Norsk informatikkonferanse **NIK 2010**

Høgskolen i Gjøvik 22.–24. november 2010

NIK-styret og redaksjonskomité

John Markus Bjørndalen	Universitetet i Tromsø
Trond Aalberg	NTNU
Arne Løkketangen	Høgskolen i Molde
Dag Haugland	Universitetet i Bergen
Andreas Prinz	Universitetet i Agder
Ragnhild Kobro Runde	Universitetet i Oslo
Birger Møller-Pedersen	Universitetet i Oslo
Frode Eika Sandnes	Høgskolen i Oslo (styreleder)
Erik Hjelmås	Høgskolen i Gjøvik (redaktør og arrangør)

Norsk informatikkonferanse

NIK 2010

Høgskolen i Gjøvik
22. – 24. november 2010

NIK-styret og redaksjonskomité

John Markus Bjørndalen	Universitetet i Tromsø
Trond Aalberg	NTNU
Arne Løkketangen	Høgskolen i Molde
Dag Haugland	Universitetet i Bergen
Andreas Prinz	Universitetet i Agder
Ragnhild Kobro Runde	Universitetet i Oslo
Birger Møller-Pedersen	Universitetet i Oslo
Frode Eika Sandnes	Høgskolen i Oslo (styreleder)
Erik Hjelmås	Høgskolen i Gjøvik (redaktør og arrangør)

© NIK-stiftelsen og Tapir Akademisk Forlag, 2010

ISSN 1892-0713 (trykt utg.)
ISSN 1892-0721 (online)
ISBN 978-82-519-2702-4

Det må ikke kopieres fra denne boka ut over det som er
tillatt etter bestemmelser i «Lov om opphavsrett til åndsverk»,
og avtaler om kopiering inngått med Kopinor.

Redaktør: Erik Hjelmås, Høgskolen i Gjøvik
Digital trykk og innbinding: Tapir Uttrykk

Tapir Akademisk Forlag har som målsetting å bidra til å utvikle gode læremidler og alle typer faglitteratur. Vi representerer et bredt fagspekter, og vi gir ut ca. 100 nye titler i året. Vi samarbeider med forfattere og fagmiljøer i hele landet, og våre viktigste produktområder er:

Læremidler for høyere utdanning
Fagbøker for profesjonsmarkedet
Vitenskapelig publisering

Forlagsredaktør for denne utgivelsen:
Lasse.Postmyr@tapirforlag.no

Tapir Akademisk Forlag
7005 TRONDHEIM
Tlf.: 73 59 32 10
Faks: 73 59 32 04
E-post: post@tapirforlag.no
www.tapirforlag.no

Forord

Norsk informatikkonferanse har vært arrangert hvert år siden 1988 ved de forskjellige universitetene og høyskolene i Norge. I år er turen for første gang kommet til Gjøvik. Formålet har hele tiden vært det samme:

- å være et møtested for datafaglig ansatte og studenter i høyere undervisnings-institusjoner, forskningsinstitutter og næringsliv,
- å gi et representativt bilde av hva som foregår av forsknings- og avansert utviklingsarbeid i hele det norske fagmiljøet, dvs forsøke å vise bredden i forskningsmiljøet, både faglig, organisatorisk og geografisk, samt
- å holde et høyt faglig nivå og åpne for debatt om emner som opptar det data-faglige miljøet.

Årets innsendte bidrag holder et normalt nivå, hvor ca. 70% av bidragene ble akseptert. Det er studentarbeider blant de aksepterte bidragene og disse får økonomisk støtte fra NIK for å komme til konferansen og presentere sine bidrag.

Programkomiteen har i tillegg til redaksjonskomiteen (NIK-styret) bestått av: Lars Ailo Bongo (Universitetet i Tromsø), Weihai Yu (Universitetet i Tromsø), Ali Alsam (Høgskolen i Sør-Trøndelag), Yngve Lamo (Høgskolen i Bergen), Magne Jørgensen (Simula Research Laboratory), Jan Roland Olsson (Høgskolen i Østfold) og Olaf Hallan Graven (Høgskolen i Buskerud).

I vurderingsarbeidet har programkomiteen også fått uvurderlig hjelp av frivillige faglige konsulenter, nemlig Torleiv Kløve, Anya Helene Bagge, Jari Vanhanen og Nils Fjeldsø. På vegne av redaksjonskomiteen vil jeg gjerne få takke disse.

I likhet med tidligere år har ikke NIK opphavsrett til artiklene i denne samlingen. Den enkelte forfatter må kontaktes om rettighetene til eventuell videre gjengivelse av artiklene.

I år inviterer vi Andrew McGettrick fra University of Strathclyde, som også er Chair of the ACM Education Board and Council, som keynote til å fortelle oss om arbeidet og utfordringene knyttet til ACM Computing Curricula.

Artiklene finnes også på nettet på NIKs adresse http://www.nik.no der man også finner annen informasjon om NIK.

Programkomiteen for NIK 2010

Frode Eika Sandnes

Oslo, oktober 2010

Innhold

All closest neighbors are proper Delaunay edges generalized, and its application to parallel algorithms
Arne Maus, Jon Moen Drange 1

Formal Verification of a Cooperative Automatic Repeat Request MAC Protocol
Xin He, Ram Kumar, Liping Mu, Terje Gjøsæter 13

Monitoring and Control of QoE in Media Streams by Click
Bjørn J. Villa, Poul Heegard 24

Groovy and Grails meets Eclipse Modelling Framework
Kent Inge Fagerland Simonsen, Florian Mantz, Alessandro Rossini, Adrian Rutle 34

Prosjekt Ymer, grafikkprosessering på sky
Linda Granstad, Kyrre Begnum 44

Controlled Experiments on Pair Programming: Making Sense of Heterogeneous Results
Reidar Conradi, Muhammmad Ali Babar 56

A semantic parser and proofreading tool for all languages based on a text repository
Kai A. Olsen, Bård Indredavik 66

Teaching Requirements Engineering - An experimental approach
Asbjørn Danielsen 77

Game Design for Computer Science Education
Simon McCallum 87

Teaching Computer Sceience Students with Asperger's Syndrome
Kirsten Ribu 99

UNSTUKK – Utpreget iNteraktivt STUdieteknikK Kurs
Frode Haug, Øivind Kolloen 112

Etterspørsel etter viderutdanning i informatikk: Interessen for ulike fagemner
Lars Groth 121

Modelling Video-Quality Shaping with Interpolation and Frame-Drop Patterns
Håvard Berge, Mauritz Panggabean, Leif Arne Rønningen 132

Resampling HD Images with the Effects of Blur and Edges for Future Musical Collaboration
Mauritz Panggabean, Leif Arne Rønningen 144

Multi Exposure Image Fusion
Ali Alsam 156

Rendering Leaves using Hardware Tessellation
Jo Skjermo 162

Evaluation of Geometric Depth Estimation Model for Virtual Environment
Puneet Sharma, Jan H. Nilsen, Torbjørn Skramstad, Faouzi A. Cheikh 166

Improving user interaction with image collections
Christian Hartvedt 178

Towards calibration-free geo-localization of stationary outdoor webcams
Frode Eika Sandnes 190

Teaching Requirements Engineering
An experimental approach

Asbjørn Danielsen
Faculty of Technology
Narvik University College
Narvik, Norway
asd@hin.no

Abstract

The software industries use of Requirements Engineering (RE) is obstructed by the industries relatively poor understanding of RE practices and benefits. Teaching RE at university level is in this context an important endeavor.

This paper reports on an experimental approach on teaching RE and the results recorded during two years at Narvik University College. The course described in this paper uses experimental pedagogy to give the students experience in close to real-life work environment, demonstrating social and problem complexity of RE. The course uses social simulations rather than software simulations, making the students learn through interactions with real people and confronted with the complexity of social relationships.

Students, participating in the course, where divided into two groups. The first group had no previous experience with software development or programming languages. The other group was familiar with both object orientation and programming.

Introduction

The use of RE in the industry is often colored by a relatively poor understanding of RE practices and their benefits. Due to this, RE education is an important endeavor that can improve the RE methods used in businesses. In an ideal world, a course in RE should be provided at university level, before the students become software developers. This is however not so in most cases [Ber05]. When they do, these courses are often given in the traditional lecture/exercise format, and only few reports exist on other types of pedagogy used in such courses, i.e. [DAC05], [MM05], and [RGW08].

It is a common understanding that requirements represent the expression of peoples desires [GW89]. To understand the desires of people and to be able to express these desires is essentially a social construction. Consequently, much social wisdom is packed into RE methods. Further on, it is unrealistic to expect students with little organizational experience to understand this body of knowledge and to appreciate even the need for RE methods, much less to be able to use them. In the software engineering discipline, it is essential that software engineering students understand the latest accepted methods and practices in use today in the design of complex computing systems.

If we want to provide more than just a shallow understanding of RE to the students, we need to provide more than just lectures about RE methods and academic problems where they can exercise the knowledge they have been provided in our traditional teaching environment. It would also be good if we could define business experience as a prerequisite or co-requisite to an RE course. Unfortunately this is not an applicable option. To compensate for this situation, a curriculum in which students do internships in the middle of their studies may be an option. However, even this is no guarantee that they will be exposed to the experiences that they will need to more fully appreciate and understand an RE course.

The experiences we seek to impart to students are directly linked to the issues found in the workplace when we understand what the business is about and what the desires of

This paper was presented at the NIK 2010 conference. For more information see http://www.nik.no/

people are. A short list of these issues is: dealing with ambiguity, uncertainty, confusion, fear, time pressure, collaboration, corporate politics etc. In sum, what some call the "messy" part of organizations [CH97]. The messy part, recognized by scientists and mathematicians as wicked problems, exemplify the differences between classroom and workplace problems.

To be able to handle the messy part, one needs to combine both emotions and techniques in one combined effort in RE. Whereas the use of specific techniques and algorithms can be learned through lectures and exercises, emotions can only be learned through real life experiences.

This paper describes the requirements engineering part of a System Development course delivered at Narvik University College. In this paper the reasons for creating the course, its pedagogical features, and the experiences giving the course will be presented. The course is a 10 point course, spanning from spring to autumn, and is now running in its second year. This paper mainly describes the experiences during spring 2009 and spring 2010 in which RE is the main focus. The students in the course were divided into groups based on previous experience; groups with no previous experience of software development, and groups with previous experiences of programming and OO.

The course was a result of a major restructuring of the computer science education program at Narvik University College. It was designed to create a realistic organization in order to provide the students with an opportunity to experience the "messiness" they can expect in the workplace. Situations which one may be faced with in a real workplace was created, and framed the problems presented to the students in an uncertain and confusing reality, often relying only on verbal, word-of-mouth communication as this is an important part of design and management information transfer in reality.

The system the students were working on was a stock-exchange game using close to real time data from the Oslo Stock Exchange combined with locally defined stocks. The approach was to immerse the students into a more realistic social environment with real data instead of simulated data, real events instead of simulated ones, interviewing real people, using real planning and reporting tools.

In the next section of this paper a number of typical RE education related challenges will be presented. Further on, the basics of experimental learning, and how these theories was adopted into the course is presented, and finally our own experiences with the course including students evaluation before presenting related work and summarizing the contribution to the area of expertise.

RE challenges

Traditional education tends to be separated into separate disciplines or courses. Few, if any, courses seek to integrate disciplines. The result is that students going through the educational system acquire much factual knowledge about specific areas but little synthesis is provided. This pattern is also present in courses where each course is divided into topics which is lectured in a sequential manner, one by one, with the implication that there is little, if any, relationship between topics, and without any or just a few references to other disciplines. We, as university teachers are unwittingly creating a student mindset that partitions out methodologies. This observation became evident during the course when it was discovered that students that had previous experience with OO and programming had a more difficult time grasping the full context of RE than the students with no programming experience. This observation is further discussed in the Instructional Scaffolding section.

Another challenge is related to wicked problems and the fact that these problems tend to be difficult to define, and the definition, if it exists, change over time. Their solution is not known at the beginning and whether they were correctly solved will not be known often until long after a solution is proposed or implemented. Solving one problem often brings about a number of other problems that could not be foreseen before the solution was implemented, and so on. Wicked problems have the following properties [SMB97]:

- cannot be easily defined so that all stakeholders agree on the problem to solve
- have no clear stopping rules
- have better or worse solutions, not right and wrong ones
- have no objective measure of success
- require iteration – every trial counts
- have no given alternative solutions – they must be discovered
- require complex judgments about the level of abstraction at which to define the problem
- often have strong moral, political or professional dimensions which cannot be easily formalized.

Finally we have the differences between the classroom experience and the real time workspace. [RGW08] presents a number of differences between the classroom and the real-life workplace experience. These differences impose a number of challenges that need to be addressed as well.

In fact, RE may be considered a meta-discipline in that it integrates a number of other disciplines, e.g. ethics, psychology, organizational theory, sociology, software engineering, etc. RE also demands the students to be creative, using techniques like brainstorming, metaphoric thinking, analogical thinking, idea sketching, pragmatic thinking, and other techniques used in more mature areas of engineering design.

The properties of wicked problems are such that every trial count and it is neither possible nor advisable to give the students the rules of business. They have to live through it. The kind of approach towards learning and the pedagogy used is challenging for many students since it does not fall into the previous well-defined form of getting all relevant information up front making the student capable of efficiently solving the problem presented by the teacher.

In our experience most students fail to see the point of using a lot of time learning the business rules and understanding the business requirements. They live under the false impression that if anything is unclear, someone will tell them what is unclear and provide a solution. The students resist living through the problems related to changing requirements, unclear requirements, unfinished requirements, and resist asking questions that may result in more problems.

Experimental learning

The theory of experimental learning is generally attributed to Kolb [Ko83]. Kolb developed this theory based on the previous pioneering work of educators such as Dewey, Lewin, Piaget, and Freire. He proposed a model of the underlying structures of the learning process based on research in psychology, philosophy, and physiology, and bases its typology of individual learning styles and corresponding structures of knowledge in different academic disciplines and careers on this structural model.

His model of experimental learning is divided into four modes, usually assembled into the experimental learning cycle presented in Figure 1.

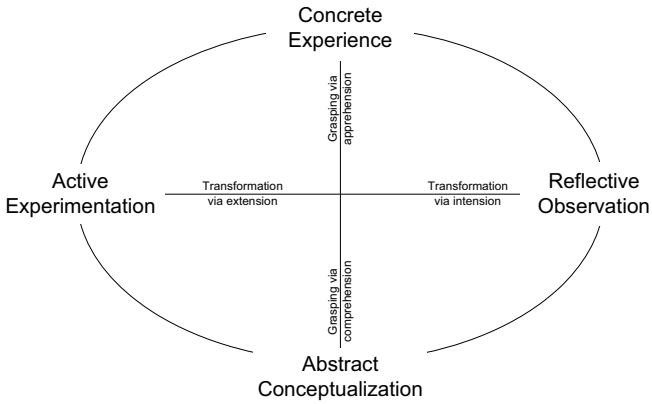

Figure 1. Kolbs model of the Experimental Learning Cycle [Ko83].

According to Kolb, the learners "must be able to involve themselves fully, openly, and without bias in new experiences (CE). They must be able to reflect on and observe their experiences from many perspectives (RO). They must be able to create concepts that integrate their observations into logically sound theories (AC), and they must be able to use these theories to make decisions and solve problems (AE)."

Kolb notes that this cycle is an ideal that is difficult to achieve because learners cannot easily reconcile these modes, which require different ways of interacting with one's environment and thinking about it. Further on, in the model, the modes are "dialectically" opposed along two dimensions. The first dimension, prehension, opposes Concrete Experience of events (apprehension) and Abstract Conceptualization that seeks to make generalizations of these events (comprehension). The second dimension, transformation, opposes Reflective Observation about experience (intension) and Active Experimentation that seeks to make decisions about future experience (extension). For Kolb, the level of learning is determined by the way the learner can resolve the conflicts present in these two dimensions.

Another important aspect of experimental learning has been described by Lev Vygotsky as the zone of proximal development (ZPD). ZPD is the difference between what a learner can do without help and what he or she can do with help, and is defined as "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance, or in collaboration with more capable peers" [Vy78].

As already described in the previous sections, RE consists of a number of wicked problems that need an iterative process to be gradually refined. During the process, the students should become more familiar with the problem domain in question, business rules and hopefully the students should be able to comprehend the nature of RE and business problems through their apprehension of the concrete classroom experience.

Instructional scaffolding is based on Vygotsky's theory where he defined scaffolding instruction as the role of teachers and others in supporting the learner's development and providing support structures to get to that next stage or level. An important aspect of scaffolding instruction is that the scaffolds are temporary. As the learner's abilities increase the scaffolding provided by the more knowledgeable is progressively withdrawn. Finally the learner is able to complete the task or master the

concepts independently [CCS02]. Therefore the goal of the educator when using the scaffolding teaching strategy is for the students to become independent and self-regulating learners and problem solvers. As the learner's knowledge and learning competency increases, the educator gradually reduces the supports provided until the students complete the tasks without interference from the teacher.

The Course

The System Development course was designed following the experimental learning cycle, incorporating the idea of instructional scaffolding. The course grade was defined to be Passed or Not Passed. First the class was divided into two distinct categories; those with no prior knowledge of software development, and those with some experience in OO or programming. This separation was done because it was expected that the student with no prior experience would need more assistance than the other group.

The typical course session consisted of two times two periods of forty-five minutes every week. In the beginning of the term, the course focused on initial analysis and Business Process Modeling Notation (BPMN). Then we started working with RE phase of the course. RE is at the interface between the business understanding and the product or service to be created, and the students need to understand the business function before they can start on RE process of conducting interviews, drafting requirements, etc.

A typical lesson in the RE phase would include a presentation of a number of real-life examples, e.g. business game or a RE interview, followed by a discussion on different social and technical issues of the examples. The lesson would then be followed by a shuffle of people within groups into teams of 3 – 7 people. The shuffling of people within the group was done to provide a kind of uncertainty related to people attending the group and to be able to reduce the predictability of the working environment for the students. Each team was given identical exercises in form of similar activities as those being presented during lecture. The exercise was usually in the form of a RE interview related to a limited number of problems within a system. Most experience sessions where followed by a 45 minute reflective observation phase in which emotional and technical debriefing was performed. After the debriefing was performed, the groups exchanged documents developed during the process and commented on the other groups work. To finish the session off, a lesson to sum up the experiences with the exercise was held. The typical week doing RE-lessons looked like:

4	Summary lecture and discussions	Session 1
1	Introducing a RE subject, exemplifying with real-life examples	
2	Shuffling group-members into work-groups Perform exercise, create RE descriptions	Session 2
3	Reflective Observation Phase Emotional and technical debriefing Change of RE work between groups	

After having lectured after this model for five weeks, a new shuffle of the members of the groups where performed creating a project group which should be stable for the rest of the term. Project management techniques were lectured and the class was at a stage in which they could embark on a journey into doing a larger RE exercise resulting in the requirements for a stock-exchange game, and using instructional scaffolding in

the process. Each of the teams was organized as different companies, competing in the same market, developing competing systems.

The system to be developed was described as an internet-based game which should be as close to real-life experience as the real stock exchange. The game period should be limited to a number of months after startup. All buying and selling within the game should be done using virtual money, and should include actual stocks on Oslo Stock Exchange (OSE), real-time stock rates and availability. The game should also support virtual companies and their stocks which also should be tradable. All selling and buying should be done using a broker. Further on the system should incorporate some kind of security to minimize the problem that a single user may hold several identities within the game, making it possible to do illegal trades.

All groups got the same information. Within the first two weeks, the students had to deliver a business game document on how stock trading is performed in Norway, different techniques of stock trading, which stock types exist and what is the difference, what laws apply, and to some extent give an explanation on a number of financial and economical terms used in stock trading. They also had to deliver the first project plan, and their first project description.

As a result of severe time pressure, each group struggled to understand the problem domain, describe and document it, and create a project plan. We observed that in most cases the students neglected the text given in the exercise, and that there was no overall picture on what was going on. Frustration and emotions surfaced quite often, and some students even expressed doubts about the competence of the lecturer. The planning aspect was put out until last minute and some groups didn't even manage to come up with a plan. After this exercise, an emotional and a technical debriefing were performed, evaluating the results. Then all groups exchanged business game documents, and commented on differences.

This period was then followed by a 7 week module focusing on RE and specifications. At the start of this period the students got a document from the company's sales- and marketing departments about sales and marketing problems, and the teams was instructed to deliver a simplified Software Requirement Specification (SRS) using BPMN as modeling notation where applicable. They were also instructed that further information was available from the CEO (lecturer), CIO (student assistant #1), customer (student assistant #2) as well as one outside stakeholder (lecturer imposing as an advertising interest). A document was prepared for each role in the game, ensuring that the same main messages and information were delivered to all groups during interview. During the whole process, except during interviews, the lecturer performed instructional scaffolding as described.

After the first round of interviews, each group wrote its own simplified version of the SRS and presented it to the CEO of the company. The CEO did not approve the SRS, and the students was instructed to do further investigations, especially into the feasibility and validity part of their SRS. The students concluded they needed more interviews and a second interviewing session was set up.

At the completion of the 7 week module, a new and updated version of the SRS where delivered to the CEO, which this time accepted and approved it. This document was later used to design and implement a prototype of the system.

Experiences

The RE-part of the course was given two spring-terms with 15 and 18 students in each year. The course itself was mandatorial for all students in the Bachelor of Engineering study of Computer Science at Narvik University College.

The course has been well evaluated during mid-term in both years. The first time the course was performed, all attendants had completed a number of OO and programming courses, but in 2010 about half of the class had no prior experience or knowledge of OO.

Course rating

In both years the course has been evaluated by the students. The overall evaluation of the course is presented in Figure 2.

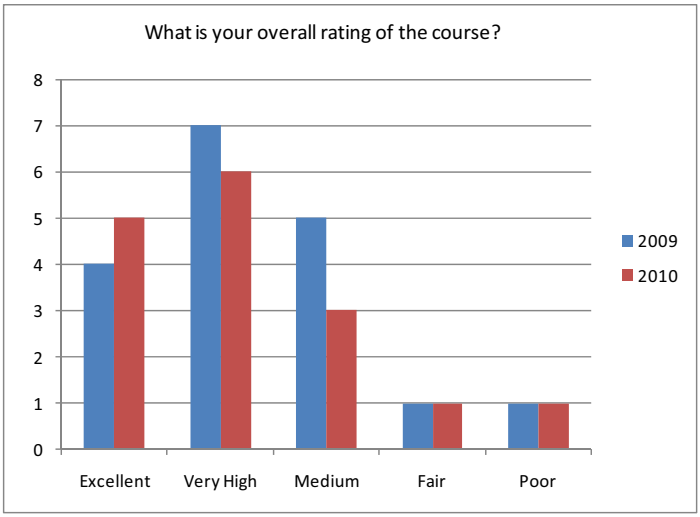

Figure 2. Overall rating of mid-term course by students in 2009 and 2010

Even though the course itself is fairly good rated by the students, there is room for improvements. The following areas have been recognized for improvement.

Learning Style

The course is problem-based and the approach towards learning is completely different from other courses at campus. This had to be explained frequently during the course. At the beginning of the course several students found the course to be disorganized and found it difficult to accept this kind of pedagogy. However, at the end of the course most students found the problem-based approach to be fruitful. No students quit the course.

Some students found the debriefing sessions boring and cumbersome due to the fact that most groups had the same or similar findings and experiences. This problem may be solved by having the debriefing sessions of individual groups outside the classroom, and later make a synthesis of the findings.

Group dynamics

Even though the work was done in groups, the most common scenario was that two or three persons in the group executed the work while the rest observed. It was also evident that when all members of the group worked on separate parts of the project, they did not communicate well with the other parts of their group working on related material. This resulted in incompatible models and specifications. The latter situation occurred more often during time pressure.

Frequent reporting on time used and progress had the effect that all participants in the group became more conscious of what they were doing, how much time they had used and how unsecure they were about how much time had to be used before they could complete the task at hand. This situation also made the students more aware of how much the other group members had left of their work.

The frustration related to estimating the remaining work, combined with some degree of timeboxing of activities resulted in emotional outbursts and the need to frame somebody for the situation (most usually the teacher). It seemed to have the effect that the group united and worked together more closely towards a single goal.

Each student would get a grade in the course as Passed or Not Passed, dependent on the work of the group, how much the student had participated in the groups work, and how the student handled other wicked problems related to RE. Some students found this to be unfair due to the fact that some individuals contributed more than others. This challenge may be solved by changing the grade to an A-F scale, making the completion of the project mandatorily and having a final oral exam.

Credibility

It is of utmost importance that a course like this which differs drastically from other university courses has credibility. As the course puts the student in a stressful situation, the student might infer that the course is disorganized. With adequate communication, the students eventually believe that the course puts them in a situation related to the "real life" situation and learn from their experience.

A few skeptical students challenged the approach on learning and expressed doubts about the competence of the lecturer in the first weeks of the course. The challenge was expressed as comments during exercises, and as criticism during debriefing sessions.

The following strategy was used to raise the credibility of the course:

- Used own industrial experience as examples
- Used real-life project examples to demonstrate problems
- Open discussions
- Open scenarios based on student experiences

It is preferable to have guest speakers from the software industry to do some talks on different aspects of RE. It would also be preferable that more lecturers shared a course like this, partly because of workload, but also by increasing the industrial experience and by so broadening the horizon of the students.

Instructional Scaffolding

Initially it was expected that the students which had no previous experience with OO would need more coaching and assistance during the RE-phase than the group with such experience. This proved to be wrong. The student groups that had OO competence and programming experience tended to delve into descriptions on how some problem should be solved on an object-level rather than specifying what problem should be solved, under which circumstances, consequences, describing processes and so forth. This resulted in a situation where the lecturer repeatedly had to re-establish RE goals in the group, guiding the group out of its OO-sphere and back into the game.

The groups with no previous OO experience handled RE differently. They focused on the business processes, describing these, doing their interviews, refining their documents and worked better doing RE than the other project groups.

Resources and results

Doing a course like this require more planning and more resources than a standard course. It is also more difficult to teach since it involves wicked problems and consequences of such problems. The debriefing sessions has to be performed to adjust the students perspective of RE, and they also need to be done to focus the summary lessons on the important aspects. Consequently, the course uses more man-hours than other courses.

The effect of the course is evident in the student evaluations of the course, and it is also possible to see the effect in B.Sc. theses of students completing the course. The B.Sc. thesis of students who has completed the course seems to have a more balanced approach towards software development in general, and RE in specific. The main component contributing to this is most likely the experience each student gets by working with wicked problems in the course discussed.

We think we have succeeded with providing a context in which the students may place their functional knowledge obtained during their studies. Further on, we have indications that the course has prepared students for the workplace experience. One former student testimonial said " … the course was difficult to comprehend at first, and at times it was difficult to understand what was expected of us as students. I am now working as a software engineer, and I realize how unclear and confusing problem statements and goals may be. I still find the situation uncomforting, but I feel confident that I am able to handle the situation and find solutions".

Related work

There seems to be few publications on experimental pedagogy related to RE. Some publications were published at the REET 2005 workshop, e.g. [Be05], [DAC05], and [MM05]. [DAC05] discuss challenges encountered teaching across universities, different cultures and time zones, while [MM05] focus on an investigation-based approach towards RE. [RGW08] presents an immersive and problem based approach closely related to instructional scaffolding. It further on use low-tech and social simulations rather than computer based. Consequently, [RGW08] is an approach which in many aspects resembles the approach presented in this article.

Conclusions

In this paper an experimental approach toward teaching RE has been presented with its theoretical foundation. The course was designed with two major objectives in mind; prepare the students for the real-world experience in which they will embark after a completed study, and, create an arena in which the students may place their functional knowledge obtained during their studies.

References

[Be05] Berenbach B., 2005, "A hole in the curriculum", *Proceedings 1st International Workshop on Requirements Engineering Education and Training (REET 2005)*, http://re.cti.depaul.edu/REET05/P2_Berenbach.pdf

[CCS02] Chang K., Chen I., and Sung Y., 2002. "The effect of concept mapping to enhance text comprehension and summarization", *The Journal of Experimental Education 71(1)*, pp. 5-23

[CH97] Checkland P. and Holwell S., 1997,"Information, Systems and Information Systems – making sense of the field", John Wiley and Sons Ltd, ISBN: 0-471-95820-4

[DAC05] Damian D, Al-Ani B., Cubranic D, and Robles L, 2005, "Teaching Requirements Engineering in Global Software Development: A three-University Collaboration", *Proceedings 1st International Workshop on Requirements Engineering Education and Training (REET 2005)*, http://re.cti.depaul.edu/REET05/F6_Damian.pdf

[GW89] Gause D.C. and Weinberg G.M., 1989, "Exploring Requirements: Quality Before Design", Dorset House Publishing, ISBN: 0-932-63313-7

[Ko83] Kolb D.A., 1983, "Experiential Learning: Experience as the Source of Learning and Development", Prentice Hall, ISBN: 0-132-95261-0

[MM05] Madhavji N.H. and Miller J, 2005, "Investigation-based Requirements Engineering Education", *Proceedings 1st International Workshop on Requirements Engineering Education and Training (REET 2005)*, http://re.cti.depaul.edu/REET05/P3_Madjhavi.pdf

[SMB97] Shum S.J.B, MacLean A., Bellotti, V.M.E., Hammond N.V., 1997, "Graphical Argumentation and Design Cognition", *Human-Computer Interaction, 12(3)*, pp. 267-300

[RGW08] Regev G., Gause D.C., and Wegmann A, 2008, "Requirements Engineering Education in the 21st Century, an Experimental Learning Approach", *Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*, pp. 85-94

[Vy78] Vygotsky L.S., 1978, "Mind in Society: The Development of Higher Psychological Processes", Harvard University Press, ISBN: 0-674-57629-2, p. 86