A Comparative Study of Languages for Model-Based Systems-of-Systems Engineering (MBSSE)

Dov Dori[1,2], Niva Wengrowicz[1,2], and Yehudit Judy Dori[1,2]
[1]Technion, Israel Institute of Technology Haifa, Israel
[2]Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract—Teaching model-based systems engineering has become more challenging as research and development has shifted from systems to include systems-of-systems (SoSs). SoSs involve primarily combination of hardware and software, as well as humans and organizations. The need to teach model-based system-of-systems engineering has thus raised the need to determine what the most suitable language for modeling and teaching SoSs is. We developed a new course format, in which groups jointly reverse-engineer and model Web-based information systems in two different modeling languages, UML and OPM, and then individually assess their peers' projects. The goal of this study was to compare UML with OPM as two candidates. We developed and evaluated an online peer assessment tool which students used. About 130 undergraduate students in groups of six divided into teams of three modeled 23 systems in both UML and OPM. They then evaluated, compared, and ranked the clarity & understandability and the completeness of the four models of two systems that their peers had constructed. Findings demonstrate that neither the order of the models assessment nor the assessor gender affected the grading, indicating assessment reliability. We found significant differences in model clarity and understandability in favor of OPM and no differences in completeness between OPM and UML models. These findings validate our approach of teaching both conceptual modeling languages and using peer assessment in large-scale project-based undergraduate information systems engineering courses.

Keywords— Unified Modeling Langugae (UML), Object-Process Methodology (OPM), model-based systems enginering (MBSE), model-based systems-of-systems enginering (MBSSE), conceive-design-integrate-operate (CDIO), large scale assessment, project-based learning (PBL), systems engineering education

I. INTRODUCTION

Modern engineering education programs aim to enrich students with the necessary knowledge, skills, and attitudes for becoming successful young engineers [1]. 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 [2]. Most of the CDIO curriculum features are related to experiential learning [2]. 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 [1].

Teaching systems engineering has become more challenging as research and development has shifted from systems to include systems-of-systems (SoSs). Integrating disparate systems to make them an interoperable formation is more challenging than the integration of several products or subsystems to a functioning system. Each system engineer takes care of his or her own system's functionality, but one must define and control the integration requirements and specifications, and engineer the services through which interaction is enabled [29].

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 [3]. PBL is characterized by authentic investigation, collaboration among peers, the use of technology to support inquiry processes, and delivery of an end product [4], [5]. 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 [6]. Researchers [7]–[9] 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 [10] 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 [11] and OPM [12].

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 [13], as it provides means for coordination and caters to common understanding among colleagues and customers. Such common understanding is even more critical in the case of model-based systems-of-systems engineering (MBSSE), as each system is likely to have a different way of being documented, expressed, and modeled.

Evaluating the quality of a conceptual model is a major issue professionals in the field of system engineering tackle [14]. 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 [14]–[16]. Our evaluation instruction to the students, embedded in our specially-designed tool, included three criteria in addition to the model correctness: model completeness, model clarity and understandability and documentation. Completeness of conceptual model means that the model contains all the requirements included in the scope [16] and all of the system main aspects, i.e. functional, structural and behavioral aspects are shown compatible. Model clarity and understandability are key quality characteristics of conceptual models [14], [17]. Understandability, i.e., a model's ability to be easily understood, is a model property that has been investigated quite intensely [18].Documentation focuses on the contribution of available system documentation to the understanding of the considerations that guided the construction of the model and on the documentation appropriateness [15].

The objective of our study was to determine what language or combination of languages is most suitable to teaching MBSE with an eye towards the emerging challenges of MBSSE. Our study focused on comparing two measures: (1) model clarity and understandability and (2) completeness of a given conceptual system model expressed in two different modeling languages: Unified Modeling Language (UML) and Object-Process Methodology (OPM). We teach UML rather than SysML in this course s we wish to balance modeling systems in general and modeling information systems in particular, for which UML is especially suited. 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.

UML – the Unified Modeling Language, developed by Object-Management Group [11], is the current de facto software modeling language. Developed by Rumbaugh, Booch, and Jacobson in 1996 as a non-propriety modeling language [19], [12]. UML currently consists of fourteen diagram types – seven structural and seven behavioral. Researches [21], [22], 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 [18] 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 (OPM) [12] 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 [23]. 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, defined as a collection of information on students' outcomes [24], is commonly applied to evaluate students' thinking skills. Bloom's taxonomy and its revisions [25] 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 [26]. 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.

A central theme in the peer assessment we have used in the course is the *wisdom of the crowd* theory [27], which 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.

II. RESEARCH GAOL AND METHOD

The overall research concerns developing and assessing a new approach and adequate languages for teaching and assessing undergraduate students in large engineering courses focusing on MBSE and MBSSE that use a PBL approach. In this paper we focus on the quantitative aspect of model clarity and understandability (MCU) and completeness (COMP) of the two evaluated languages, UML and OPM. Accordingly, the main goal of the study described in this paper was to develop and validate an approach and a supporting Web-based tool for students' peer assessment of MCU and COM of UML and OPM conceptual models. These models were developed by teams of students in a large PBL information system engineering course.

A. 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. 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. The research was conducted during the winter 2012-3 Semester. Almost exactly half of the 130 students who took the course were females. Employing a project-based learning (PBL) approach, we presented students with realistic problems that have no single correct answer. Students were asked to "reverse engineer," specify, and model a given complex Web-based system, such as Gmail, Expedia, or eBay.

Groups of students constructed models of these Web-based systems using two conceptual modeling approaches and languages: OPM and UML. OPM and UML 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 part 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. In the middle of the semester, the teams swapped, and each team continue working on the model that the other team in the group had started: The group who started to create the OPM model continued developing the UML model that their peers started and vice versa. This way, at the end of the semester, each group had two system model: one expressed in OPM and the other in UML, both constructed by all the six team members. Each student therefore had the opportunity to practice modeling in both OPM and UML.

After submitting the final project, each student was asked to assess individually OPM and UML models of two different systems created by other groups. Thus, each student had to assess in total four models of two systems: two in UML and two in OPM. To make the assessment effective and collect the large amounts of data efficiently, we developed a dedicated Web-based assessment tool (see Figure 1).

Students had to assess the four models individually based on a list of criteria, two of which were MCU and COMP, the focus of this research. 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 randomly tasked with assessing a unique couple of projects, and a 24 hour time window was allotted for this assessment. The assessment related to a list of criteria that had been exposed to the students beforehand and discussed extensively during the course.

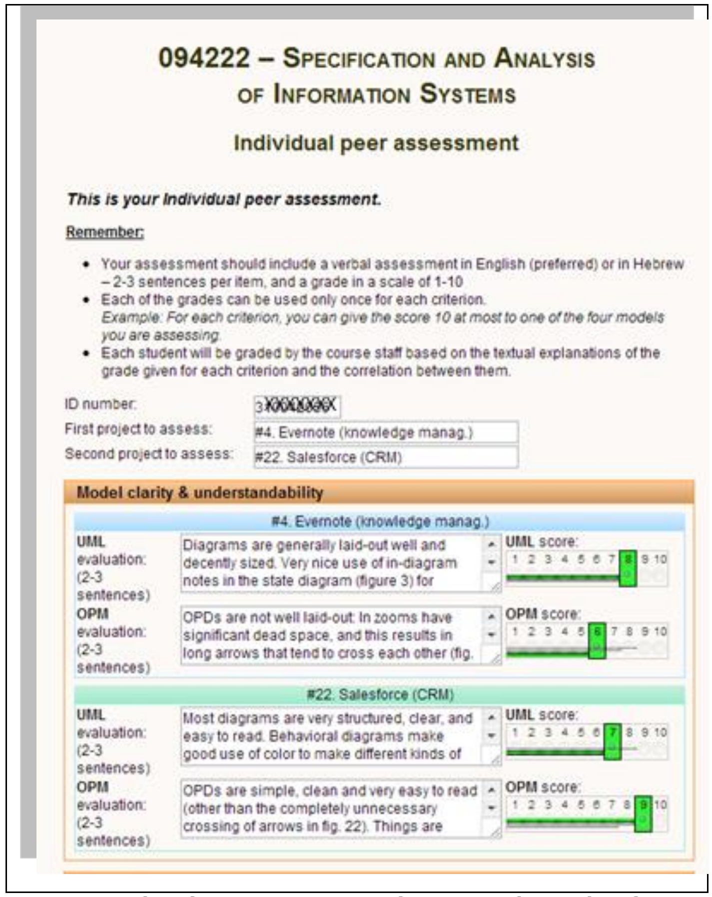

Figure 1. The online peer-assessment tool screen snapshot: Each student scored four models: two in OPM and two in UML, of two systems designed by peers The assessing stdeunt could give only one score value to each one of the four models

The online assessment included a written assessment and a corresponding grade in a scale of 1-10. To prevent students from giving the highest grade (10) to all the four models they had to assess, the tool does not allow giving any grade to more than one of the four models. Therefore, no grade could be given for each criterion in each checked model more than once (see Figure 1). For example, in Figure 1, the student (the assessor) gave a grade of 8 in the "model clarity & understandability" criterion to the Evernote UML model, and a grade of 6 to the Evernote OPM model, the Salesforce UML model received grade 7, and the Salesforce OPM model received grade 9. Thus, the highest set of grades a student could give to the four models was 10, 9, 8, and 7. The grade that each team member received for being part of the team that developed the UML and OPM models was the average of the 12 individual grades on the 1-10 scale given by peers.

B. Research objectives and hypotheses

In order to carry out the validation of the approach and the tool, and to gain insight into the benefits and drawbacks of each modeling language, we set up the following three research objectives.

1. Determine whether there was any order effect between the first and second evaluated projects (systems) on the MCU and COMP grades. As explained in the sequel, each student was assigned with checking two different projects. For each project, the students were asked to assess an OPM model and a UML model. For example, in Figure 1, the "first project" was Evernote and the "second project" was Salesforce. The motivation here was to rule out effects of fatigue, experience, or other unexpected biases do to the order of project evaluation. We assumed that if the tool is reliable, there would not be significant differences in MCU and COMP between the model evaluated first and second.
2. Examine, for each system and overall, whether there were differences between the grades for MCU and COMP of the OPM model and the UML model of the same system.
3. Determine whether there was any gender effect in the grades assessors gave for model clarity and understandability.

Our first research hypothesis, pertaining to Objective 1, was that there would not be significant differences in model clarity and understandability (MCU) nor completeness (COMP) between the system evaluated first and second.

Our second research hypothesis, pertaining to Objective 2, was that we would not find significant differences in MCU and COMP grades between the OPM and the UML models of the same system. Models' COMP is related to the specification and analysis work that the group did as a whole regardless of the language they use to represent the system in each team. As for MCU, while OPM puts all the system aspects in a single diagram type at varying detail levels, integrating function, structure, and behavior, UML breaks the model into up to 14 diagram types, leaving the mental integration of the various aspects to the model reader. On the other hand, each aspect is separate may clarify individual aspects better. Indeed, we found such a trend while comparing OPM with SysML [30].

Our third research hypothesis, pertaining to Objective 3, was that there may be a significant gender difference between grades given by the students in respect to MCU and COMP. Based on previous studies [28] although females and males studied and designed the 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 grading.

III. DATA ANALISYS AND RESULTS

In order to test our three research hypotheses, three-way repeated measures ANOVA was conducted twice. In the first time, the dependent variable was MCU (*MCU-grade*) and in the second, the dependent variable was COMP (*COMP-grade*). The independent variables in both were (1) *system order* (within-subject variable), whose values were *first* and *second*, (2) *model type* (within-subject variable), whose values were *OPM* and *UML*, and (3) *gender* (between-subject variable), whose values were *female* and *male*. Table I presents the main effect results of the repeated measures ANOVA for the MCU dependent variable. Table II presents the main effect results of the repeated measures ANOVA for the COMP dependent variable.

The findings for the first hypothesis were the following. (1) There was no significant main effect of system checking order on *MCU-grade* $F(1,128)=.077$, $p>.05$, $\eta_p^2=.00$. In other words, no significant difference was found between the *MCU-grade* of the first assessed system OPM and UML models (M=8.24, SD=.07) and the second assessed system models (M=8.27, SD=.09; see Figure 2). (2) There was no significant main effect of system checking order on *COMP-grade* $F(1,128)=2.83$, $p>.05$, $\eta_p^2=.02$. In other words, no significant difference was found between the *COMP-grade* of the first assessed system OPM and UML models (M=8.34, SD=.09) and the second assessed system models (M=8.12, SD=.08).

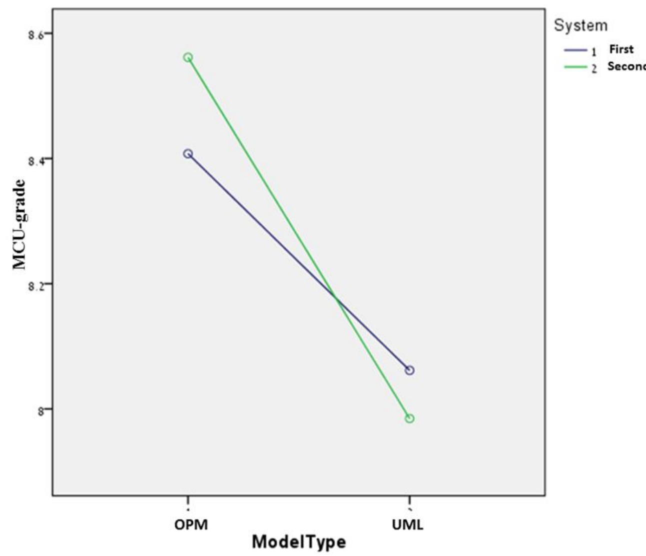

Figure 2. MCU by grading order: significant main effect of model type, (OPM vs. UML), but no significant main effect of system order and no significant interaction

The findings for the second hypothesis were the following. (1) There was a significant main effect of model type $F(1,128)=14.40$, $p<.001$, $\eta_p^2=.10$ (see Figure 2 and Figure 3). In other words, the *MCU-grade* of the OPM model (M=8.49, SD=.08) was significantly higher than the MCU-grade of the UML model (M=8.02, SD=.07). (2) There was no significant main effect of model type on *COMP-grade* $F(1,128)=.26$, $p>.05$, $\eta_p^2=.00$. In other words, no significant difference was found between the *COMP-grade* of the OPM model (M=8.26, SD=.09) and the UML model (M=8.20, SD=.07) of the same system.

TABLE I. *MCU-grade* means, standard deviation, and the repeated measures ANOVA main effect results

Main Effect		Mean (SD)	F(1,128)	p
System checking order	First	8.24 (.07)	.077	n.s.
	Second	8.27(.09)		
Model type	OPM	8.49 (.08)	14.40	p<.001
	UML	8.02 (.07)		

Main Effect		Mean (SD)	F(1,128)	p
Gender	Female	8.33 (.07)	2.75	n.s.
	Male	8.18 (.07)		

TABLE II. COMP-grade means, standard deviation, and the repeated measures ANOVA main effect results

Main Effect		Mean (SD)	F(1,128)	p
System checking order	First	8.34 (.09)	2.83	n.s.
	Second	8.12 (.08)		
Model type	OPM	8.26 (.08)	.26	n.s.
	UML	8.20 (.07)		
Gender	Female	8.26 (.06)	.58	n.s.
	Male	8.20 (.06)		

Table III. MCU-grades means, standard deviations and the three-way repeated measures ANOVA interaction result

Main Effect			Mean (SD)	F(1,128) p
Female	First System	OPM	8.55 (1.28)	.42 n.s.
		UML	8.06 (1.18)	
	Second System	OPM	8.78 (1.05)	
		UML	7.92 (1.23)	
Male	First System	OPM	8.26 (1.24)	
		UML	8.06 (1.24)	
	Second System	OPM	8.34 (1.75)	
		UML	8.05 (1.50)	

Finally, findings for the third hypothesis indicated that (1) there was no significant main effect of gender F(1,128)=2.75, p>.05, η_p^2=.02. There was no significant difference between the *MCU-grade* given by females (M=8.33, SD=.07) to those given by males (M=8.18, SD=.07; see Figure 3).

No significant interaction was found between *system* and *model type* F(1,128)=1.16, p>.05, η_p^2=.01 (see Figure 2). Likewise, no significant interaction was found between *gender* and *model type* F(1,128)=3.13, p>.05, η_p^2=.02. However this statistical difference in the interaction was borderline p=.08, which may represent a weak interaction (as can be seen in Figure 3) and should be studied further. We found no significant interaction between all the three independent variables—*gender*, *system*, and *model type* F(1,128)=.42, p>.05, η_p^2=.00. Table III presents the *MCU-grades* means, standard deviations and the three-way repeated measures ANOVA interaction result.

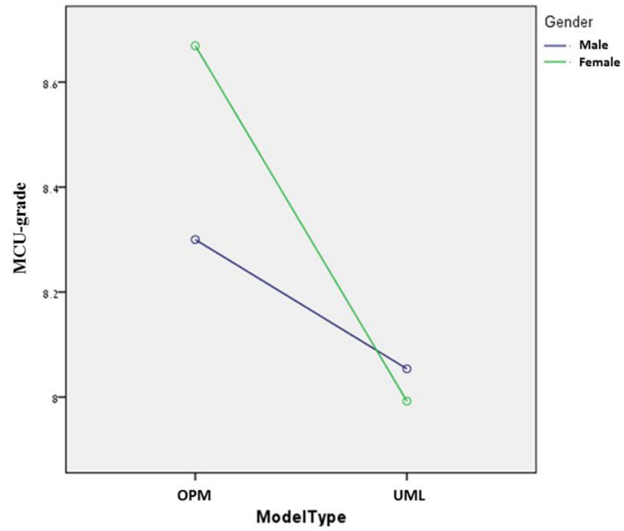

Figure 3. MCU by gender: significant main effect of model type, (OPM vs. UML), but no significant main effect of gneder and no significant interaction

TABLE VI. *COMP-grade* means, standard deviation and the three-way repeated measures ANOVA **interaction** results

Main Effect			Mean (SD)	F(1,128) p
Female	First System	OPM	8.62 (.15)	.14 n.s.
		UML	8.25 (.15)	
	Second System	OPM	7.91 (.18)	
		UML	8.28 (.14)	
Male	First System	OPM	8.46 (.15)	
		UML	8.05 (.15)	
	Second System	OPM	8.06 (.18)	
		UML	8.23 (.14)	

With respect to model comprehension (COMP) we got the following results. (1) There was no significant main effect of gender on *COMP-grade* F(1,128)=.58, p>.05, η_p^2=.00. In other words, no significant difference was found between the *COMP-grade* given by female (M=8.26, SD=.06) to those given by males (M=8.20, SD=.06). (2) No significant interaction was found between the pairs and all the three independent variables-*gender, system,* and *model type* F(1,128)=.14, p>.05, η_p^2=.00. Table IV presents the *COMP-grades* means, standard deviations and the three-way repeated measures ANOVA interaction result.

IV. DISCUSSION AND CONCLUSIONS

The two goals of the study described in this paper on large scale undergraduate information systems engineering course were to (1) develop and evaluate a new assessment approach and a supporting tool for large information system engineering project-based learning (PBL) courses, and (2) evaluate the clarity and understandability and of models of systems expressed in UML and OPM as candidate languages to be taught in MBSE and MBSSE undergraduate course. For goal (1), we validated the model clarity and understandability and the completeness criteria used in our designated online assessment tool. To ensure grade distribution, our online peer-assessment tool includes a written assessment and an adequate grading scheme on a scale of 1-10, where no grade is allowed to repeat more than once. As we hypothesized, there was no significant difference in the assessments of model clarity and understandability and completeness between the first and the second system that each student was assigned to assess. No difference between females and males' assessment outcomes was found either. For goal (2) we found no significant deference in the assessment of model completeness between OPM and UML. However, we did was a significant difference in the model clarity and understandability grades between the OPM and UML models within each project in favor of OPM. At least on average, the OPM model in each project received a higher grade compared with its UML counterpart. This finding reinforces previous experimental results, indicating that an OPM model of a complex information system provides superior visual specification and comprehension capabilities than the corresponding UML model of the same system.

However, due to the fact that both UML and OPM turned out to be statistically equal in terms of model completeness and due to the prevalence of UML as the de-fact standard software engineering modeling language, in future courses we will continue teaching both UML and OPM as complementary languages for MBSE and MBSSE. This decision is in line with previous finding of comparing OPM to SysML [30] that the addition of auto-generated SysML views to an OPM system model increases system comprehension. Future research will continue to examine the large amount of data with respect to the tool's validation while we continue to use and improve our PBL approach for teaching information system engineering courses.

ACKNOWLEDGMENT

This work was funded by the EU FP7 Project #262044-VISIONAIR—Vision Advanced Infrastructure for Research and by the Technion Post-Doctoral Fellowships.

REFERENCES

[1] E. F. Crawley, J. Malmqvist, W. A. Lucas, and D. R. Brodeur, "The CDIO syllabus v2.0. An updated statement of goals for engineering education," in *Proceedings of 7th International CDIO Conference*, 2011.

[2] E. F. Crawley, D. R. Brodeur, and D. H. Soderholm, "The education of future aeronautical engineers: conceiving, designing, implementing and operating," *J. Sci. Educ. Technol.*, vol. 17, pp. 138–151, 2008.

[3] C. Hsieh and L. Knight, "Problem-Based Learning for Engineering Students: An Evidence-Based Comparative Study," *J. Acad. Librariansh.*, vol. 34, no. 1, pp. 25–30, Jan. 2008.

[4] J. S. Krajcik, C. Czerniak, and C. Berger, *Teaching Children Science: A Project-Based Approach*. Boston: McGraw-Hil, 1999.

[5] E. S. Klein and E. Merritt, "Environmental Education as a Model for Constructivist Teaching," *J. Environ. Educ.*, vol. 25, no. 3, pp. 14–21, Apr. 1994.

[6] U. Zoller, "Problem-Solving and the 'Problem-Solving Paradox'," in in *Conceptual issues in Environmental Education*, S. Keiny and U. Zoller, Eds. New York: Peter Lang, 1991, pp. 71–87.[7] J. H. Mathewson, "The visual core of science: Definition and applications to education," *Int. J. Sci. Educ.*, vol. 27, no. 5, pp. 529–548, 2005.

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

[9] A. Paivio, *Mental representations: A dual coding approach*, 2nd ed. NY: Oxford University Press, 1990.

[10] Y. Dori and J. Belcher, "Learning Electromagnetism with Visualizations and Active Learning," in in *Visualization in science education*, vol. 1, J. K. Gilbert, Ed. Berlin/Heidelberg: Springer-Verlag, 2005, pp. 187–216.

[11] OMG UML, "Information technology - Object Management Group Unified Modeling Language (OMG UML), Infrastructure," 2012. [Online]. Available: http://www.omg.org/spec/UML/ISO/19505-1/PDF.

[12] D. Dori, *Object-Process Methodology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[13] D. Thomas, "MDA: Revenge of the Modelers or UML Utopia?," *IEEE Softw.*, vol. 21, no. 3, pp. 15–17, 2004.

[14] J. Akoka, I. Comyn-Wattiau, and S. S.-S. Cherfi, "Quality of conceptual schemas an experimental comparison," *2008 Second Int. Conf. Res. Challenges Inf. Sci.*, pp. 197–208, Jun. 2008.

[15] P. Mohagheghi and J. Aagedal, "Evaluating Quality in Model-Driven Engineering," in *International Workshop on Modeling in Software Engineering (MISE'07: ICSE Workshop 2007)*, 2007, pp. 6–6.

[16] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *IEEE Softw.*, vol. 11, no. 2, pp. 42–49, Mar. 1994.

[17] B. Selic, "The pragmatics of model-driven development," *Software, IEEE*, vol. 20, no. 5, pp. 19–25, 2003.

[18] J. A. Cruz-Lemus, M. Genero, M. E. Manso, S. Morasca, and M. Piattini, "Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies," *Empir. Softw. Eng.*, vol. 14, no. 6, pp. 685–719, 2009.

[19] S. Covert, "OMG's Unified Modeling Language (UML) Celebrates 15th Anniversary," 2012. [Online]. Available: http://www.omg.org/news/releases/pr2012/08-01-12-a.htm.

[20] D. Dori, "Why significant UML change is unlikely," *Commun. ACM*, vol. 45, pp. 82–85, 2002.

[21] J. A. Cruz-Lemus, A. Maes, M. Genero, G. Poels, and M. Piattini, "The impact of structural complexity on the understandability of UML statechart diagrams," *Inf. Sci. (Ny).*, vol. 180, pp. 2209–2220, 2010.

[22] S. Zugal, J. Pinggera, B. Weber, J. Mendling, and H. A. Reijers, "Assessing the Impact of Hierarchy on Model Understandability–A Cognitive Perspective," in in *Models in Software Engineering*, J. Kienzle, Ed. Berlin Heidelberg: Springer, 2012.

[23] M. Peleg and D. Dori, "The model multiplicity problem: experimenting with real-time specification methods," *Softw. Eng. IEEE Trans.*, vol. 26, no. 6, pp. 742–759, 2000.

[24] D. Nevo, *School-based evaluation: A dialogue for school improvement*. Oxford: Pergamon, 1995, p. 213.

[25] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *Theory Pract.*, vol. 41, no. 4, pp. 212–218, Nov. 2002.

[26] S. Bedford and S. Legg, "Formative peer and self feedback as a catalyst for change within science teaching," *Chem. Educ. Res. Pract.*, vol. 8, no. 1, pp. 80–92, 2007.

[27] J. Surowiecki, *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, 2004.

[28] T. Pekkarinen, "Gender differences in education," in *Nordic Economic Policy Review Conference*, 2012.

[29] Sage, Andrew P., and Charles L. Lynch. 1998. "Systems Integration and Architecting: An Overview of Principles, Practices, and Perspectives." Systems Engineering 1 (3): 176–227.

[30] Yariv Grobshtein and Dov Dori, Generating SysML Views from an OPM Model: Design and Evaluation. Systems Engineering, 14 (3), pp. 327-340, 2011.