Pair modeling with DynaLearn - Students' attitudes and actual effects

Or-Bach, R.; Bredeweg, B.

DOI
10.28945/1371

Publication date
2011

Document Version
Final published version

Published in
Interdisciplinary Journal of Information, Knowledge, and Management

Pair Modeling with DynaLearn – Students' Attitudes and Actual Effects

Rachel Or-Bach
Academic College of Emek Yezreel, Israel

orbach@yvc.ac.il

Bert Bredeweg
Informatics Institute, University of Amsterdam, the Netherlands

B.Bredeweg@uva.nl

Abstract

With DynaLearn learners can construct scientific knowledge by manipulating icons and their inter-relationships, using a diagrammatic representation. The diagrams represent models that can be simulated, confronting learners with the logical consequences of the knowledge they expressed. Such modeling activities are highly advocated by science educators. Learning from the construction and debugging processes of modeling can be enhanced by collaboration. The modeling elements can serve as anchors for discussing, justifying, and explaining the model. Researchers have suggested various ways of supporting collaboration. In this study we employed Pair Modeling, which is an adaptation of the pair programming technique that is used for enhancing collaborative programming both in the industry and in academia. In this paper we present encouraging results for the use of this collaboration technique based on assignments' scores, observations, and a questionnaire. Students' attitudes were neutral on the average, but the average score of the group that employed Pair Modeling was significantly higher than the average score of the control group that employed unstructured pair collaboration. We discuss the implications of the obtained results and the limitations of the study.

Keywords: Modeling, pair modeling, collaborative modeling, collaboration, science education

Introduction

Modeling is regarded fundamental to human cognition and scientific inquiry and therefore science educators strongly advocate the importance of scientific modeling within science education. A scientific model is an abstract, simplified, representation of a system or of a phenomenon that makes its central features explicit and that can be used to generate explanations as well as predictions. Involving learners in modeling practices, and especially having students develop models to articulate their own understanding, can help them build subject matter knowledge, epistemological understanding, and expertise in the practices of building and evaluating scientific knowledge (Lesh & Doerr, 2000; Penner, 2001; Schwarz & White, 2005; Schwarz et al., 2009; Wilensky & Reisman, 2006; Windschitl, Thompson, & Braaten, 2008). A model can have a variety of representations. Visual models have the advantage of providing concrete reification of the respective concepts. This motivated the design of sev-

Editor: Eli Cohen

eral modeling environments for science education (Frederiksen & White, 2002; Jackson, Krajcik, & Soloway, 1998).

The DynaLearn modeling environment (http://www.DynaLearn.eu) differs from other educational model building environments by the type of simulation engine that enables the evaluation of the generated models. The simulation engine is built on the Qualitative Reasoning (QR) approach (Bredeweg, Linnebank, Bouwer, & Liem, 2009). Qualitative reasoning is a formal approach to knowledge capture originating from artificial intelligence that can be used to express, analyze, and communicate conceptual models (Bredeweg & Salles, 2009). Qualitative models excel when theoretical background on the target system is weak, when the problems are ill-defined, and when data are incomplete, uncertain, or simply not available. Conceptual models based on qualitative reasoning are valuable tools both for pre-mathematical modeling and as standalone artifacts developed for understanding, predicting, and explaining the system's behavior. Such conceptual models are also 'animated', as they capture the dynamic aspects of a system by reasoning about quantities that change over time, using a well-defined set of modeling primitives grounded in a mathematical foundation.

The DynaLearn modeling environment (Bredeweg, Liem, Beek, Salles, & Linnebank, 2010) includes icon-based modeling primitives, such as entities, quantities, and causal dependencies, which are combined into model fragments and scenarios. Given a library of model fragments and a scenario describing an initial situation, the qualitative simulation engine generates predictions in the form of state-transition graph.

Collaboration within scientific domains is advocated for Mathematics, Engineering, and Technology learning (Springer, Stanne, & Donovan, 1999) and for discovery in scientific domains (Okada & Simon, 1997). Learning from the construction and debugging processes of modeling can be enhanced by collaboration: by having learners compare, reflect, and discuss their constructed models and the relations between the models and the real (or simulated) phenomenon. The modeling elements can serve as anchors for discussing, justifying and explaining the model. Dillenbourg (1999) stressed the need for supporting collaborative learning and suggests four support dimensions: a) Set up initial conditions, such as group size and tasks nature; b) Use of clear specifications of roles or matching of learners, for example by complementarity's of their skills or knowledge (Hoppe & Ploetzner, 1999); c) Scaffold productive interactions by semi-structured interfaces, using special buttons, menus, or sentence-openers; d) Monitor and regulate the interactions, by some diagnosis and intervention, or provision of some mirroring tools to enable self-regulation. In this paper we suggest the Pair Modeling approach for dealing with the two first dimensions. Results from our study provide directions for further research and development that is related to the other two dimensions.

Modeling is a complex cognitive activity that involves working simultaneously in different problem spaces, using formal notation to represent abstract notions and performing iterations of construction, evaluation, and debugging. Modeling activities within a formal framework that enables simulation (execution of the model) resemble programming activities, and programming can be viewed as modeling where the modeling tool is a programming language. Both modeling and programming involve the use of a formal language to represent real world situations and processes. The complexity of programming as explained by various researchers (Blackwell, 2002; Brooks, 1983; Penington, Lee, & Rehder, 1995) seems very relevant also for modeling. The main challenge is working with a number of different levels of abstraction and coordinating between these levels and respective knowledge spaces. Because of the great similarity between modeling as applied in DynaLearn and programming, it makes sense to look at techniques that are employed for supporting collaborative programming. Pair programming is such a technique, and it is used both in the industry and in introductory computer science courses where programming is taught. It is a collaboration technique that defines the size of the collaborating team, two roles, and the requirement to switch roles. The Pair Modeling technique that we introduced in our study is an adaptation of the pair programming technique. It is worthwhile to investigate ways for supporting effective collaborative modeling as in many situations even when collaboration is not the declared pedagogy, students work collaboratively anyway because of limited resources of the educational system or because of students' own choice.

The following section presents a review of research regarding pair programming as a context and motivation for the Pair Modeling technique applied in this study. Next we explain the study setting, design and tools followed by the results section. The last section presents conclusions along with implication of the study results.

Pair Programming and Pair Modeling

Pair Modeling (PM) is a technique that we introduced for supporting collaborative modeling with DynaLearn. This technique is an adaptation of the pair programming collaboration technique which is used in software development projects as well as in learning programming in introductory Computer Science (CS) courses in higher education institutions. Pair programming has been formalized as one of the primary practices of the Extreme Programming (XP) approach to software development (Beck & Andres, 2004). Williams, Kessler, Cunningham, and Jeffries (2000) defined pair programming as a practice in which two programmers sitting side-by-side using only one computer to work collaboratively on the design, algorithm, code, or test. The pair consists of two developers who change their role alternately as the "driver" and "navigator". The "driver" is responsible for typing the code and has control over the resources such as computer, mouse, and keyboard, whereas the "navigator" or "reviewer" has the responsibility of observing the driver's work. The idea being captured by these metaphors is that the driver is the programmer at the keyboard focusing on the task of typing in the code, while the navigator is less occupied with the immediacy of code production and can concentrate on the overall direction of the development of the program (Bryant, Romero, & du Boulay, 2008).

A wide range of studies investigated the benefits of pair programming in terms of its effect on the quality of the resulting software. These studies have taken place in both commercial environments (Jensen, 2003; Nosek, 1998) and academic environments (Brereton, Turner, & Kaur, 2009; Nagappan et al., 2003; Williams, Wiebe, Yang, Ferzli, & Miller, 2002). For our study, the research in academic environments is more relevant, so the following literature review relates mainly to studies on the use of pair programming in CS1, an introductory programming course for computer science students. Results from an empirical research in a CS1 course indicate that students who practice pair programming perform better on programming projects and are more likely to succeed by completing the class with a C or better. Student pairs are more self-sufficient which reduces their reliance on the teaching staff (Williams et al., 2002). Qualitatively, paired students demonstrate higher order thinking skills than students who work alone (Williams et al., 2002). Nagappan et al. (2003) reported an improvement in pairs' grades on exams over students who programmed individually. Brereton, Turner, and Kaur (2009) report the results of a systematic literature review of ten empirical studies. They conclude that pair programming may improve undergraduate students' pass and retention rates on programming modules and is likely to improve their confidence in their work and attitude towards programming. Cliburn (2003) presents survey results that showed that a majority of the students liked working in pairs on the projects, most students thought pair programming improved their grade, and an overwhelming majority of students thought that pair programming made them better at working with others.

Various methods have been implemented for forming the groups. As Cliburn (2003) highlights in his paper, McDowell et al. (2002) allowed students to choose their own partners. Nagappan et al. (2003) used a software program to make random partner assignments. Thomas, Ratcliffe, and Robertson (2003) had students rate themselves on a scale of 1 to 9 in terms of their programming ability and then assigned partners based on these ratings. In some projects, students who rated themselves highly were paired with students who rated themselves poorly. In other projects, students were paired with those who rated themselves similarly. Williams and Kessler (2003) also discussed forming pairs based on personalities (whether someone is extrovert or introvert). Sfetsos, Stamelos, Angelis, and Deligiannis (2009) have shown better performance and collaboration-viability for pairs with heterogeneous personalities and temperaments. On the other hand, Salleh, Mendes, Grundy, and Burch (2009) showed that differences in personality traits did not significantly affect the academic performance of students who pair programmed. In many studies (Nagappan et al., 2003; Thomas et al., 2003; Williams et al., 2002) programmers were assigned new partners throughout the semester.

The term Pair Modeling was suggested by Kamthan (2008) as an adaptation of the pair programming technique for software modeling. Kamthan regards programming as involving only implementation issues and therefore addresses explicitly software modeling difficulties. He stresses the need for collaborative modeling these days when modeling has become complex enough for an individual, requiring intimate knowledge of both the domain being modeled and the knowledge and skills in using a high-level modeling language. This is the complexity we are trying to help students deal with by providing visual collaboration anchors and structure for the collaboration.

The Study Setting

The Conceptual Modeling Course and the DynaLearn Modeling Environment

The Conceptual Modeling course is a mandatory course within the Future Planet Studies program in the University of Amsterdam (http://www.studeren.uva.nl/fps). This internationally oriented programme deals with current challenges posed by the relationship between humans and nature, such as climate change, the demand for natural resources, sustainability, food and drinking water, and water management. The course Conceptual Modeling is concerned with the use of computer models and simulations as a means for describing, analyzing, and explaining systems and their behavior, with an emphasis on non-numerical methods, and uses the notion of processes (and associated cause-effect relationships) and the idea of system's theory and principles. The models have a logical basis and as such contribute to the systematic processing of knowledge and formalization. Students are expected to develop interdisciplinary thinking and learning to see and make use of analogies between systems from different disciplines.

This year's (2010) version of the course took 3 EC (European credit transfer system) and ran for 8 weeks. Students were expected to work 10 hours per week on the course. Each week the theoretical lesson (2 hours) was accompanied by a 3 hours practical class in a computer laboratory using the DynaLearn software. During these hours students were required to work on a set of DynaLearn modeling assignments related to the theoretical material of that week. Students were required to submit these assignments that were graded as part of the course final grade. For the laboratory session the students were divided into two groups, each in a separate classroom, in order to enable proper help for the students working with the DynaLearn software. During the first five weeks, students were required to work individually on the laboratory assignments. Starting from the 6th week students were required to work collaboratively. The assumption was that at this stage, when students were more fluent with the software, collaboration could be beneficial for their learning.

Appendix A presents an example of a modeling exercise that students had to work on and submit during lab hours (classwork). The exercise in Appendix A is the third one of the 6th week assignment (named 7c assignment).

Design and Tools for the Study

The main goal of the study was to investigate whether the Pair Modeling (PM) technique is beneficial for collaborative modeling with DynaLearn. An additional goal was to find students' attitudes towards collaboration and especially towards PM. The study involved 56 students taking the Conceptual Modeling course. The age of these students was around 20 years old: 31 males and 25 females. The Pair Modeling technique described in the previous chapter was employed during two of the laboratory sessions (in consecutive weeks), when students had to submit the 6th week and the 7th week classwork assignments working collaboratively in pairs. In order to study the effect of the PM technique one of the groups employed PM for the first (6th) week and the other during the second week, enabling the existence of a control group during the first week with exactly the same assignment and exactly the same course materials and exactly the same previous assignments. Students in the control group worked also in pairs but with no predefined structure for the collaboration. Students were introduced to the PM idea in the beginning of the respective session and received a short written description of the technique, the two roles, the expectance that the partners communicate often, and the importance of switching roles. They were also told who should start with which role and the frequency of switching roles. During the 3 hours of the laboratory session each student had a chance to perform the same role twice. We wanted students to experience one role but also be able to experience the other role, and then get back to perform the first role with a better understanding of the other partner role. Students were reminded each time a switch of roles was required.

One of the authors was observing the two groups during these two weeks (the labs were in adjacent rooms), with more attention to the group using the PM technique that week.

At the end of the second session (second week) a questionnaire was administered to both groups (Appendix B). The goal of the questionnaire was to investigate students' attitudes towards collaboration in general, towards PM in particular, and also to get from the students their own suggestions for effective collaboration with DynaLearn. The questionnaire included 20 items. Most of the items were of Likert type employing a 5 point scale. The neutral point was intentionally included in the scale to allow the neutral attitude. The questionnaire included also three questions where students were expected to use free text to write about their experience, views, and suggestions.

To summarize the research design and tools, we had a controlled experiment during the first week, with an experimental group of 28 students (14 pairs) using PM and a control group also with 28 students employing unstructured collaboration in pairs. All pairs were required to submit the 6th week assignment as part of their final grade for the course. Observations were conducted with both groups during this week. The questionnaire was administered after the second week to the two groups in order to have all the students' reactions to PM based on both a session with PM and a session with unstructured collaboration. The second week was not considered a controlled experiment because for the 7th week assignment students were not graded for the final score and also we assumed that there might be some contamination of the experiment by the fact that the experimental group of the first week might use some structuring based on their previous experience.

Results

The results description consists of four parts relating to the research tools that were employed. First we present the analysis of the assignments' scores and then insights from the observations. The last two parts present the questionnaire's analysis, where the analysis of the free text input is described separately.

Scores

The scores for the 6th week assignment, the assignment administered during the first week of the experiment, were used to investigate the actual effect of employing the PM technique on students' achievements. The maximum total score for all the 5 modeling exercises of this assignment was 5. The experimental group (using the Pair Modeling that week) and the control group (with unstructured pair collaboration) consisted each of 14 pairs (28 students). The average score for the experimental group was 3.18 (standard deviation 0.9) with 4 pairs having a score higher than 3.5 and one pair of them had a score of 4.5. In the control group the average score was 2.57 (standard deviation 0.6) with no score higher than 3.5. The difference between the averages of the groups is statistically significant (T-test resulting in $p=0.023$). The fact that the average for the experimental group was 24% higher than the average of the control group required additional checking whether the experimental group consisted by some chance of better students with regard to DynaLearn modeling. In order to check this we compared the average of the same groups of students that participated in the study on the scores obtained in the two previous assignments of the course. These scores were on individual modeling exercises with DynaLearn. The comparison showed a very slight difference, 4% and 2% respectively, in favor of the other group, the control group. From these results it seems that the students in both groups were on about the same level and the differences in achievements that were found in the 6th week assignment may be attributed to the use of the Pair Modeling technique.

Observations

The Observations were focused on the communication dynamics and not on the content. Observations were conducted in both groups during two weeks (two sessions) for comparing the communication dynamics.

During the first week the general impression was that the group employing PM was working more seriously, concentrating more on the task on hand. Most students were keeping their roles (of "driver" and "navigator") and changing them when asked to. It was interesting to see how they talked and used the pointing devices: the driver with the mouse and the navigator with a finger or a pen. In most pairs there was a lot of communication between the driver and the navigator as expected with this technique. When the students had to switch roles some of them exchanged sitting places. This was a clear indication of who had which role. There were incidents where the navigator in order to explain an idea took the mouse and explained by respective execution. This move was probably motivated by speed and/or convenience considerations but the move misses the PM goals – to support effective collaboration. An explicit explanation of the intention is supposed to be more beneficial for the learning of both partners and for improving the communication. Requests for help from the teaching assistants were in some pairs stated and explained by the navigator and in others by the driver. In one pair personality differences were clearly affecting the collaboration; the student that started with the driver role continued to control the activity without communicating what he was doing and with very little patience for inputs from his partner. Even after switching roles this student continued to take the mouse often with minimal communication. In another pair, where there seem to be a big difference in mastery of the DynaLearn software between the two partners, one student performed the driver role through the whole session because the partner was not able to perform the required interface actions.

The group that worked with unstructured collaboration seemed more 'easy going' and less concentrated on the task. Some pairs were working together with one (or the other) using the mouse during the collaboration to point out or to perform some modeling or simulation related activity. For these pairs the mouse was between them and was used by both when appropriate for the collaborative modeling. Other pairs in this unstructured collaboration group were working differently, with one person using the computer all the time (or most of the time) and the other commenting, using pen and pencil, looking at the respective lecture notes etc.

During the second week the exercises of the 7th week assignment were different from previous assignments. During the observations it became clear that the type of tasks had an effect on the collaboration dynamics. The 7th week assignment contained two types of exercises, both dealing with specific aspects of modeling and using DynaLearn. One type was diagnosing erroneous models, submitting corrected ones, and reporting about the corrections; the other type concerned regular modeling. The students in the group that employed PM during this week seemed much less motivated to keep the roles and/or to switch roles when asked to in comparison with the group that employed the PM technique the week before. Observing these students showed that in several pairs the students switched roles for the reporting phase disregarding the driver/navigator roles at that stage. During the diagnosing phase (of erroneous models) the collaboration seemed more intense, with a lot of communication, but with many incidents where both students were using the mouse, disregarding their roles. In summary, for the 7th week assignment the two groups seemed to work more similarly to each other than in the previous week.

Questionnaire

The questionnaire (Appendix B) was filled out by 51 students, which is a major portion of the participants. Analysis of the free text inputs is described separately. Most of the questionnaire items involved a scale of 5 points: strongly disagree, disagree, neutral, agree, and strongly agree; with respective values 1-5. Three items had a dedicated five points scale according to the specific statements. Along with reporting the average score for the items we present some distribution data when appropriate in order to provide additional informative data.

Students seem to appreciate the contribution of collaboration to modeling. Aggregation of the "agree" and "strongly agree" value points on the 5 points scale for the respective item showed that 82% of the students think that constructing the models collaboratively produces better models than individual construction. The average score for this item was 4.2. But along with this appreciation, about half of the students (57%) prefer to study by themselves as opposed to studying collaboratively. Two items of the questionnaire dealt with students' self-perception of mastery with regard to the DynaLearn software and to "population behavior" which was the content topic for the modeling assignments in the 6th and 7th classwork. These items were included as informative related data under the assumption that students might look for collaboration as a source of help when their self-perception of their mastery is low. These items had a dedicated scale and the results were that students rated their mastery between average and good with a small difference in favor of mastering the content over mastering the software. These results do not indicate an extreme need for help.

We were interested in students' attitudes towards PM as a technique that contributes to collaborative modeling. The averages for the respective items range from 2.7 to 3.2., which shows that in general students were quiet neutral with regard to the contribution. A closer look at these items shows that students' attitudes differ with regard to the different aspects of possible contributions. The following tables and diagrams present this data. The tables present two sets of representative items with the respective average scores for each item. The diagram that follows each table presents distribution data for these items. Table 1 and Figure 1 relate to the use of the PM technique, while Table 2 and Figure 2 relate to additional items dealing with a more specific characteristic of the PM technique – The scheduled switch of roles.

Table 1: Items concerning use of PM and their average score		
	The statement/item	Average (S.D)
1	I believe with "pair modeling" the models are constructed faster than with unstructured collaboration	2.7 (1.1)
2	I believe that with "pair modeling" the constructed models have fewer errors than with unstructured collaboration	3.0 (1.0)
3	I believe with "pair modeling" the constructed models are more general than models constructed with unstructured collaboration	2.9 (0.8)
4	I believe that with "pair modeling" I learn and understand more deeply	3.2 (0.9)
5	I believe the "pair modeling" technique blocks natural collaboration	3.0 (1.0)

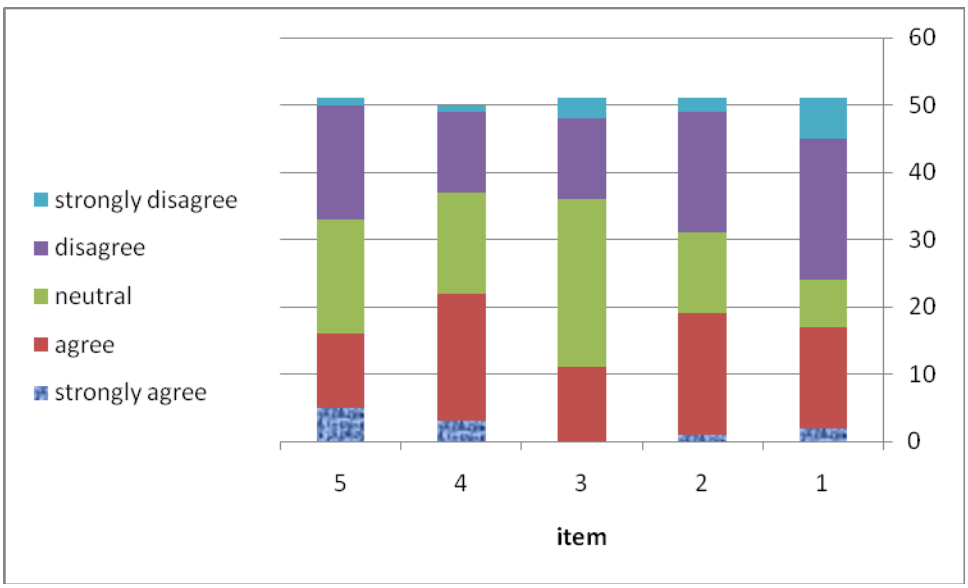

Figure 1: Distribution data for Table 1 items

Table 2: Items concerning the switch of roles and their average scores		
	The statement/item	Average (S.D.)
1	Switching roles contributed to the communication between me and my partner	3.3 (0.9)
2	Switching roles helped me to better understand the topic ("populations' behavior")	3.0 (0.8)

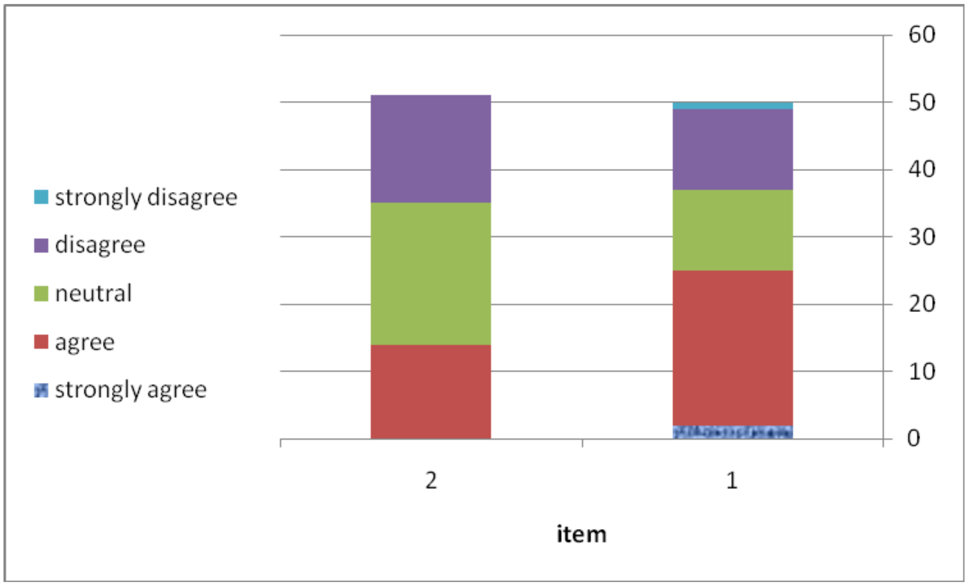

Figure 2: Distribution data for Table 2 items

Looking at all the items as presented in these diagrams it is clear that students differ in their attitudes. This is exhibited by the fact that along with the neutral attitude there are for each item a significant number of students that agree and a significant number that disagree. Note that the last statement in the first set presents a negative view regarding PM, but still we see both agreement and disagreement. When the various items are compared, one can see differences with regard to the different aspects. For example, a relatively large proportion of the students disagree that with PM models are constructed faster than with unstructured collaboration. Another example is that a relatively large proportion of the students think that switching roles contributed to the communication within the pair. These results will be further discussed in the following section.

Students' Free Text Inputs

We present here the analysis of students' inputs for two of the open-ended items.

The first one was: "In what type of activities related to modeling (e.g., defining the quantity space) you think collaboration is most useful with comparison to individual work."

We were positively surprised by the students' willingness to share their experience and views regarding this question. Only four students (out of 51 students that filled the questionnaire) did not provide an answer to this question. The inputs were all in favor of collaboration for various activities, but the activities that were mentioned varied in their level of specification. We included an example in this item in order to explain the level of specification we expect, but still most inputs related to general activities. Some inputs included several general activities such as the following input: "To create a model together is better than alone, because two have more ideas, understand better (by explaining if necessary) and can correct each other. The model is more complete and has less mistakes. Two can help each other and that makes the process a lot nicer." Related general terms that were mentioned were "critical thinking", "getting feedback", and mutual explanations ("your partner explains certain things to you and you to him/her").

Students' inputs that were more related to modeling mentioned activities such as "prediction", "noticing errors", "correcting errors" and also specific DynaLearn modeling activities (the numbers in parentheses specify the number of students that mentioned this activity): Working with quantities, defining quantity spaces and relations (5), definition of entity hierarchy (3), putting together the whole model (3), debugging a model (4), and defining scenarios (2). The last two are highly correlated as defining scenarios might be related to debugging activities and also debugging is triggered by scenario simulation results (as stated by one of the students "when the scenario does not respond the way you want it to respond").

The second open-item question was: "Please write any suggestions that you have for obtaining effective collaboration for modeling with DynaLearn". For this item we got much less inputs. Some of the inputs were suggestions, but most were reactions to the Pair Modeling approach.

Some of the suggestions related to the role of the collaboration in the course, especially as a source for feedback: "A suggestion for future collaborative work is giving feedback to each other. The one thing missing is feedback, we don't know whether we do something wrong or don't. Or at least I know I do something wrong but I don't know what it is." Students also suggested starting the collaboration earlier in the course.

Some suggestions were with regard to the grouping or pairing. One of the suggestions related to the size of the collaborating group, a suggestion to work with larger groups of 3 or 4 students. Other suggestions regarding the grouping dealt with the assignments to groups/pairs. Suggested criteria for matching included social issues like "I think that having some sort of social experience with your partner helps you collaborating more effectively because it makes it easier to relate to each other"; along with motivation/work moral criteria "It is important that both students have the same level of commitment to what has to be done". It is interesting that any criteria regarding the level of mastery for matching pairs was not mentioned by the students.

Additional inputs were related to the PM method and included general remarks about structuring collaboration, comments about the PM roles, comments about switching roles and suggestions. In general, there were inputs in favor of the PM technique and inputs against. Some students advocated sticking to the roles ("Both modelers should stick to their roles, and not forget what role they have during modeling") while others advocated adapting roles naturally during the process ("I think in natural collaboration everybody involved gives his best qualities to the process….. So if you force the roles you might ask for qualities of one that the other has more naturally").

There were also a few comments that switching roles in a structured way is beneficial only in order to support the development of communication skills ("I think pair modeling is just better in case of unequal collaboration, so if one does much more than the other or if the communication is not functioning well between the involved people"). One of the students suggested enhancing the PM technique further by changing couples some times.

Conclusions and Discussion

The main goal of our study was to examine the usefulness of the Pair Modeling (PM) technique for effective collaborative modeling. The results of the experimental study that we conducted showed that the average score for the experimental group using PM was significantly higher than the average score of the control group employing unstructured collaboration. The average score of the group that used the PM technique was 24% higher than the average score of the control group. Moreover, the four highest scores (over 70% of the maximum score) were present only in the group using PM. Based on the comparison we made with the scores of the two previous assignments, the scores' difference may be attributed to the use of the PM technique. The observations also supported the conclusion about the effect of using the PM technique on supporting productive collaboration. But one should be cautious with these results and conclusions due to several reasons. One is that the groups were relatively small - each including 28 students (14 pairs). The second reason is that the comparison to previous scores was on the basis of individual work and not on any skills influencing collaboration. It might be that the experimental group by some chance included students with better collaboration related skills. The effect that was measured in our study was on collaborative modeling as exhibited in a collaborative product. Further research might also examine effects on individual learning by individual testing.

An interesting thing we found in the results is that even though the scores seem to indicate the effectiveness of the PM technique, students did not seem to appreciate the contribution of this technique to their collaborative work. In the questionnaire that students filled out, they highly rated their appreciation for collaboration versus individual work, but their reactions towards PM were generally neutral. Closer look at the questionnaire results shows that the neutral attitude is for most items not necessarily students' choice of the neutral attitude, but really different attitudes for different students. Free text inputs also demonstrated clearly differences between students. There were inputs with clear objection to the use of PM along with other inputs recommending following strictly the roles and the switch of roles. Potential advantages of using the PM technique were not discussed with the students before or during using the technique. Such a discussion might make students more aware and maybe more motivated to use the technique and might have produced different reactions in the questionnaire.

In our study we did not use any systematic way for matching pairs. Some chose their partners (about 50%) and others were paired arbitrarily. As described in the section about pair programming, there are different ways for matching pairs. Studies that are descriptive are interesting, but prescriptive views are problematic, especially when incorporating various personality traits. Many individual differences are affected also by content and context therefore limiting the usability of any guidelines for matching. Letting students choose who to pair with might be a preferred method as was also recommended by several students in their free text input.

Students' answers to the question "In what type of activities related to modeling (e.g., defining the quantity space) you think collaboration is most useful with comparison to individual work" are important feedback for design of the visual modeling environment. All the specific things that students mentioned are concepts that have a visual reification in the environment making it useful for collaboration. The observations showed that the students often pointed on the object to clearly focus their discussion. The pointing was done by both partners, one with the mouse and the other with a finger or a pen.

The significance of this study lies in the adaptation of the pair programming technique to modeling and in studying the effects in comparison to unstructured, natural collaboration. Pair programming evolved in software development projects in the industry and this is where initial studies of the effects of pair programming started. In the industry cost effectiveness is a major consideration, so naturally pair programming was compared with individual work to justify additional expenses. The studies within CS1 courses in higher education continued this line of comparison to individual work. But in the educational context there are different consideration and collaboration is usually employed for two different reasons: collaboration is advocated as promoting learning, and shortage in resources (computer labs, teaching assistance) result in pair/team collaborative assignments (classwork and homework).This means that in the educational context further attention should be given to the exploration of collaboration support procedures with comparison to unstructured (natural) collaboration. In our study we mainly compared the employment of PM to unstructured collaboration (by scores, observations), but the questionnaire addressed also students' attitudes with comparison to individual learning in order to get a wider view on students' attitudes towards collaboration in general and towards PM in particular.

The encouraging results we got with regard to the PM technique calls for research and development within the learning companion research (Chou, Chan, & Lin, 2003). A learning companion is a computer-simulated character, which has human-like characteristics and plays a non-authoritative role in a social learning environment. Such a learning companion could alternate between the 'driver' and 'navigator' roles for supporting a student in learning to model within the DynaLearn modeling environment.

Acknowledgements

The work presented in this paper is co-funded by the EC within the 7th FP, Project no. 231526, and Website: http://www.DynaLearn.eu. The study was conducted while the first author visited the Informatics Institute, University of Amsterdam, the Netherlands.

References

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Upper Saddle River, NJ, USA: Pearson Education.

Blackwell, A. (2002). What is programming? *Proceedings of the 14th workshop of the Psychology of Programming Interest Group* (pp. 204-218). Brunel, Middlesex, UK, June 2002.

Bredeweg, B., Liem, J., Beek, W., Salles, P., & Linnebank, F. (2010). Learning spaces as representational scaffolds for learning conceptual knowledge of system behaviour. In M. Wolpers, P. A. Kirschner, M. Scheffel, S. Lindstaedt, & V. Dimitrova (Eds.), *Lecture Notes in Computer Science, Volume 6383*, 2010, Sustaining TEL: From Innovation to Learning and Practice. *5th European Conference on Technology Enhanced Learning*, EC-TEL 2010, p47-61, Barcelona, Spain, September 28 – October 1, 2010.

Bredeweg, B., Linnebank, F., Bouwer, A., & Liem, J. (2009). Garp3 - Workbench for qualitative modelling and simulation. *Ecological Informatics, 4*(5-6), 263-281.

Bredeweg, B., & Salles, P. (2009). Mediating conceptual knowledge using qualitative reasoning. In S. V. Jrgensen, T-S. Chon, & F. A. Recknagel (Eds.), *Handbook of ecological modelling and informatics* (pp. 351-398). Southampton, UK: Wit Press.

Brereton, P., Turner, M., & Kaur, R. (2009). Pair programming as a teaching tool: A student review of empirical studies. *Proceedings of 22nd Conference on Software Engineering Education and Training* (pp. 240-247). IEEE.

Brooks, R. (1983). Towards a theory of the comprehension of computer programs. *International Journal of Human-Computer Studies*, *18*, 543-554.

Bryant, S., Romero, P., & du Boulay, B. (2008). Pair programming and the mysterious role of the navigator. *International Journal of Human-Computer Studies, 66*(7), 519–529.

Chou, C-Y., Chan, T-W., & Lin, C-J. (2003). Redefining the learning companion: The past, present, and future of educational agents. *Computers & Education*, *40*, 255-269.

Cliburn, D. C. (2003). Experiences with pair programming at a small college. *Journal of Computing Sciences in Colleges*, *19*(1), 20-29.

Dillenbourg, P. (1999) Introduction: What do you mean by "collaborative learning." In P. Dillenbourg (Ed.), *Collaborative learning*. Oxford: Pergamon.

Frederiksen, J. R., & White, B. Y. (2002). Conceptualizing and constructing linked models: Creating coherence in complex knowledge systems. In P. Brna, M. Baker, K. Stenning, & A. Tiberghien (Eds.), *The role of communication in learning to model* (pp 69-96). London: Lawrence Erlbaum Associates.

Hoppe, U., & Ploetzner, R. (1999). Can analytic models support learning in groups? In P. Dillenbourg (Ed.), *Collaborative learning*. Oxford: Pergamon.

Jensen, R. (2003). A pair programming experience. *The Journal of Defensive Software Engineering*, *16*(3), 22-24.

Jackson, S. L., Krajcik, J., & Soloway, E. (1998). The design of guided learner-adaptable scaffolding in interactive learning environments. *Proceedings of ACM CHI'98*, pp 187-194. ACM Press.

Kamthan, P. (2008). Pair modeling. In G. D. Putnik & M. M. Cunha (Eds.), *Encyclopedia of networked and virtual organizations* (pp 1171-1178). IGI Global.

Lesh, R., &Doerr, H. M. (2000). Symbolizing, communicating, and mathematizing: Key components of models and modeling. In P. Cobb, E. Yackel, & K. McClain (Eds.), *Symbolizing and communicating in mathematics classrooms: Perspectives on discourse, tools, and instructional design* (pp. 361–383). Mahwah, NJ: Lawrence Erlbaum Associates.

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *Proceedings of the Thirty-Fourth Technical Symposium on Computer Science Education (SIGCSE 2003),* pp. 259-362. ACM Press.

Nosek, J. T. (1998). The case for collaborative programming. *Communications of the ACM*, *41*(3), 105-108.

Okada, T., & Simon, H. A. (1997). Collaborative discovery in scientific domain. *Cognitive Science*, *21*, 109–146.

Penner, D. E. (2001). Cognition, computers, and synthetic science: Building knowledge and meaning through modeling. *Review of Research in Education*, *25*, 1-35.

Pennington, N., Lee, A., & Rehder, B. (1995). Cognitive activities and levels of abstraction in procedural and object-oriented design. *Human-Computer Interaction*, *10*, 171-226.

Salleh, N., Mendes, E., Grundy, J., & Burch, G.J. (2009). An empirical study of the effects of personality in pair programming using the five-factor model. *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pp.214-225.

Schwarz, C. V., & White, B. Y. (2005). Metamodeling knowledge: Developing students' understanding of scientific modeling. *Cognition and Instruction*, *23*(2), 165–205.

Schwartz, C. V., Reizer, B., Davis, E. A., Kenyon, L., Acher, A., Fortus, D., Schwartz, Y., Hug, B., & Krajcik, J. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. *Journal of Research in Science Teaching*, *46*(6), 632–654

Sfetsos, P., Stamelos, I., Angelis, L., & Deligiannis, I. (2009). An experimental investigation of personality types impact on pair effectiveness in pair programming. *Empirical Software Engineering*, *14*(2), 187-226.

Springer, L., Stanne, M. E., & Donovan, S. (1999). Effects of small group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis. *Review of Educational Research*, *69*, 21–51.

Thomas, L., Ratcliffe, M., & Robertson, A. (2003). Code warriors and code-a-phobes: A study in attitude and pair programming. *Proceedings of the Thirty-Fourth Technical Symposium on Computer Science Education (SIGCSE 2003),* pp. 363-367. ACM Press.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—An embodied modeling approach. *Cognition and Instruction*, *24*(2), 171–209.

Williams, L., & Kessler, R. (2003). *Pair programming illustrated*. Boston, MA: Addison Wesley.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software, 17*(4), 19-25.

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science courses. *Computer Science Education, 12*(3), 197 - 212.

Windschitl, M., Thompson, J., & Braaten, M. (2008). Beyond the scientific method: Model-based inquiry as a new paradigm of preference for school science investigations. *Science Education*, *92*(5), 941–967.

Appendix A

Course - Conceptual modelling

Assignment 7c: Population by Birth, Mortality and Migration

Session:	Thursday, October 14th, 2010
Deadline:	Thursday during the session (before 14.00 hours)
Duration:	Approximately 30 minutes
Software:	Group 1 (Room P126): DynaLearn – LS6
	Group 2 (Room P127): DynaLearn – LS6

Note: Read the entire document carefully before starting

1. Procedure – Always proceed as follows

Step 1: Open assignment 7b and take the results obtained there as a starting point.
Step 2: Do the assignment (see details below).
Step 3: **Submit via Blackboard**

- If you think you are finished: upload your result to Blackboard (please give the file a good name).

Note. Assistants only provide feedback on software usage, and not on the domain knowledge. You yourself must think about what the model should look like and what knowledge is applicable!

2. Assignment details

The goal of this assignment is to add Migration. Consider the population discussed in the previous assignments as the system we are dealing with. Migration has two aspects: individuals joining a population (*Immigration*), and individuals leaving a population (*Emigration*). In this assignment each should be represented as a process by its own. Notice that *Immigration* can be seen as analogous to the *Birth* process, while *Emigration* can be seen as analogous to the *Death* process. However, there are subtitle differences! E.g., here we take the position that *Emigration* has a positive feedback coming from the population size (*Number_of*), while *Immigration* does not. Moreover, *Immigration* has exogenous quantity behaviour (for instance, parabola positive, or increase).

When simulating the model, run a simulation in which the *Number_of* starts at *Zero*. This should result in a state-graph in which this population starts growing from *Zero* and reaches it maximum size. Also run a simulation in which *Number_of* starts at a value higher than *Zero*. Does this situation lead to the population becoming extinct (thus *Number_of* going to *Zero*)?

Thoroughly, check your model running different scenarios. Ensure that your model covers all the 'obvious' states and behaviour paths given the four processes now active in the model. Save at least 4 alternative scenarios in your model.

Appendix B

Dear Student,

The role of this questionnaire is to gather information about your attitude to collaborative learning and especially to the "Pair Modeling" technique. The questionnaire should be filled individually.

1. I usually prefer to study by myself

   Yes No

2. I consider myself:

   Introvert Extrovert

3. Rate your mastery of the DynaLearn software on a scale 1-5

Below average	average	good	Very good	excellent

4. Rate your mastery of "populations' behavior" on a scale 1-5

Below average	average	good	Very good	excellent

5. Rate the degree you and your partner stick to the roles of "driver" and "navigator" (5 – strictly)

Not at all				strictly

6. I believe constructing the models collaboratively produces better models than individual construction

Strongly disagree	disagree	neutral	agree	Strongly agree

7. I would prefer to start modeling by myself and only later on work collaboratively

Strongly disagree	disagree	neutral	agree	Strongly agree

8. I would like to always play the role of the "driver"

Strongly disagree	disagree	neutral	agree	Strongly agree

9. I would like to always play the role of the "navigator"

Strongly disagree	disagree	neutral	agree	Strongly agree

10. I believe the "pair modeling" technique blocks natural collaboration

Strongly disagree	disagree	neutral	agree	Strongly agree

11. I believe with "pair modeling" the models are constructed faster than with unstructured collaboration

Strongly disagree	disagree	neutral	agree	Strongly agree

12. I believe that with "pair modeling" the constructed models have fewer errors than with unstructured collaboration

Strongly disagree	disagree	neutral	agree	Strongly agree

13. I believe with "pair modeling" the constructed models are more general than models constructed with unstructured collaboration (so that multiple scenarios can be simulated with the same modeling details and/or have more model fragments that can be reused)

Strongly disagree	disagree	neutral	agree	Strongly agree

14. I believe that with "pair modeling" I learn and understand more deeply

Strongly disagree	disagree	neutral	agree	Strongly agree

15. Switching roles contributed to the communication between me and my partner

Strongly disagree	disagree	neutral	agree	Strongly agree

16. Switching roles helped me to better understand the topic ("populations' behavior")

Strongly disagree	disagree	neutral	agree	Strongly agree

17. Order the following DynaLearn ingredients according to the degree you find them useful for focusing discussions with your partner about the model

    Entity hierarchy
    States graph
    Value history graphs for specific paths
    Value history graphs for specific sets of states (not necessarily paths)
    Dependencies diagrams for specific states

18. In what type of activities related to modeling (e.g. defining the quantity space) you think collaboration is most useful with comparison to individual work.
19. What procedures did you and your partner use for checking the correctness of a model? If you remember please add for each procedure whether it was suggested/initiated by the "driver" or the "navigator" at that stage.
20. Please write any suggestions that you have for obtaining effective collaboration for modeling with DynaLearn.

Biographies

Rachel Or-Bach is a senior lecturer in the Information Management Systems department in the Academic College of Emek Yezreel . or-bach@yvc.ac.il . She received her Ph.D. from the Technion-Israel Institute of Technology, her M.SC. from the Weizmann Institute; and her bachelor degree in applied mathematics from the Technion. Her main research interest is design of interactive learning environments: representations, interactivity, and intelligent support and adaptation.

Bert Bredeweg is as an associate professor at the Informatics Institute in the University of Amsterdam. He leads the Qualitative Reasoning group and is project coordinator of the international EU co-funded FP7 project DynaLearn (http://www.DynaLearn.eu). His research is driven by fundamental questions about computational intelligence and includes themes such as: knowledge capture, qualitative reasoning, learning by modelling, cognitive diagnose, and human-computer interaction. Bredeweg has published over 117 international refereed academic publications and over 116 other academic publications.