Collaborative Learning of UML and SysML

doi:10.3991/ijep.v1i2.1663

Ervin Ramollari[1], Matthias Heintz[2], Sebastian Weber[2], Sonja Trapp[2], Dimitris Dranidis[1], Jürgen Börstler[3]
[1] South East European Research Centre (SEERC), Thessaloniki, Greece
[2] Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany
[3] Blekinge Institute of Technology, Karlskrona, Sweden

***Abstract*—In the EU-funded project Embed4Auto, we have developed an Integrated Learning Environment (ILE) to support UML/SysML training. The ILE combines independent distance learning with collaborative problem solving and social networking. The learning environment incorporates a set of learning resources on the modeling languages UML2 and SysML, an educational modeling tool supporting class and sequence diagrams and a tool for sharing and discussing solutions to modeling exercises. First evaluations of the ILE have shown promising results.**

***Index Terms*—Collaborative learning, computer-assisted instruction, distance learning.**

I. INTRODUCTION

Software plays an increasingly important role in the engineering of embedded systems. With a rising level of abstraction necessary to handle such complex systems of hardware and software, model-based development is indispensable for reducing complexity and making the development of such systems more efficient and more failure resistant.

In *Embed4Auto* (http://www.embed4auto.org/), we analyzed the training needs of European industry in the area of embedded systems development. Based on these needs, we developed an integrated learning environment (ILE) that combines individual distance learning with collaborative problem solving and social networking.

The ILE incorporates (1) a set of learning resources on the modeling languages UML [1] and SysML [2], (2) a Web-based collaboration tool for sharing and discussing (solutions to) exercises, as well as (3) a modeling tool with educational capabilities [3], [4].

The learning resources include (1) short traditional lessons, (2) examples and case studies, and (3) multiple choice questions as well as interactive modeling exercises.

The paper is structured as follows: in chapter II, we present the relevant results of the educational needs analysis, and provide an overview of the developed system in chapter III. In chapter IV, we give an introduction to the developed collaboration tool, and then describe the new modeling tool in chapter V. Chapter VI sums up the results of two evaluation sessions and several usability tests. In chapter VII, we compare our newly developed ILE with several other tools available on the educational market. A short conclusion can be found in the last chapter.

II. EDUCATIONAL REQUIREMENTS

A major outcome of the project has been an analysis of the training needs and preferences for learning support in European industry. A questionnaire was distributed to a selected target group of individuals; see Table I for an overview. In total, 272 responses were received, mainly from the countries involved in the project (Germany, 28.79%; Greece, 6.82%; Spain, 37.88%; Sweden, 12.12%, Turkey, 9.85%; Others 4.54%).

TABLE I.
STRUCTURE OF NEEDS ANALYSIS QUESTIONNAIRE.

Part	Description	Questions
1	Demographical information about respondent (general and professional)	4
2	Demographical information about respondent's company/organization	5
3	Respondent's practical experience relevant to embedded/automotive modeling and development	2
4	Respondent's perceived training needs regarding his or her company/ organization	5
5	Open questions regarding further comments and feedback	5

Respondents identified the following three areas as most important for further training (in decreasing order of importance):

1) UML2,
2) Requirements engineering, and
3) (Software) Architecture.

These three top areas were highly ranked fairly independently of user experience, company size, and sector. The areas served as the basis for designing the course material, which mainly covers UML2 and SysML diagrams.

Although the preferences for learning support showed a less uniform picture, all groups had a very strong and consistent preference for active support from tutors and some form of tool-based assessment. There was also a consistent preference for self-paced learning with self-selected learning activities. Groups showed a varying preference for group vs. independent learning. Regarding supporting materials and tools, all groups rated case studies and exercises with interactive feedback very high [5].

Summing up, it seemed that some form of blended learning was required to balance the conflicting preferences regarding social interaction and the degree of freedom in learning sequence and activities.

III. SYSTEM OVERVIEW

Dealing with the above educational requirements we developed the PENCIL (Platform for Exercise sharing aNd Collaborative and Interactive Learning) integrated learning environment (ILE). To allow for independent and self-paced learning, the PENCIL ILE incorporates predefined courses and learning content, similarly to a traditional learning management system. To enable active support from tutors and learning in groups, a Web 2.0 platform for sharing and discussing exercises was integrated into the ILE. To address the preferences for tool support and exercises with interactive feedback, we also integrated an educational modeling tool for a subset of UML diagrams.

Taken together, the PENCIL ILE supports both individual self-directed learning, and collaborative learning in social groups. Fig. 1 illustrates the high-level components of the implemented ILE.

Part of the ILE is Moodle[1], which is an open-source course management system used for structuring courses, announcements, general information, contact information, etc. Moodle can be seen as an entry point for the learners and tutors into the other ILE components. Moodle also contains the actual learning content structured into courses and lessons. Notably, this architecture is not specifically tied to Moodle, but can accommodate any other content management platform or even website.

The learning content is designed for web-based training, with the opportunity for students to resort to print versions (pdf) of the lessons if desired. Powerpoint slides were at the starting point of content creation: with the help of a dedicated tool, the slides were then transformed into the more ubiquitous Flash format so that the lessons can be run on any Web page or learning management system (LMS). Furthermore, SCORM (Sharable Content Object Reference Model) and AICC (Aviation Industry Computer-Based Training Committee)-compliant content can be created for sequencing and tracking purposes on any kind of LMS.

Besides the theoretical content, the learning materials include exercises. These range from browser-based multiple choice questions, with automatic feedback, to modeling exercises in UML. For the modeling exercises, we utilize StudentUML [4] for drawing UML diagrams as solutions. With its built-in consistency checking mechanisms, StudentUML identifies many defects that are common for beginners.

To enable sharing and discussion of models, we developed and integrated a tool called PINBOARD (PENCIL client for dIscussing solutioNs collaBOratively by leARners anD tutors). PINBOARD is accessed from within the ILE through links in the modeling exercise descriptions, leading to a listing of the solutions submitted for each of the exercises. Thereon, learners can open, comment, and even modify submitted solutions. Doing so, learners collaboratively contribute step by step towards a commonly accepted solution. Educators are also able to join learners and provide comments or feedback on specific “states” of a solution.

PENCIL offers a common server-side API (Application Programming Interface) for client tools to connect. The API operates on a database which serves as a common repository for all the data, including exercises, solutions and users. The PENCIL API promotes loose coupling in a way that the ILE remains independent of specific applications, such as PINBOARD, StudentUML, or Moodle. This makes it possible to add further tools, for example solution authoring applications that support notations other than UML.

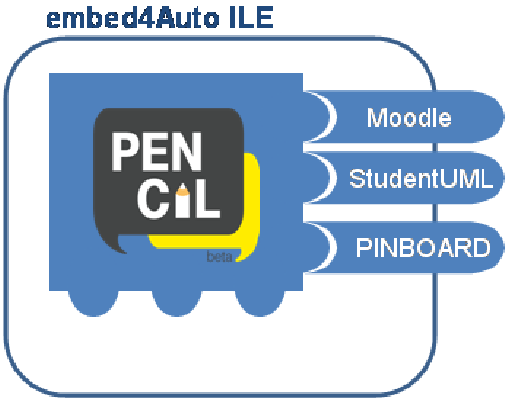


Figure 1. The high-level overview of the Embed4Auto ILE.

IV. COLLABORATION

PINBOARD is a social Web 2.0 tool that encourages collaborative problem solving in a community of learners and tutors. It supports collaboration in terms of discussions about solved and partly solved exercises.

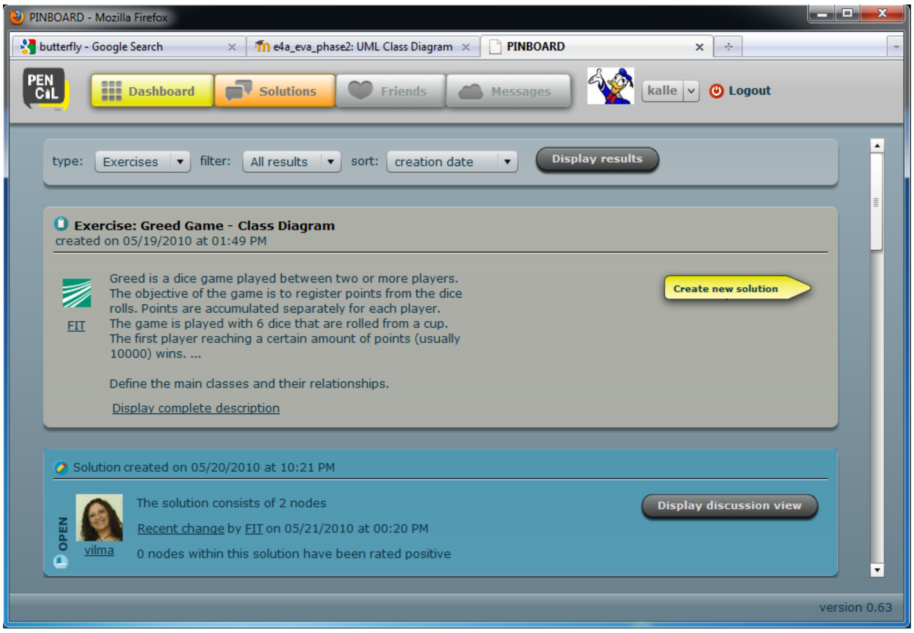
Besides starting a discussion during the solution creation process, e.g. by publishing different stages of a solution, PINBOARD also allows learners to attach their already created StudentUML diagrams to solutions of other students, e.g., as a hint (in a way that a student will know how to go on), as an alternative (in order to show, e.g., a simpler solution), or as a suggestion for improvement (in order to reveal flaws). In the context of PINBOARD, interaction means providing feedback in the form of comments or attached modified solutions. Learners do not actually work simultaneously on the creation of a single diagram. Instead, the term “collaborative” is used in the sense that students engage in problem solving activities with others (fellow students, tutors, and educators).

PINBOARD is designed to foster the idea of learning by teaching, where learners help their peers. Learners, tutors, and educators can upload solutions into StudentUML, modify them, and then share their new/revised solutions. Such shared contributions can be understood as hints which bring fellow learners back on track or provide alternative solutions to an exercise. In this way, learners not only acquire theoretical knowledge and actively create diagrams with StudentUML, they also learn by helping others or by receiving help from peers and by discussing the problems with fellow students.

In the solutions view, learners can search or navigate all solutions to all exercises. When selecting an exercise, all discussion threads are presented in a list. Each node in this list contains some basic information about the thread, like its original author, how many contributions (“nodes”) it contains, etc., see fig. 2.

When opening a discussion thread (button “Display discussion view”), the learner can access all contributions in that particular thread and comment on them. Fig. 3 shows an example of user “mia” commenting on a diagram posted by user “kalle”. Existing diagrams can be opened for viewing or for editing by all users of the system. Editing diagrams is done using the integrated modeling tool (see next section).

[1] http://www.moodle.org



Exercise: Greed Game - Class Diagram
created on 05/19/2010 at 01:49 PM

FIT

Greed is a dice game played between two or more players. The objective of the game is to register points from the dice rolls. Points are accumulated separately for each player. The game is played with 6 dice that are rolled from a cup. The first player reaching a certain amount of points (usually 10000) wins. ...

Define the main classes and their relationships.

Display complete description

Solution created on 05/20/2010 at 10:21 PM

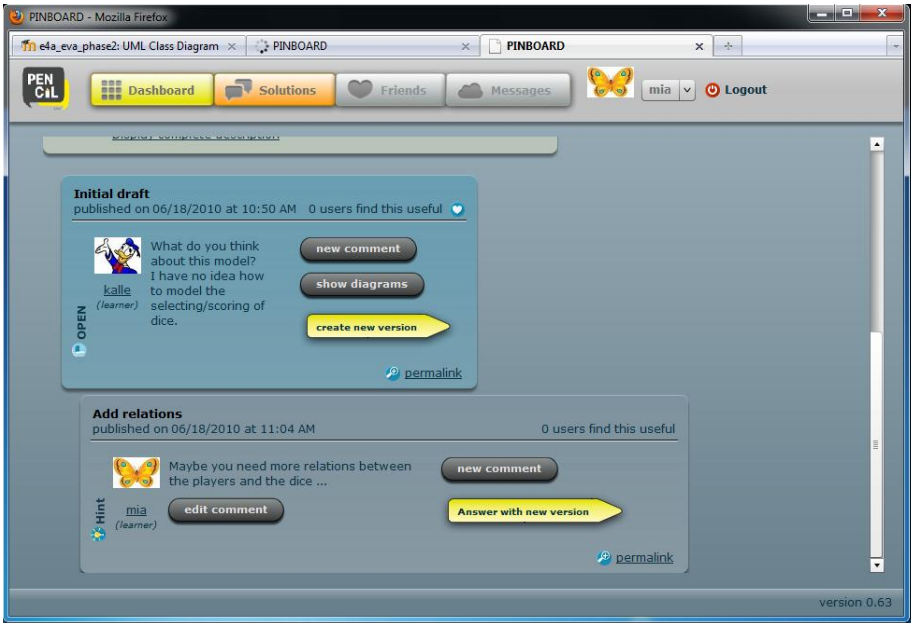
vilma

The solution consists of 2 nodes

Recent change by FIT on 05/21/2010 at 00:20 PM

0 nodes within this solution have been rated positive

Figure 2. Example solution view (user “kalle”).



Initial draft
published on 06/18/2010 at 10:50 AM 0 users find this useful

kalle (learner)

What do you think about this model? I have no idea how to model the selecting/scoring of dice.

Add relations
published on 06/18/2010 at 11:04 AM 0 users find this useful

mia (learner)

Maybe you need more relations between the players and the dice ...

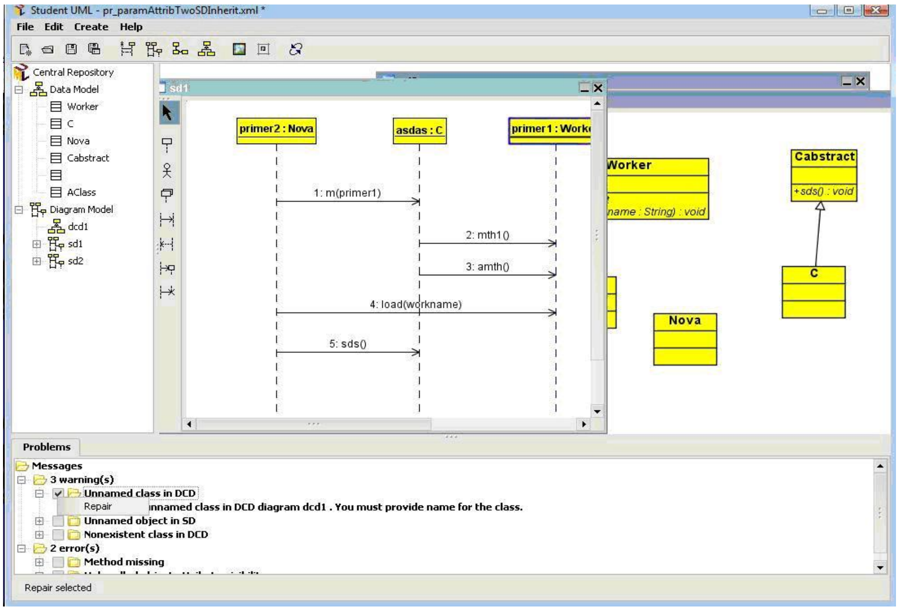
Figure 3. Example discussion thread.

V. MODELING

The modeling tool forming part of our ILE has been developed by extending StudentUML [3], [4]. StudentUML is an educational tool that provides students with a simple yet effective tool that meets their learning needs. A crucial requirement is that the modeling tool be easy to learn and use by the learner community. StudentUML was therefore designed as a minimalist tool. It excludes any unnecessary features found in professional modeling and CASE tools, which would distract students from the learning process. In addition, StudentUML consists of an uncluttered graphical user interface (GUI) and supports only a subset of the UML2 notation (see fig. 4). Currently, UML class and sequence diagrams are supported, which we consider the most important diagrams for teaching object-oriented software development [6].

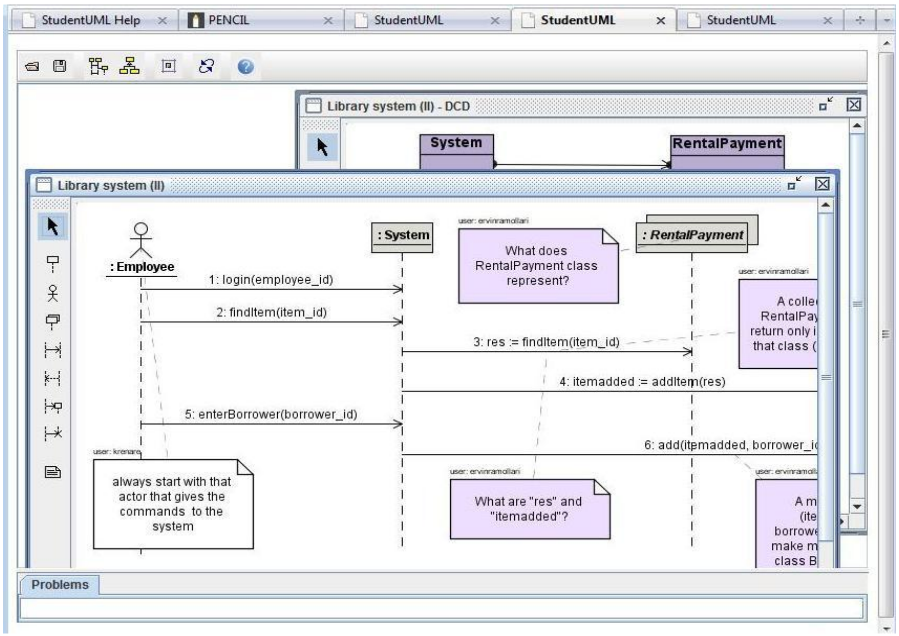
An important feature of StudentUML is its ability to validate diagrams for consistency. A number of studies have shown that awareness of consistency between modeling artifacts is of high importance when learning UML [7], [8]. For example, Lange et al. [8] have shown that a large number of consistency defects are frequently not detected and that undetected defects cause serious misinterpretations later in the development lifecycle.

StudentUML supports real-time consistency validation of individual and multiple UML diagrams. The validation is based on editable Prolog rules, allowing for adaptation or extension of this functionality to specific educational needs. Validation results are structured in terms of errors and warnings that point out the type of problem. StudentUML also gives the learner the possibility to repair consistency problems automatically. Evaluations indicate that the consistency validation feature of StudentUML sub-



Messages
3 warning(s)
☑ Unnamed class in DCD
Repair
nnamed class in DCD diagram dcd1 . You must provide name for the class.
□ Unnamed object in SD
□ Nonexistent class in DCD
2 error(s)
□ Method missing
Repair selected

Figure 4. Screenshot of the standalone version of the StudentUML modeling tool.



StudentUML Help
PENCIL
StudentUML
StudentUML
StudentUML
Library system (II) - DCD
System
RentalPayment
Library system (II)
: Employee
: System
: RentalPayment
user: ervinramollari
What does RentalPayment class represent?
user: ervinramollari
A colle RentalPay return only i that class (
1: login(employee_id)
2: findItem(item_id)
3: res := findItem(item_id)
4: itemadded := addItem(res)
5: enterBorrower(borrower_id)
6: add(itemadded, borrower_i
user: krenare
always start with that actor that gives the commands to the system
user: ervinramollari
What are "res" and "itemadded"?
user: ervinramoll
A m (ite borrowe make m class B
Problems

Figure 5. Screenshot of the StudentUML Web application as part of the learning environment.

stantially supports students in their learning of the taught concepts [10].

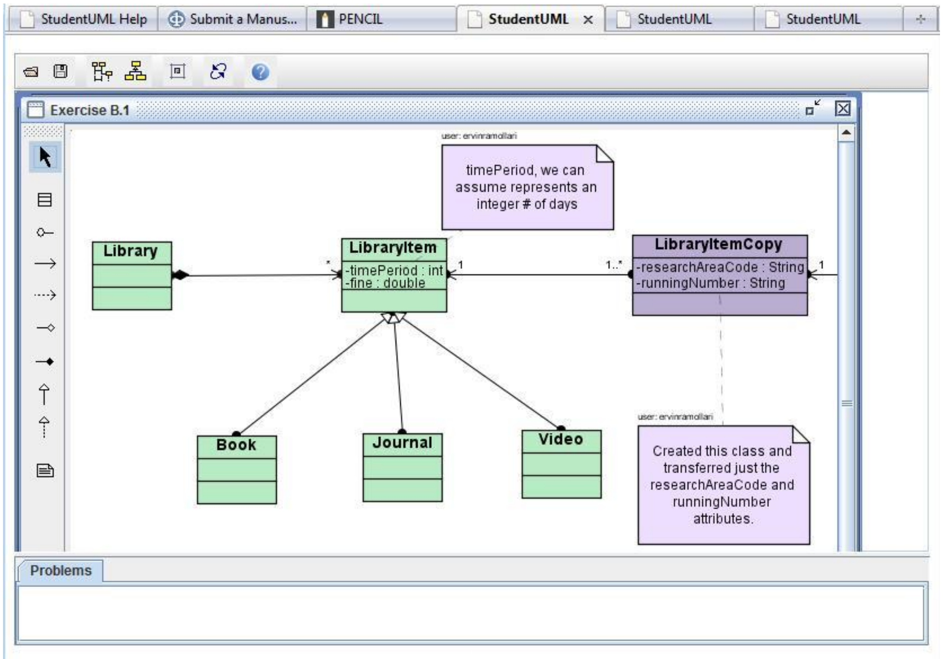
Since a Web-based learning environment is central to the concept of our ILE, StudentUML has been transitioned into a Rich Internet Application (RIA) embedded in a Web page (see fig. 5). Currently, the tool has been implemented as a JavaTM applet, which is accessible from external links with well-defined URIs and initialised by URI parameters. As a result, users do not need to download and install any software locally and solution files are stored remotely on the server.

StudentUML has been integrated with the PINBOARD solution sharing platform; both applications access the same data repository via a well-defined API of query and update operations offered by PENCIL.

To enhance user collaboration on creating UML diagrams, the tool facilitates distinguishing content contributed by different users. Each contributor in a UML diagram is assigned a unique random colour and UML elements created by that user are filled with the corresponding colour. Moreover, UML notes are preceded by the authors' PENCIL nicknames in order to know who writes what (Fig. 6).

For individual usage mode, the tool feedback mechanism has been extended with help links pointing to the learning material contained in the course repository. For instance, whenever a consistency defect is detected, the tool provides a help link to the already existing learning material related to that defect.

A planned feature for future releases of the ILE is support for additional types of UML2 and SysML diagrams. We are also experimenting with semi-automatic assessment of diagrams by matching diagrams with model solutions by utilizing the Prolog rule-base.



timePeriod, we can assume represents an integer # of days

Created this class and transferred just the researchAreaCode and runningNumber attributes.

Figure 6. UML elements created by a user are filled with that user's color while notes are preceded by the user's PENCIL nickname.

VI. EVALUATION

An evaluation of the first release of the system took place in all partner countries (Germany, Greece, Spain, Sweden, and Turkey) during a two-week period with more than 50 participants (students and professionals) [11].

Prior to the evaluation sessions some arrangements were made. Participants who could communicate in English were identified. A questionnaire with a validation checklist was created and put on the open source survey application Lime Survey[2]. An introductory meeting with all participants was conducted by the instructors. Participants were helped regarding system problems, bugs and difficulties, and they were reminded of and encouraged to ask questions in English on PENCIL to enable communication or collaboration. In case of a conflict between students and their solution, the instructors provided resolving feedback and concrete solutions in cases one was not found already.

During the evaluation, several modeling exercises were made available, which the participants had to solve. The possibility to share and discuss solutions was well accepted. Most participants did actively participate in the discussions by sharing their solutions (class and sequence diagrams), commenting solutions by others, and by actually modifying and resubmitting shared solutions.

In the follow-up questionnaire, the learning environment was evaluated with respect to overall design, pedagogical design and user satisfaction. See table II for a selection of results.

The evaluation results showed that participants enjoyed working with the ILE tools and that the collaboration, in the form of solution discussion, with the other participants and the tutors helped them in comprehending the taught material. Most of the problems reported by users concerned usability issues which are going to be constantly addressed in later versions of the tools.

TABLE II.
EVALUATION RESULTS CONCERNING PEDAGOGICAL DESIGN.
N=NUMBER OF RESPONSES, M=MEAN, SD=STANDARD DEVIATION.

Pedagogical Design	N	M	SD
Feedback from other users was helpful to my comprehension	39	3,9	0,8
Feedback I wrote was helpful to my comprehension	34	4,0	0,7
Feedback from the tutor was helpful to my comprehension	34	4,0	0,9
Discussing the exercises & solutions contributed to my learning	37	4,2	0,8
I enjoyed interacting with the community of learners	38	4,2	0,8
The modeling tool is easy to learn and use	39	4,2	0,8
The consistency checking capabilities in the modeling tool helped me better understand the relationships between UML diagrams	35	4,0	0,9
Instant feedback mechanism of Student UML helped me to learn from my mistakes	35	3,9	0,8
Student UML was helpful to me	40	4,2	0,7
PENCIL was helpful to me	36	4,0	0,8
Learning Resources (Content) was helpful to me	39	3,8	0,8
Multiple Choice questions were helpful to me	36	3,8	1,1
The combination of online material and collaboration/exercise tools motivated me to study	36	3,9	1,0
I found the multiple choice questions more helpful to my comprehension	33	3,6	0,9
I found the problem solving exercises more helpful to my comprehension	36	4,1	0,8

[2] http://www.limesurvey.org/

The second evaluation [12] was focused on user satisfaction and technology acceptance. More than 60 participants answered the questionnaire and the results were again largely positive. Most of the users found the discussions helpful for understanding other points of view (mean=3.7) and thought that the course was a useful learning experience for them (mean=3.6). The users were satisfied with the collaborative learning environment and with being part of a learning community in their groups (mean=3.3). They developed their problem solving skills through peer collaboration (mean=3.4) and found collaborative learning effective (mean=3.4). They were only prudent towards the item “Collaborative learning experience in the computer-mediated communication environment is better than in a face-to-face learning environment.”

In addition, think-aloud tests [13] were carried out with a total of 23 users and protocols were documented by all partners. The results showed that there is still some work to be done in improving the graphical interface and the navigation in order to make the system a fully satisfying user experience.

VII. RELATED WORK

In the area of teaching UML, there are tools that support independent learning as well as tools for collaborative learning in groups. Related tools that support independent learning are stand-alone modeling applications with various educational features. These features include minimalism, consistency checking, exercise support, and other intelligent tutoring features.

Minimalist tools aim to be easy to learn and use by students in objects-first classes. Therefore, they are characterized by a reduced number of features, a simple user interface, and support for a small subset of the graphical modeling notation (usually UML) in order to reduce confusion. Among these tools are QuickUML [10], minimUML [6], UMLet[3], and Violet[4]. A more detailed review of their capabilities is provided in [4]. These minimalist tools lack support for educational features.

Exercise support is another feature that may refer to different kinds of capabilities. Some existing standalone tools that we have reviewed support automated exercise assessment, a useful capability that we are investigating for further releases of the ILE. Two representative tools in this category are Diagram and COLLECT-UML. Diagram [14] is a learning environment for object-oriented modeling that compares the student’s diagram with an expert’s diagram in order to find the differences between those two diagrams. The principles of the comparison algorithm rely on graph-matching methods and algorithms. On the other hand, COLLECT-UML [15] is a constraint-based tutor for UML modeling, which provides both individual and collective feedback. The system compares the submitted student solution with its own ideal solution, according to the system’s domain model, represented in the form of constraints. If a constraint is violated, the associated message is provided to the student.

On the other hand, there exist Web 2.0 tools that support learning in groups. Many Web 2.0 tools have been designed for a different purpose but can be used for the discussion of exercises, e.g., Question & Answer (Q&A) pages like Yahoo! Answers[5]. The purpose of such Q&A platforms might not be learning, but you can ask about a problem you have with your (UML) exercise and could get helpful answers. Besides this platform where interaction does not happen in real time, there are also some interactive online whiteboard tools (e.g., imaginationcubed[6]), which were not created for e-learning purposes either, but could be used to draw and chat simultaneously about UML diagrams. Although those tools could be used for the same purpose as our ILE, we believe that their integration would not be as smooth as the combination of PENCIL and StudentUML. Besides, they would not give the user the best possible support for his current task, which our ILE is intended to do, because it has been specifically designed to support learners and tutors.

Mathepower[7] is a platform to help students solve math exercises. The approach is similar to our planned ILE: A student can post his maths problem and other users can answer him in the forum to provide hints or the solution. This forum-based platform is great for maths exercises or any other exercise that can be answered with a text. However, it reaches its limits if graphical solutions are necessary, which is the case with UML exercises. Because our ILE integrates the presentation of UML diagrams through StudentUML, we go beyond this limitation of pure text.

With complex highly interactive exercises, such as designing a UML diagram, computer-based revision is difficult or even impossible to perform. Our tool is a good alternative to the common multiple-choice tests[8] with exercises that are not very close to real-world problems but can easily be solved by a machine. With online support for any open question, either from fellow students or tutors, the learner can solve real-world problems and receive human support. To the best of our knowledge PENCIL is the first ILE for UML/SysML with support for collaborative problem solving and social networking.

VIII. CONCLUSION

The paper presented a novel integrated learning environment that addresses the unique needs of the Embed4Auto project. The learning environment combines individual learning with collaborative problem solving and social networking. Web 2.0 technologies are exploited, adapted, and integrated in order to create a challenging environment for flexible training. The mutual combination and seamless integration of the learning resources, the PINBOARD collaboration platform for solution sharing and discussing, and the StudentUML educational modeling tool for authoring exercise solutions assist collaborative learning and problem solving in an integrated learning environment for UML/SysML. Results obtained from two evaluation sessions indicate that the objectives have been achieved.

REFERENCES

[1] OMG, Unified Modelling Language, [Online]. Available: http://www.omg.org/spec/UML/

[2] T. Weilkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design* (The OMG Press), Morgan Kaufmann, 2008.

[3] http://www.umlet.com/

[4] http://www.horstmann.com/violet/

[5] http://answers.yahoo.com

[6] http://imaginationcubed.com

[7] http://www.mathepower.com/english/index.html

[8] which e.g. have been used in the Up2uml project, see http://www.up2uml.org

[3] D. Dranidis, “Student UML: Learning to Model Consistent UML Diagrams”, In: *Teach UML: An interdisciplinary workshop on teaching Software Modeling with the Unified Modeling Language in Academia and Industry,* 2007.

[4] E. Ramollari and D. Dranidis, “StudentUML: An Educational Tool Supporting Object-Oriented Analysis and Design”, In: *Proceedings of the 11th Panhellenic Conference on Informatics (PCI 2007)*, pp. 363-373.

[5] J. Börstler, Needs Analysis Report, Internal Project Report D2.1, Embed4Auto Consortium, Mar 2009, unpublished.

[6] S.A. Turner, M.A. Perez-Quinones and S.H. Edwards, “minimUML: A Minimalist Approach to UML Diagramming for Early Computer Science Education”, Computing Research Repository, http://arxiv.org/abs/cs.HC/0603121, 2005.

[7] L. Kuzniarz and M. Staron, “Best Practices for Teaching UML Based Software Development”, Satellite Events at the *MoDELS 2005 Conference*, pp. 320-332.

[8] C.F.J. Lange and M.R.V. Chaudron, “Effects of Defects in UML Models: an Experimental Investigation”, In: *Proceedings of the 28th International Conference on Software Engineering,* Shanghai, China: ACM, 2006, pp. 401-411.

[9] J. Simmonds and M.C. Bastarrica, “Description Logics for Consistency Checking of Architectural Features in UML 2.0 Models”, Departamento de Ciencias de la Computacion, Facultad de Ciencias Fısicas y Matematicas, Universidad de Chile.

[10] E. Crahen, C. Alphonce and P. Ventura, “QuickUML: a Beginner's UML Tool”, In *Companion of the 17th Annual ACM SIGPLAN Conference on Object-oriented Programming,Systems, Languages, and Applications*. Seattle, Washington, 2002, pp. 62-63.

[11] Ö.F. Bay and M. Baturay, Evaluation Report, Phase 1, Internal Project Report D5.3, Embed4Auto Consortium, Feb 2010, unpublished.

[12] Ö.F. Bay and M. Baturay, Evaluation Report, Phase 2, Internal Project Report D5.5, Embed4Auto Consortium, Oct 2010, unpublished

[13] M.W. van Someren, Y.F.Barnard and J.A.C. Sandberg, *The Think-Aloud Method - A Practical Guide to Modelling Cognitive Processes,* 1994.

[14] L. Auxepaules, D. Py and T. Lemeunier, “A Diagnosis Method that Matches Class Diagrams in a Learning Environment for Object-Oriented Modeling”, *Eighth IEEE International Conference on Advanced Learning Technologies (ICALT'08),* pp. 26-30.

[15] N. Baghaei and A. Mitrovic, “A Constraint-Based Collaborative Environment for Learning UML Class Diagrams”, *Intelligent Tutoring Systems,* Lecture Notes in Computer Science, LNCS 4053, Springer, 2006, pp. 176-186.

AUTHORS

Ervin Ramollari holds an MSc in Software Engineering and Telecommunications from CITY College, affiliated institution of the University of Sheffield. Currently, he is a PhD candidate at the South East European Research Centre (SEERC), a research centre of the University of Sheffield and CITY College, in Thessaloniki, Greece. His research interests include SOA and Web services, formal verification of systems, object oriented A&D with the UML, and CASE tools.

Matthias Heintz is a scientist at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany. He holds a Master's degree in Computer Science in Media. His research interests are Web 2.0, human computer interaction and serious games.

Sebastian Weber is a scientist at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany. He an M.S. (Diplom) in applied computer science from the University of Kaiserslautern, Germany. His research interests are eLearning, Web 2.0, semantic technologies, intelligent assistance, and software diagnosis.

Sonja Trapp is a senior scientist at Fraunhofer IESE, holding University degrees in Linguistics and Adult Education. She is the project manager of the Embed4Auto project and has been working in research and industrial projects on technology-enhanced learning for more than 10 years.

Dimitris Dranidis is a senior lecturer in the Department of Computer Science at CITY College, an International Faculty of the University of Sheffield and an associate researcher at SEERC (South East European Research Centre). His research interests are in the areas of service-oriented computing, object-oriented analysis and design, model-based verification, formal methods, software engineering education, and agile software development methodologies.

Jürgen Börstler is an associate professor of Software Engineering at Blekinge Institute of Technology, Sweden and a founding member of the Scandinavian Pedagogy of Programming network. His main research interests are in requirements engineering, object-oriented modeling, software process improvement, and computer science education

This work has been performed as part of the Embed4Auto project, which is supported by the Leonardo da Vinci programme of the European Commission (grant DE/08/ LLP-LdV/TOI/147109).

Received, May 5th, 2011. Published as resubmitted by the authors June 6th, 2011.