describe the difference between subtype polymorphism and inheritance	Polymorphism, inheritance
LO9	Create a collection, insert elements into a collection, and iterate over elements in a collection	Class

4.1.4 Current progress

A few experts has been selected and invited to collaborate with this research. Most of the experts are educators, researchers and software developers who have experience in object-oriented programming. A first round Delphi method is currently being conducted by distributing the first round questionnaires to the expert. The questionnaire is divided into two parts. Part one consists of questions related with identifying the core concept for object-oriented programming. Experts were then asked to put the weightage for each of the OO concept according to their importance. In this phase, experts can include another OO concept that not included by the researcher from the questionnaires.

In part two, experts need to identify the specific OO skill for each of the OO concept in part one. The OO skills are basically the expectation of the experts on how students will apply these OO concepts in their program source code. This will include the OO concept that has been added by the expert in part 1.

A sample answer has been received from the Delphi round one. Some of the feedbacks are shown in Table 2 below. The feedback from the experts will be collected and analyse by the researcher. The next step will be the second round of the Delphi study.

Table 2. Part of Delphi Round One Feedback

OO Concept	**Specific OO skills on how students should apply these concept in their program source code**
Class	Identify the proper class, method and attribute to solves a particular problem
Object	Able to create a structure of object collaborating among themselves to carried out task (vs. one object doing everything itself)
Abstraction	Define the class at the proper level on abstraction
Encapsulation	Able to appropriately encapsulate attribute method of a class
Inheritance	Able to create an inheritance hierarchy of classes (not by convenience – should be is-a

	relationship)
Polymorphism	The ability to send the same message to different object which then invokes different responses
Message passing	Able to construct a message to an object based on the signature & description of the class
Method	Able to define accessors and mutators by method

4.2. Defining Questions

The next step is to define a set of questions that will help to achieve the goals. For each of the OO skills that have been identified, a question will be asked on how to achieve the goals. This question will help to defined what factors or attribute that are related with the goals. In our context, the questions will reflect on identifying concept of OO that related with the OO skills. There are eight OO core concepts that need to be captured for those who learned OO [12]. These concepts are: class, object, inheritance, polymorphism, encapsulation, method, message passing and abstraction.

4.3. Defining Metrics

For each of the questions that we have identified earlier, we will come out with a set of metrics that serves to answer each of the questions. This means for each of OO skills that related with OO concept, we will identify what are the suitable metrics that we can use to measure the goals. For example, we can apply OO metrics like Number of Class (NOC) to identify number of classes that has been identify in the program source code.

4.3.1 Object-oriented metrics

Analyzing OO systems in order to evaluate their quality gains its importance as the paradigm continues to increase in popularity. Consequently, several object oriented metrics have been proposed to evaluate different aspects of the system. The next paragraph will discuss briefly on OO metrics as proposed by Chidamber and Kemerer [22].

Metric 1: Weighted methods per class (WMC)

This metrics refers to the number of methods implemented within a class or the sum of the complexities of the methods. The predictor of how much time and effort required to develop and maintain the class is using the number of methods and the complexity of the methods.

Calculation:

For a class C with methods M_1, M_2,…, M_n, with respective weight of c_1, c_2,…., c_n, the calculation is

$$\text{WMC}= \sum_{i=1}^{n} c_i$$

Metric 2: Depth of inheritance tree (DIT)

DIT refers to the concept of the depth of a class within the inheritance hierarchy is the maximum number of steps from the class node to the root of the tree. DIT concept is measured by the number of ancestor classes. The length of the maximum path from the node to the root of the three is DIT.

Metric 3: Number of children (NOC)

NOC refer to the number of immediate subclasses subordinate to a class in the hierarchy. It serves and an indicator of the potential influence a class can have on the design and on the system.

Metric 4: Coupling Between Object Classes (CBO)

CBO refers to the concept of amount of other classes to which a class is coupled. CBO is normally measured by calculating the amount of discrete non-inheritance related class hierarchies on which class depends. Unnecessary coupling is detrimental to modular design and prevents reuse.

Metric 5: Response for Class (RFC)

RFC is the number of the set of all methods that can be invoked in response to a message to an object of the class or by some method in the class. All methods available within the class hierarchy were included. Complexity of the class can be seen through the number of methods and the amount of communication with other classes.

Metric 6: Lack of Cohesion Metric (LCOM)

Measures the correspondence of methods in a class by instance variable or attributes. A highly cohesive module should stand alone.

Metric 7: Cyclomatic Complexity (CC)

Cyclomatic complexity (Mccabe) is used to evaluate the complexity of an algorithm in a method. It is a count of the number of test cases that are needed to test the method comprehensively.

Calculation:

$$CC = \text{edges} - \text{nodes} + 2$$

Based on the review towards these OO metrics, some of them are suitable to be used to extract OO concept from the source code. The suitable metrics can be tailored so that the output of the metrics can be used as one of the approach to extract certain OO information. Table 3 below shows the related OO metrics with OO concepts:

Table 3. OO metrics with respective OO concepts

OO metrics	OO Concepts
Cyclomatic complexity (CC)	Method
Weighted Methods per Class (WMC)	Method
Response for Class (RFC)	Message
Depth of Inheritance (DIT)	Inheritance, Class
Number of Children (NOC)	Inheritance, Class

The output from this process is a model that consist a list of measurable OO skills that students should acquire when they learn OO programming courses. The example of our model development is shown in three steps below.

Step 1: Defined Goals

Goal	Detailed LO
LO2: describe the concept of object interaction: {G.2}	1. Understand that an object-oriented program consists of several different objects {G.2.1}
	2. Understand object connection within certain structures {G.2.2}
	3. Understand the dynamics of these object structures {G.2.3}

Step 2: defined the questions:

Goal	Detailed SubGoal (if Applicable)	Defined Questions
LO2: describe the concept of object interaction: {G.2}	Understand that an object-oriented program consists of several different objects {G.2.1}	How to identify object in a program source code {Q.2.1}
	Understand object connection within certain structures {G.2.2}	Do students know how to create a connection between objects? {Q.2.2}
	Understand the dynamics of these object structures {G.2.3}	Do students understand connection between objects? {Q.2.3}

Step 3: Defined metrics

Goal	Detailed SubGoal (if Applicable)	Defined Questions	Defined metrics
LO2: describe the concept of object interaction: {G.2}	Understand that an object-oriented program consists of several different objects {G.2.1}	How to identify object in a program source code {Q.2.1}	Number of objects involved {M.2.1}
	Understand object connection within certain structures {G.2.2}	Do students know how to create a connection between objects? {Q.2.2}	Complexity of the method-call sequence {M.2.2}
	Understand the dynamics of these object structures {G.2.3}	Do students understand connection between objects? {Q.2.3}	Complexity of structural changes (creating or deleting objects and references) {M.2.3}

4. Current progress

Currently the research is still in Delphi round one. Only

few experts have submitted their answers and the rest still in the progress. Delphi round two can only be started after all the feedback from the experts has been collected and analyse. This is the major drawback of Delphi study because it involves human for providing data. It is hope that the process of collecting data from the experts will run smoothly and the development of the assessment model can be done.

5. Conclusion

In this paper, the issues and possible solution in enhancing the assessment of Computer Programming courses, specifically OOP are presented. Current work on the approach used to develop assessment model for object-oriented programming assessment are being discussed. The extent of the results for the analysis of learning objectives on object-oriented programming courses can be used to prepare an assessment model are also shown. The process of collecting and identifying OO skills based on the interviews with OO experts and reviewing learning objectives for OO courses to complete the assessment model is still being carried out A Delphi study were incorporated in the process of developing the assessment model. For further work, the focus isv on the development of the assessment method. The appropriate assessment instrument to the validate model need to be identify.

References

[1] AAHE. What research says about improving undergraduate education", *AAHE Bulletin*, AAHE, Washington, DC,1996.

[2] Dunn, L., Morgan, C., O'Reilly, M., & Parry, S. "The Student Assessment Handbook", London: Routledge Falmer. 2004.

[3] Rust, C., Price, M., & O'Donovan, B. "Improving students' learning by developing their understanding of assessment criteria and processes " *Assessment & Evaluation in Higher Education*, 28(2), 147.2003

[4] Biggs, J. "What the student does: Teaching for enhanced learning", *Higher EducationResearch and Development*, 18(1), 57. 1999.

[5] S. Messick, *Educational Measurement*, 3rd Ed. Oryx press. 1993. Pp13-103.

[6] J. Salvia and J.E Ysseldyke. *Assessment*, 6th Ed, Houghton Mifflin. 1995.

[7] J. Bennedsen and E.M Carpersen, "A Competence Model for Object Interaction in Introductory Programming" in Proceeding of 18th Workshop of the Psychology of Programming Interest Group, 2006

[8] Norazlina Khamis and Sufian Idris, "Investigating Current Object Oriented Programming Assessment method in Malaysia's Universities", in Proceeding of ICEEI 2007, Bandung, Indonesia, 2007.

[9] Norazlina K., and Sufian. Idris. "Issues with Object Oriented Programming Assessment in Malaysia's Univerisities" in Proceeding of 1st International Malaysian Educational Technology Convention, 2007.

[10] Anderson, L. W., & Krathwohl, D. R.. *A taxonomy learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. New York: Longman, 2001.

[11] V.R Basilli, Gianlugi. C and Rombach H. D, The Goal Question Metric Approach, 1996 Available *on* http://wwwagse.informatik.uni-kl.de/pubs/repository/basili94b/encyclo.gqm.pdf.

[12] D. J Amstrong, The Quarks of Object Oriented Development, *Communication of The ACM, 49(2)*, 2006, 123-128.

[13] Norazlina. K, Sufian. I and Rodina A, Applying GQM Approach Towards Development of Criterion-Referenced Assessment Model for OOP Course in Proceedings Of World Academy Of Science, Engineering And Technology Volume 26 December 2007 Issn 1307-6884.

[14] Duse Dan-Maniu and Duse Carmen S., Desirable performances and limits of students' assessment in engineering education in Proceedings of the 2nd WSEAS/IASME International Conference on Educational Technologies, Bucharest, Romania, 2006. pp 70-73.

[15] Norfazlida M. S, Abdullah M.Z, Sufian. I and Zarina. S., Designing an Understanding And Debugging Tool (UDT) for OOP Languange, in WSEAS Transaction on Computers, Issue 2, Volume 4, February 2005.

[16] Stelios X, Maya. S. and Vassilios D., Teaching Fundamental Notions of Object-oriented Programming With ObjectKarel in WSEAS Transactions on Advances in Engineering Education, Issue 11, Vloume 3, November 2006.

[17] Susan Bergin & Ronan Reilly. 2005. Programming: factors that influence success. ACM SIGCSE Bulletin Volume 37 , Issue 1. Pp 411-415.

[18] R. Lister, E. Adams, S. Fitzgerald, W. Fone, J. Hammer, M. Lindholm, R. McCartney, J.E

Mostrom, K. Sanders, O. Seppala, B. Simon and L.A Thomas, "A Multi National Study of Reading and Tracing Skills in Novice Programmers" ACM SIGCSE Bulletin, 36,4. 119-150, December 2004.

[19] M. McCracken, V. Almstrum, D.Diaz, M. Guzdial, D. hagen, Y. Kollokant, C. Lazer, L.A. Thomas. I. Utting and T Wilusz, "A Luti national Study of Assessment of Programming Skills of First Year CS students" in SIGCSE Bulletin, 33, 125-140,.Dec 2001.

[20] Computing Curricula 2001 Computer Science, Final Report, December 2001.

[21] Decker, A. "How Studente Measure Up: An Assessment Instrument for Introductory Computer Science" Dissertation. 2007.

[22] SR Chidamber, CF Kemerer, A metrics suite for object oriented design C MIT - Software Engineering, IEEE Transactions on, 1994 - ieeexplore.ieee.org