Peer and meta-assessment in a project-based large systems engineering course

Niva Wengrowicz
Dept. of Education in Science and Technology
Industrial Engineering and management
Technion, Israel Institute of Technology
Haifa, Israel
nivawen@technion.ac.il

Yehudit Judy Dori
Dept. of Education in Science and Technology
Technion, Israel Institute of Technology
Haifa, Israel
yjdori@technion.ac.il

Dov Dori
Industrial Engineering and management
Technion, Israel Institute of Technology
Haifa, Israel
dori@technion.ac.il
Engineering System Division
Massachusetts Institute of Technology
Cambridge, USA
dori@MIT.edu

Abstract

Modern engineering education programs aim to endow students with knowledge, skills, and attitudes necessary to become successful engineers. Engineering students should be able to conceive, design, implement, and operate complex value-added engineering systems—products, processes, and services. Our research goal was to develop and validate a formative assessment method and tool for a large mandatory undergraduate engineering course. The research population consisted of about 130 undergraduate students who studied a systems and information engineering course in a project-based learning (PBL) environment. The students constructed models of Web-based systems using two conceptual modeling approaches and languages: Object Process Methodology (OPM) and Unified Modeling Language (UML). Formative assessment was carried out along these PBL processes. We developed a Web-based formative assessment method for this PBL engineering course that is aligned with the course objectives and accounts for its large scale. The method consisted of peer-assessment and meta-assessment. In the peer assessment, each student ranked four peer projects according to pre-defined criteria via a specially-designed Web-based tool. Finally, we carried out meta-assessment, in which the course team assessed each student based on the quality of his/her arguments and the thinking levels demonstrated in the peer assessment. The findings validated our meta-assessment tools.

Keywords*- conceive-design-integrate-operate (CDIO), large scale assessment, project-based learning (PBL), systems engineering education*

Introduction

Engineering Education

Modern engineering education programs aim to enrich students with the necessary knowledge, skills, and attitudes for becoming successful young engineers (Crawley, Malmqvist, Lucas, & Brodeur, 2011). The goal of such engineering education programs is to train students to be able to conceive, design, implement, and operate complex value-added engineering products, processes, and systems in modern, team-based environments. These insights formed the basis for the CDIO – Conceive-Design-Implement-Operate educational framework (Crawley, Brodeur, & Soderholm, 2008).

Most of the CDIO curriculum features are related to experiential learning (Crawley et al., 2008). This approach emphasizes the importance of active and hands-on learning in both the classroom and modern learning workspaces. CDIO facilitates students' exposure to experiences that engineers are likely to encounter during their professional lives. To enable these kinds of experiences, a typical CDIO-oriented syllabus contains significant elements of project-based learning (Crawley et al., 2011), discussed next.

Project Based Learning

Project-based learning (PBL) is a teaching method in which students are challenged with solving realistic problems that do not have a single correct answer. They are guided through a process of analyzing the problem, investigating the solution space in search for alternatives, arguing for and against them, and ultimately presenting a recommended solution (Hsieh & Knight, 2008). PBL is characterized by authentic investigation, collaboration among peers, the use of technology to support inquiry processes, and delivery of an end product (Klein & Merritt, 1994; Krajcik, Czerniak, & Berger, 1999). Through students' active participation in the project execution process, they form original opinions and are encouraged to express individual standpoints. The project fosters students' awareness of the complexity of systems they would tackle and encourages them to explore the consequences of their own values (Zoller, 1991). Researchers (Mathewson, 2005; Mayer, 2002; Paivio, 1990) have suggested that human cognition is divided into two major processing subsystems: the verbal and the non-verbal, and that knowledge is represented and manipulated through visual and verbal channels. Dori and Belcher (2005) claimed that science teaching which jointly exploits the visual and verbal channels can enhance learning and understanding processes, and improve students' learning outcomes. The information systems engineering project-based course studied and presented in this article is based on constructing conceptual modeling projects which students constructed by using visual diagrams and accompanying text in two leading conceptual modeling languages—UML(OMG UML, 2012) and OPM (Dori, 2002a).

Conceptual Models

Model-based systems engineering (MBSE) is an emerging approach to coping with the complexity of current and future systems. Conceptual models represent visually and/or textually human thoughts, ideas, and purposes. MBSE is a necessary tool for coherent thinking, sharing ideas, providing common ground for communication, and solving problems jointly. Conceptual modeling helps understand a complex problem and its potential solutions through abstraction and is therefore an important component in system engineering. MBSE facilitates the construction and communication of complex systems (Thomas, 2004), as it provides means for coordination and caters to common understanding among colleagues and customers.

Evaluating the quality of a conceptual model is a major issue professionals in the field of system engineering tackle (Akoka, Comyn-Wattiau, & Cherfi, 2008). Beside the syntax and structure correctness evaluation, which is generally used to evaluate students' outcomes, there are more criteria which can serve to evaluate undergraduates' conceptual models (Akoka et al., 2008; Lindland, Sindre, & Solvberg, 1994; Mohagheghi & Aagedal, 2007). Our evaluation instruction to the students, embedded in our specially-designed tool, included three criteria in addition to the model correctness: model completeness, documentation, and model clarity and understandability. Completeness of conceptual model means that the model contains all the requirements included in the scope (Lindland et al., 1994). The documentation focuses on the contribution of the documentation to the understanding of the considerations that guided the construction of the model and on the documentation appropriateness (Mohagheghi & Aagedal, 2007). Model clarity and understandability (MCU) are key quality characteristics of conceptual models (Akoka et al., 2008; Selic, 2003). Understandability, i.e., a model's ability to be easily understood, is a model property that has been investigated quite intensely (Cruz-Lemus, Genero, Manso, Morasca, & Piattini, 2009).

Our study focused on comparing the MCU, correctness, completeness and documentation of a given conceptual system model expressed in two different modeling languages: Unified Modeling Language (UML) and Object-Process Methodology (OPM). As we explain in the sequel, the models were constructed in a large-scale undergraduate course by teams of students, based on reverse-engineering a complex Web-based system and authoring an appropriate scope and requirements document for that system.

Unified Modeling Language

UML – the Unified Modeling Language, developed by Object-Management Group (OMG UML, 2012), is the current *de facto* software modeling language. Developed by Rumbaugh, Booch, and Jacobson in 1996 as a non-propriety modeling language (Covert, 2012; Dori, 2002b), UML currently consists of fourteen diagram types – seven structural and seven behavioral. Researches (Cruz-Lemus, Maes, Genero, Poels, & Piattini, 2010; Zugal, Pinggera, Weber, Mendling, & Reijers, 2012), who analyzed the understandability of UML, have identified many related factors, including the size of the model, control flow complexity, and the impact of hierarchy and modularity on model understandability. Meta-analysis of UML understandability (Cruz-Lemus et al., 2009) concluded that UML understandability results are mainly affected by subjects' previous experience and the size and complexity of the UML diagrams modeled.

Object-Process Methodology

Object-Process Methodology (OPM) (Dori, 2002a) is a holistic formal graphical and textual paradigm for the representation, development, and lifecycle support of complex systems. OPM enables representing systems using a highly compact set of concepts in a single diagram type and equivalent natural language. The graphical OPM model is translated on the fly to a subset of natural English, complementing the visual representation with a textual one, catering to "left brainers" and "right brainers" alike. OPM has proven to be better in visual specification and comprehension quality when used for representing complex systems compared to OMT, a UML predecessor (Peleg & Dori, 2000). OPM's formal yet intuitive graphics and text combination makes it ideal for communicating and collaborating knowledge and ideas, even between inexperienced and novice users and domain experts who are not systems engineers. By using a single holistic hierarchical model for representing structure and behavior in the same diagram type, clutter and incompatibilities can be significantly reduced even in highly complex systems, thereby enhancing their understandability.

Assessment and peer assessment in large scale engineering courses

Assessment, defined as a collection of information on students’ outcomes (Nevo, 1995), is commonly applied to evaluate students' thinking skills. Bloom's taxonomy and its revisions (Krathwohl, 2002) classify students' thinking skills into six hierarchical levels: remembering, understanding, applying, analyzing, evaluating, and creating. Effective and efficient assessment of students' thinking skills, as reflected in projects they carry out in large engineering PBL courses, requires creative approaches to cope with the need to devote much time and attention to examining a large number of different projects. In our case, each project contained several diagrams in two models of the same system in two languages.

Peer assessment is an assessment method in which students are asked to evaluate each other's work. Peer assessment helps students develop higher order thinking skills (Bedford & Legg, 2007). The peer assessment evaluation categories and their related criteria are defined in advance and should conform to the requirements presented for the task at hand. We used peer assessment as a means not only to develop students' higher order thinking skills, but not less importantly, to overcome the problem of the need to evaluate the massive amount of projects.

The wisdom of the crowd theory (Surowiecki, 2004) claims that the crowd evaluation and decision making can be more accurate and valuable than expert estimations. Based on the central limit theorem, crowd evaluation can be modeled as the mean of the probability distribution of individuals' responses, which is centered near the true mean of the quantity to be estimated. Crowds' wisdom is a function of two factors: expertise and diversity. The crowd has to be comprised of individuals with some knowledge or expertise about the question they are asked to respond to. Additionally, individuals making up the crowd should have diverse perspectives on the issue being judged. If the expertise and diversity conditions hold, at least to some extent, the wisdom of the crowd can serve as a means to assess a large amount of projects in large-scale courses. Indeed, this is exactly what we did in our research. Students assessed their peer team projects, and since we had 12 individual assessments for each project, we could use the wisdom of the crowd as a reliable tool for assessing project performance level.

Reseach Goal

Our overall research goal concerns developing and assessing a new approach and support tool for teaching and assessing undergraduate students in large PBL engineering courses. The main goal of the study described in this paper was to develop and validate an approach and a supporting our specially-designed Web-based peer assessment tool. Specifically, we focused on evaluating and validating the meta-assessment—our assessment of the students' peer assessment of the UML and OPM conceptual models developed by teams of students in a large PBL engineering course.

Research Method

Research participants and setup

The research was conducted within the framework of the course Specification and Analysis of Information Systems at the Faculty of Industrial Engineering and Management at the Technion, Israel Institute of Technology during the winter 2012-3 Semester. The course objective is to familiarize 5th Semester Industrial Engineering and Information Systems Engineering undergraduate students with analysis, modeling, design, and assessment of systems in general and of information systems in particular.

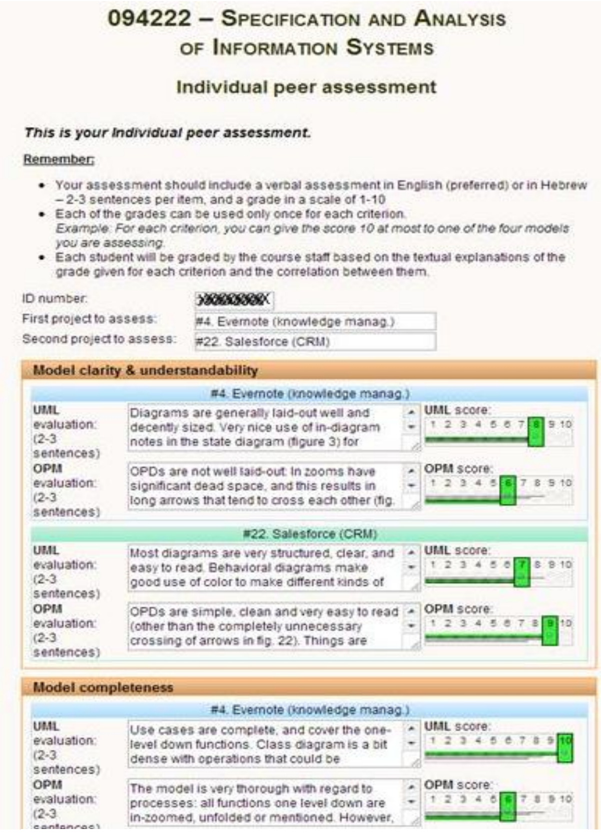
Exactly half of the 130 students who took the course were females. Employing a PBL approach, we tasked students with reverse engineering a widely-used Web-based system, such as Gmail, Expedia, or eBay, derive a requirements document for that system as if it does not exist but needs to be developed, specify. They were then required to model the system as best they can. Groups of students constructed models of these Web-based systems using both UML and OPM conceptual modeling languages. These were taught every other week alternately. The students were assigned into 23 groups of six (some of five) students divided into two teams of three (some of two). After defining the requirements document for their reverse-engineered system and getting feedback during the first three weeks of the semester, the two teams within each group modeled the same system in a crossover method: In the first half of the remainder of the semester, the first team started to model the system using OPM while the second team started to model the same system using UML. Then the teams swapped, and each team continued elaborating and refining the model that the other team in the same group had started. This way, at the end of the semester, each group had two models of the same system: one expressed in OPM and the other in UML. Since both models were constructed in part by all the six team members, each student in each team had the opportunity to practice modeling in both OPM and UML. After submitting the final project toward the end of the semester, each student was asked to assess individually four models—two OPM and two UML models of two systems modeled by two other groups based on list of categories and criteria.

To collect the large amounts of data efficiently, we developed a dedicated Web-based peer assessment tool (see Figure 1). Using this tool, students had to compare and assess the four models individually based on four categories, including (1) model clarity and understandability (MCU), (2) model completeness, (3) model correctness, and (4) documentation. The students had been exposed to the categories and their related criteria beforehand, and these were discussed extensively during the course. Each model in each project was assessed individually by 12 students. To increase grading homogeneity and make it difficult for students to collaborate on this online individual assessment assignment, each student was tasked with assessing a unique couple of projects, and a 24 hour time window was allotted for this assessment.

Research tool

The online individual peer assessment included written assessment and a corresponding grade in a scale of 1-10. To prevent students from giving 10 to all four models, the tool does not allow the assessing student to give any grade more than once for any one of the four models, so the highest set of grades can be 10, 9, 8, and 7. For example, in Figure 1 the assessing student gave 8 in the MCU category to the Evernote UML model and 6 to the OPM model. The Salesforce UML model got 7, and the OPM model got 9.

The grade each team member received for being part of the team that developed the UML and OPM models was the average of the 12 individual peer grades.



094222 – SPECIFICATION AND ANALYSIS OF INFORMATION SYSTEMS

Individual peer assessment

This is your Individual peer assessment.

Remember:

- Your assessment should include a verbal assessment in English (preferred) or in Hebrew – 2-3 sentences per item, and a grade in a scale of 1-10
- Each of the grades can be used only once for each criterion.
  Example: For each criterion, you can give the score 10 at most to one of the four models you are assessing.
- Each student will be graded by the course staff based on the textual explanations of the grade given for each criterion and the correlation between them.

ID number: [redacted]
First project to assess: #4. Evernote (knowledge manag.)
Second project to assess: #22. Salesforce (CRM)

Model clarity & understandability

#4. Evernote (knowledge manag.)

UML evaluation: (2-3 sentences) Diagrams are generally laid-out well and decently sized. Very nice use of in-diagram notes in the state diagram (figure 3) for
UML score: 8

OPM evaluation: (2-3 sentences) OPDs are not well laid-out: In zooms have significant dead space, and this results in long arrows that tend to cross each other (fig.
OPM score: 6

#22. Salesforce (CRM)

UML evaluation: (2-3 sentences) Most diagrams are very structured, clear, and easy to read. Behavioral diagrams make good use of color to make different kinds of
UML score: 7

OPM evaluation: (2-3 sentences) OPDs are simple, clean and very easy to read (other than the completely unnecessary crossing of arrows in fig. 22). Things are
OPM score: 9

Model completeness

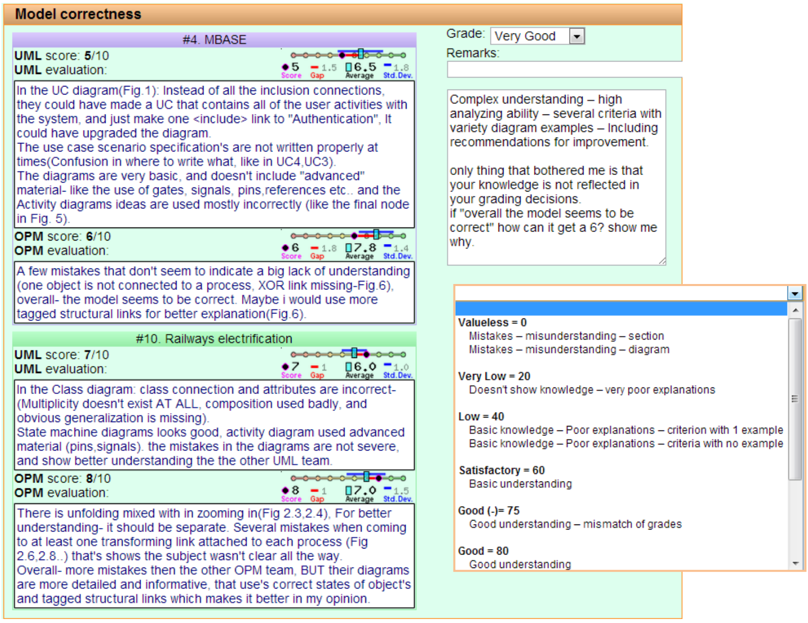
#4. Evernote (knowledge manag.)

UML evaluation: (2-3 sentences) Use cases are complete, and cover the one-level down functions. Class diagram is a bit dense with operations that could be
UML score: 10

OPM evaluation: (2-3 sentences) The model is very thorough with regard to processes: all functions one level down are in-zoomed, unfolded or mentioned. However,
OPM score: 6

Figure 1. A screenshot of the Web-based peer-assessment system – student's interface

For the written peer assessment, the students were asked to write about three sentences per item that would clarify and justify the grade given based on criteria of each category and findings from the work. A significant part (about 35%) of the final course grade each student scored was based on this peer assessment. The aim here was three-fold: (1) to endow the students with higher order thinking skills by assessing others' work, (2) to enable assessment of a large amount of projects without having to spend the prohibitive amount of time of the course team, and (3) to evaluate and grade the quality of each student's arguments in the peer assessment as reflected in her/his grading explanation and models comprehension. Since this is an assessment by the course teaching team of the students' individual assessment, we call this approach meta-assessment—assessment of assessment (see Figure 2). As noted, in this paper we focus on validating the meta-assessment.



Model correctness

#4. MBASE

UML score: **5**/10
UML evaluation:
● 5 Score — 1.5 Gap ▯6.5 Average — 1.8 Std.Dev.

In the UC diagram(Fig.1): Instead of all the inclusion connections, they could have made a UC that contains all of the user activities with the system, and just make one <include> link to "Authentication", It could have upgraded the diagram.
The use case scenario specification's are not written properly at times(Confusion in where to write what, like in UC4,UC3).
The diagrams are very basic, and doesn't include "advanced" material- like the use of gates, signals, pins,references etc.. and the Activity diagrams ideas are used mostly incorrectly (like the final node in Fig. 5).

OPM score: **6**/10
OPM evaluation:
● 6 Score — 1.8 Gap ▯7.8 Average — 1.4 Std.Dev.

A few mistakes that don't seem to indicate a big lack of understanding (one object is not connected to a process, XOR link missing-Fig.6), overall- the model seems to be correct. Maybe i would use more tagged structural links for better explanation(Fig.6).

#10. Railways electrification

UML score: **7**/10
UML evaluation:
● 7 Score — 1 Gap ▯6.0 Average — 1.0 Std.Dev.

In the Class diagram: class connection and attributes are incorrect- (Multiplicity doesn't exist AT ALL, composition used badly, and obvious generalization is missing).
State machine diagrams looks good, activity diagram used advanced material (pins,signals). the mistakes in the diagrams are not severe, and show better understanding the the other UML team.

OPM score: **8**/10
OPM evaluation:
● 8 Score — 1 Gap ▯7.0 Average — 1.5 Std.Dev.

There is unfolding mixed with in zooming in(Fig 2.3,2.4), For better understanding- it should be separate. Several mistakes when coming to at least one transforming link attached to each process (Fig 2.6,2.8..) that's shows the subject wasn't clear all the way.
Overall- more mistakes then the other OPM team, BUT their diagrams are more detailed and informative, that use's correct states of object's and tagged structural links which makes it better in my opinion.

Grade: Very Good ▾
Remarks:

Complex understanding – high analyzing ability – several criteria with variety diagram examples – Including recommendations for improvement.

only thing that bothered me is that your knowledge is not reflected in your grading decisions.
if "overall the model seems to be correct" how can it get a 6? show me why.

Valueless = 0
Mistakes – misunderstanding – section
Mistakes – misunderstanding – diagram

Very Low = 20
Doesn't show knowledge – very poor explanations

Low = 40
Basic knowledge – Poor explanations – criterion with 1 example
Basic knowledge – Poor explanations – criteria with no example

Satisfactory = 60
Basic understanding

Good (-)= 75
Good understanding – mismatch of grades

Good = 80
Good understanding

Figure 2. A screenshot of the Web-based peer-assessment system – team course interface

For data analysis we used mixed method and adopted qualitative and quantitative approaches. The students' written explanation were read, reread, and gradually analyzed from a descriptive-interpretive perspective. We created a grading scale (see Table 1) based on themes that had emerged from students' explanation combined with Bloom's taxonomy and its revision (Krathwohl, 2002). Three researchers were involved in data analysis and definition of the grading scale of the meta-assessment in order to establish its content validity.

Reliability of the meta-assessment tool was tested by five judges who were members of the course team. They were asked to evaluate several identical answers based on the grading scale. Correlation between judges ranged from .78 to .93 and were significant (p<.01).

Table 1. Meta-assessment grading scale

Evaluation	**Evaluation definition**	**Examples**
Valueless = 0	Mistakes – misunderstanding criteria – **do not fit section.**	There were some errors in the OPM that makes it unclear [Correctness or Clarity & Understandability?; without explanation, without example…]
	Mistakes – misunderstanding criteria – **do not fit diagram.**	User need to be in **unfolding** and not in **UC** diagram. ["UC" is a concept of UML while "unfolding" is a concept of OPM]
Very Low = 20	Doesn't show knowledge – very poor explanations – **no findings summarizing and no examples.**	Lines cross each other [where? what does it mean?]
Low = 40	Basic knowledge – poor explanations – **one** basic summarized criterion with **one diagram example.**	**SD1** names of objects are meaningless [What is the problem with meaningless names? All the names are meaningless? Are the meaningless names appearing just in SD?]
	Basic knowledge – poor explanations- **several basic** summarized criteria, **without any example.**	The **colors don't match** and makes it harder to understand. More **processes needed to be zoomed in** another level. [why?]
Satisfactory = 60	Basic understanding – **several basic** summarized criteria with **some diagram examples**.	On overall the work is nice but some **arrows and links cross each other** (SD0, SD1) and makes it hard to read as well as the SD1.1 diagram included **more than 5 processes** and it is over loaded.
Good (-)= 75	Good understanding – **several** criteria with **some diagram** examples – **mismatch** between the explanation and grading.	Sometimes it's hard to read the diagrams because **arrows cross** each other (fig. 1.3, 1.5) and the **caption on the lines** is too small (fig. 1.1, 1.2). Additionally, some diagrams include **unused objects** (fig 1.4, 1.5)- adding unnecessary information can create confusion [**this model was graded as 10**]
Good = 80	Good understanding – **several** criteria with **some diagram** examples – **grading correspond** to the explanations.	Most of the criteria are followed except- **missing the top-level diagram** (very important for simplifying), the class diagram is **overloaded** and **too small** (a thing that could be avoided if it weren't overloaded).
Very Good = 90	Good understanding – basic analyzing ability – several criteria with diagram examples – Including analyzing the findings (findings **generalization based on frequency**)	It might be that the **team was not fully aware about the compatibility that is required between** the behavior diagrams and the classes' attributes and methods (as shown in Class Diagram) - for example figure 5 and 6.
Excellent = 100	Complex understanding – high analyzing ability – several criteria with diagram examples – Including **recommendations for improvement**.	There is unfolding mixed with in zooming (Fig 2.3, 2.4), **For better understanding- it should be separate.** … maybe **I would use more tagged structural links for better explanation** (Fig.6).
	Complex understanding –ability to synthesis & conclude – several criteria with several different types of diagrams examples – drawing conclusions based on criteria (**findings meaning**).	A few mistakes that **don't seem to indicate a big lack of understanding** (one object is not connected to a process, XOR link missing-Fig.6), overall- the model seems to be correct.
	Complex understanding – ability to evaluate & compare - several criteria with several different type of diagram examples - drawing conclusions based on comparing to the other works (**grading explanation**).	… (Fig 2.5,2.7). Overall- **more mistakes then the other** OPM team, BUT **their diagrams are more detailed and informative**, that uses correct states of object's and tagged structural links which **makes it better in my opinion**.

Research Objectives and Hypotheses

In order to carry out the validation of our active learning approach and the tool designed to support it, we set up the following four research objectives and hypotheses: (1) Examine the correlation between the students' meta-assessment grades and their grades in the prerequisite software engineering course. We assumed that significant correlation will be found between these two students' evaluations since the prerequisite course is a condition for being admitted to our course. (2) Determine whether there was any effect of a student's level of involvement in teamwork and that student's meta-assessment grade. We assumed that students who reported relatively higher involvement will demonstrate higher thinking skills and get higher meta-assessment grades than students who were less involved in teamwork. (3) Determine whether there was any gender effect on the meta-assessment grades. Based on previous studies (Pekkarinen, 2012) although females and males studied and designed projects together in mixed teams and had the same opportunities to learn and practice modeling during the course, cultural differences in their prior education may affect their explanation and divert the meta-assessment grading. (4) Determine whether there was any order effect on the meta-assessment grades between the students who started to model with OPM (first part) and continued with UML (second part) to the crossover group of students who start with UML first and continue to model with OPM.

Findings and Analysis

A Pearson product-moment correlation coefficient was computed to assess the relationship between the meta-assessment grades and the prerequisite course grades. In line with our first hypothesis, the two grade variables were strongly correlated, r(128)=.69, p<.01. This finding indicates empirical criterion validity of the meta-assessment by preliminary course achievements.

In order to test the second, third, and fourth research hypotheses, we conducted three different t-tests. In all of them, the meta-analysis grade was the dependent variable, but they differed by the independent variable (teamwork involvement-level; gender; and first modeling language, respectively).

Teamwork involvement level was determined by the students' self-report. We asked them to evaluate their relative contribution to the teamwork, where a team usually consisted of three students in a group of six. Based on the students' answers, we divided them into two groups: those who reported involvement of themselves and contribution to the teamwork of 33% or lower and those who reported a proportion greater than 33%. In accord with the second hypothesis, there was a significant difference in the meta-assessment grades, as presented in Table 2. The meta-assessment grades of the students who reported relatively higher involvement were significantly higher than the others. The significant difference between these two groups provides evidence for construct validity by group comparison.

Table 2. Meta-assessment grades' means, standard deviation and t-test values by team work involvement level

Teamwork involvement level	***n***	***mean***	***sd***	***t(128)***
33% or lower	54	65.3	18.72	4.68***
greater than 33%	76	78.5	13.32	

*** p<.001

The third hypothesis findings indicated that there was no significant effect of gender, t(128)=.98, p>.05. In other words, there was no significant difference between the meta-assessment grades of females (M=71.54, SD=17.23) and males (M=74.46, SD=16.80).

Finally, and in line with our fourth hypothesis, there was no significant order effect, t(128)=.075, p>.05, videlicet, no significant difference was found between the meta-assessment grades of the students who started to model with OPM (M=74.09, SD=17.41) and students who start to model with UML (M=71.88, SD=16.66).

The last two findings indicate the absence of bias that might offend or unfairly penalize students due to her/his gender or the order of the modeling language used to construct the model in the course.

Discussion and Conclusions

The main goal of the study described in this paper was to develop and evaluate a new assessment approach and a supporting tool for large-scale PBL engineering courses. We have described the motivation, underlying ideas, structure, and implementation of the course, the dedicated Web-based assessment tool, and the validation of the meta-assessment, which is unique to our course. The content validity of the meta-assessment was tested by three researchers. High level of reliability was computed by correlation amongst five judges. As hypothesized, there was significant correlation between the meta-assessment grade and the prerequisite course grade, indicating empirical criterion validity. Significant difference was found between students who had reported higher-than-average involvement in their team project and those who reported lower-than-average level of involvement, indicating construct validity by group comparison. Absence of bias that might offend or unfairly penalize students due to their gender was tested and proven that there is no significant meta-assessment grades difference between females and males and last but not least there is no significant meta-assessment grades difference between students who start to model with OPM and students who start to model with UML indicating absence of bias due to the order of using the modeling languages.

The finding of the reliability and validity of the meta-assessment tool strengthens the evidence of the whole Web-based assessment tool's reliability and validity. Future research will continue to examine the large amount of data with respect to tool's validation while we continue to use and improve our PBL approach for teaching information engineering undergraduate courses. Our study and findings contribute to theory and practice of teaching and assessing project-based large-scale undergraduate engineering courses.

Acknowledgement

This work was partially funded by the EU FP7 Project #262044-VISIONAIR-Vision Advanced Infrastructure for Research and by the Gordon Center for Systems Engineering at the Technion.

References

Akoka, J., Comyn-Wattiau, I., & Cherfi, S. S.-S. (2008). Quality of conceptual schemas an experimental comparison. *2008 Second International Conference on Research Challenges in Information Science*, 197–208. doi:10.1109/RCIS.2008.4632108

Bedford, S., & Legg, S. (2007). Formative peer and self feedback as a catalyst for change within science teaching. *Chemistry Education Research and Practice*, *8*(1), 80–92.

Covert, S. (2012). OMG’s Unified Modeling Language (UML) Celebrates 15th Anniversary. Retrieved from http://www.omg.org/news/releases/pr2012/08-01-12-a.htm

Crawley, E. F., Brodeur, D. R., & Soderholm, D. H. (2008). The education of future aeronautical engineers: conceiving, designing, implementing and operating. *Journal of Science Education and Technology, 17,* 138–151. Retrieved from http://www.springerlink.com/index/10.1007/s10956-008-9088-4

Crawley, E. F., Malmqvist, J., Lucas, W. A., & Brodeur, D. R. (2011). The CDIO syllabus v2.0. An updated statement of goals for engineering education. In *Proceedings of 7th International CDIO Conference*. Denmark. Retrieved from www.springerlink.com/index/10.1007/s10956-008-9088-4

Cruz-Lemus, J. A., Genero, M., Manso, M. E., Morasca, S., & Piattini, M. (2009). Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies. *Empirical Software Engineering, 14*(6), 685–719.

Cruz-Lemus, J. A., Maes, A., Genero, M., Poels, G., & Piattini, M. (2010). The impact of structural complexity on the understandability of UML statechart diagrams. *Information Sciences, 180*, 2209–2220.

Dori, D. (2002a). *Object-Process Methodology – A Holistic Systems Paradigm*. Berlin: Springer Verlag.

Dori, D. (2002b). Why significant UML change is unlikely. *Communications of the ACM*, *45*, 82–85.

Dori, Y. J., & Belcher, J. (2005). Learning electromagnetism with visualizations and active learning. In J. K. Gilbert (Ed.), *Visualization in science education* (Vol. 1, pp. 187–216). Netherlands: Springer.

Hsieh, C., & Knight, L. (2008). Problem-Based Learning for Engineering Students: An Evidence-Based Comparative Study. *The Journal of Academic Librarianship*, *34*(1), 25–30. doi:10.1016/j.acalib.2007.11.007

Klein, E. S., & Merritt, E. (1994). Environmental education as a model for constructivist teaching. *The Journal of Environmental Education*, *25*, 14–21.

Krajcik, J. S., Czerniak, C., & Berger, C. (1999). *Teaching Children Science: A Project-Based Approach*. Boston: McGraw-Hil.

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*, *41*(4), 212–218. doi:10.1207/s15430421tip4104_2

Lindland, O. I., Sindre, G., & Solvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software*, *11*(2), 42–49. doi:10.1109/52.268955

Mathewson, J. H. (2005). The visual core of science: Definition and applications to education. *International Journal of Science Education*, *27*(5), 529–548.

Mayer, R. E. (2002). Cognitive theory and the design of multimedia instruction: An example of the two way street between cognition and instruction. In D. F. Halpern & M. D. Hakel (Eds.), *Applying the science of learning to university teaching and beyond* (pp. 55–72). San Francisco: Jossey-Bass.

Mohagheghi, P., & Aagedal, J. (2007). Evaluating Quality in Model-Driven Engineering. In *International Workshop on Modeling in Software Engineering (MISE'07: ICSE Workshop 2007)* (pp. 6–6). IEEE. doi:10.1109/MISE.2007.6

Nevo, D. (1995). *School-based evaluation: A dialogue for school improvement*. Oxford: Pergamon.

OMG UML. (2012). Information technology - Object Management Group Unified Modeling Language (OMG UML), Infrastructure. Retrieved from http://www.omg.org/spec/UML/ISO/19505-1/PDF

Paivio, A. (1990). *Mental representations: A dual coding approach*. New York: Oxford University Press.

Pekkarinen, T. (2012). Gender differences in education. In *Nordic Economic Policy Review Conference*. Oslo.

Peleg, M., & Dori, D. (2000). The model multiplicity problem: experimenting with real-time specification methods. *Software Engineering, IEEE Transactions*, *26*(6), 742–759.

Selic, B. (2003). The pragmatics of model-driven development. *Software, IEEE*, *20*(5), 19–25.

Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. New York: Doubleday.

Thomas, D. (2004). MDA: Revenge of the Modelers or UML Utopia? *IEEE Software*, *21*(3), 15–17.

Zoller, U. (1991). Problem-Solving and the "Problem-Solving Paradox." In S. Keiny & U. Zoller (Eds.), *Conceptual issues in Environmental Education* (pp. 71–87). New York: Peter Lang.

Zugal, S., Pinggera, J., Weber, B., Mendling, J., & Reijers, H. A. (2012). Assessing the Impact of Hierarchy on Model Understandability–A Cognitive Perspective. In J. Kienzle (Ed.), *Models in Software Engineering*. Berlin Heidelberg: Springer. doi:10.1007/978-3-642-29645-1_14