Role play in HCI studies
Kristina Moroz-Lapin
Vilnius University
Naugarduko 24, LT-03225 Vilnius, Lithuania
+370 5 219 3067
kristina.lapin@mif.vu.lt

ABSTRACT
This paper discusses the education of human computer interaction from the perspective of role play activities. In one case, role play is used in team projects where students are encouraged to act as persons with certain responsibilities and motivations. In the second one, roles play can be observed in discussion and debates format that is used in the undergraduate classes and graduate seminars. The multidisciplinary engineering, science and social underpinnings of human computer interaction challenge mostly technical-oriented software engineering students. In our opinion, social aspects are better assimilated by the students when teaching methods established in social science education are used. We employ these methods in traditional lecture-based teaching format and large student classes. The aim is to increase students’ involvement in seminar subject where active participants are both speakers and listeners. In our opinion, the role playing activity in the project work, discussions and debates intersperse engineering studies.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education – Computer Science Education.

General Terms
HCI education, teaching methods, social aspects.

Keywords
Roles play, discussions, debates.

1. INTRODUCTION
According to Compact Oxford English Dictionary, the first meaning of the word „play” is „games and other activities engaged in for enjoyment”. In the education and training we can consider some aspect of play, namely role play. Role play is „the acting out of a particular role, either consciously (as a technique in psychotherapy or training) or unconsciously (in accordance with the perceived expectations of society).”[6]

In this paper we are examining how role play activities are employed in the teaching of Human Computer Interaction (HCI) courses in Vilnius University, Lithuania.

Human Computer Interaction course involves engineering, science and social aspects. Engineering aspects are usual for technically-oriented computing students. Students mostly positively accept science aspects as well such as HCI principles, models and theories that accumulate the experience of the development of user interfaces. But social context of use is often a formidable part of the course.

Human Computer Interaction is delivered in Software Engineering (SE) study programme on the undergraduate and graduate levels: Human Computer Interaction (HCI) for undergraduates and Human Computer Interaction Design (HCID) for graduates.

The primary obstacle instructor faces in teaching usability goals for undergraduates includes students’ frequent attitude that the newest technology is the most important factor for the project success. This attitude discourages students from doing the deep user needs and usability goals analysis that they are expected to do at the beginning of the semester. When the user needs and usability goals specification was the first assignment, more than half of students crop had serious difficulties. As a result, the grades for the first assignment were slightly lower comparing to the following more technical assignments such us prototype building.

Trying to increase the students’ interest and comprehension for the context of use and allowing them to gradually start thinking from user’s perspective, the list of course assignments was updated. Assignments where students play a user role were included in the beginning of the semester. Reduced gap between grades that student teams received for technical and non-technical assignments shows that course updates have positive impact. Before the first user needs and usability goals specification we included homework in which inconvenient interfaces are presented. Two classes at the beginning of semester are devoted to inconveniences they faced using mobile phones, lifts in big shopping centres, microwaves, web pages, programming tools and other interactive products.

Students act as users when they test peer’s project that is one of the evaluation assignments. This assignment consolidates the first user’s thinking experience. Students observe once again that reason of user problems is not the lack of competence but design mistakes. They are often surprised that they, in the near future software engineering professionals, encounter problems with their colleges’ projects.

During the semester undergraduate students develop and evaluate prototypes. Software engineering students are trained to work according to structured procedures that are defined in particular
software engineering life cycle. They also value tools and implementation. Despite the course requirement that first prototype should be low-fidelity mock-up, part of students prepare almost finished interfaces. Tests results often require to make significant changes and this experience once again show that high-fidelity interface at the beginning of project was not a right decision.

The graduate Human Computer Interaction Design (HCID) course has been introduced since 2006. We gained experience with two student crops until now. The course is organised in lectures and seminars format. ACM SIGCHI Curricula for Human-Computer Interaction suggests several teaching approaches that support active student involvement in learning HCI issues.[1] Among others, discussion groups and debates on design trade-offs are mentioned. Besides the known benefits of debates, pro, contra or audience role play vary the tiring analysis of case studies while considering the controversial aspects.

Further we present our experience of teaching HCI from the role play perspective in both undergraduate HCI and graduate HCID courses delivered in Vilnius University. We analyse undergraduate and graduate courses from role play perspective. Finally, we summarize the experience of using role play in project work, discussions and debates.

2. ROLE PLAY IN TEAM PROJECTS

HCI is delivered in the 5th semester to about 90 students. Semester lasts 16 weeks: 2 hours of lectures and 2 hours of classes per week. This course is integrated with Team Software Process course. The lectures of both courses are separate. Integrated are the assignments [6]:

- the project task is provided by industry partner that participates in team projects activities as the customer;
- HCI deliverables cover user needs and usability goals, interface requirements, low and high fidelity prototyping and usability evaluations;
- Team Software Process deliverables covers other software engineering activities.

Students have to perform a project work that simulates usability engineering activities. Assignments cover requirement analysis, prototyping, implementation and evaluation. Students perform a project, divide up the roles and responsibilities and discuss the results. During the project they act the roles of analyst, developer, tester, expert reviewer and user for the peer team project. Diversity of roles during the project allows to avoid the tiring experience while performing course assignments.

The final project must due to the end of class. Project outcome is the complete system with user documentation. Programming can be done in any language or system. Final project is defended in public oral presentation, in which students, HCI and Team Software Process lecturers as well as industry partner participate.

Initially, the course started with user need analysis and formulation of usability goals. This assignment was perceived by many students as the most difficult. This impression was confirmed by grades. We compared the grades that the same team with the same motivation received from technical assignments such as building prototypes with less technical assignment, user needs and usability goals specification. Figure 1 shows that 66% of students received lower grades from this assignment. Identification of users’ problems, future system vision and usability goals in worst cases were formulated quite artificially. Real interest and engagement was achieved rather by modest number of students (34%). The course in contemporary setting was perceived as low related to their professional competence by almost 20% of students.

<table>
<thead>
<tr>
<th>Comparison of marks for technical and non-technical assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better marks</td>
</tr>
<tr>
<td>34%</td>
</tr>
</tbody>
</table>

Figure 1. The proportion of grades received from less technical and technical assignments

Trying to encourage the appreciation of the value of context of use classes have started with homework where students presented the inconvenient design examples they face in their everyday activities. Discussion in the class concentrated on how to improve irrelevant solutions [2]. They are not just only shown, but also elaborated from various perspectives. Their examples concern not only software but also other products, such as lifts, copiers, control devices in the cars, etc. These presentations stimulate discussions about further consequences that arise for the user while using these devices. This assignment motivates students to rethink obvious solutions and reflect them from various perspectives.

During the discussions we observed that students were often surprised when they saw acceptable solutions from their point of view whereas the colleagues asserted them as inconvenient to use. They realised that even their peers perceive interfaces in different ways.

In order to examine what effect took course modifications we compared the grades students received from technical prototype development and less technical assignments. We have to concede that overall amount of lower grades decreased only in 1%. It means that assignment related to social aspects remain more complicated than other.

But our study indicates also positive change: the range of lower grades decreased significantly (see Figure 2). From usability goals assignment nobody received more than 20% lower grades comparing with technical assignments. In earlier course (in 2007) even 21% of students received more than 20% lower grades comparing with grades they received from technical assignments. Majority of lower grades (61%) was only slightly lower than grades from technical assignments. As we can see in Figure 2 the 10% lower grades concern only 4% of assignments. These results are slightly better than earlier 57% of grades that were more than 10% lower. In our opinion, these results also show the changed attitude.
In most cases the first low-fidelity solutions were elaborated to high-fidelity design throughout the project. Though results were usable enough, they also showed up creativity gap between the definition of usability goals and prototyping. [7] Usability tests performed by peers indicated the value of using low-fidelity prototypes at the beginning of the project.

In both years the grades were given using the same assignment evaluation form with the same grading criteria. Yet maybe comparison of two years may not be very reliable from the strict scientific point of view. Because of constant modifications course in current form has been taught for two years. Last year modifications related only homework and peer’s tests. Earlier courses were not integrated with Team Software Process and assignment structure was different. Therefore grades from earlier years were excluded from the study. It would be hard to assert which modification caused the grade change.

In the course evaluation questionnaires 92% students asserted HCI course as rather important and important source of their professional competences. We assert that acting as a user is used.[4] It combines an introduction, case materials, directive and nondirective class discussion. In the first case instructor asks probing questions and students analyse the problem and try to provide answers. In nondirective discussion instructor starts with minimal introduction and then acts more as a facilitator than a dominant questioner. During seminars mostly a middle approach is used.[4] It combines an introduction, case materials, directive but not dominating questioning. The class ends with an appropriate summary.

In our seminars students from the audience interacted with speaker during and after the main presentation. They confirmed or contradicted the presented attitude. Instructor acted here as a discussion facilitator. In those successful seminars we were very close to discussion format.

In our seminars students are also encouraged to suggest topics of their interests for both one of two debates that take place at the end of semester. Students in HCI course are expected to plan and hold a briefs on both sides of the issue and are prepared to argue certain or both sides. In the second case the side for student teams is opposed views are evident. Two teams of students prepare written opposed views are evident. Two teams of students prepare written briefs on both sides of the issue and are prepared to argue certain or both sides. In the second case the side for student teams is assigned by the instructor at the beginning of a class. After the pro and con teams present their arguments, each team have rebuttal right. The audience can question both sides and evaluate the content and presentation of both teams.

In our seminars students are also encouraged to suggest topics of their interests for both discussion and debate. A course reading list suggests the main evaluation questionnaires 63% of students expressed the opinion that it would be better to make seminars more interactive.

HCID course is devoted to deepen the analysis phase. This phase involves understanding of users, conceptualising the interaction, creating personas, choosing appropriate interaction paradigms and then formulating user needs and requirements. Most topics are non-technical in their nature, for example development of on-line communities, virtual worlds, embodied interface agents and others. Software engineering students run into difficulties dealing with them. Analysing the experience gained from using more or less successful examples, students try to ascertain the factors that influence their viability.

Social aspects suggest using teaching methods already established and proven out in education of social sciences. We faced a great variety of teaching methods used in teaching social domains [4]; discussions, debates, public hearings, trials, problem-based learning, scientific research team and team learning. This fact motivated us to explore these methods because they are unusual for software engineering educators.

Traditional social disciplines like business, law and medicine educations have a long tradition of using techniques that involve learning by doing, develop analytical and decision-making skills and improve oral communication as well as team work skills. For engineering students it would be difficult to follow public hearings, and trial formats, because of the lack of skills about detailed of these procedures. The discussions, debates and problem-based learning seem to be most appropriate because of simple process and ability to involve more students from the audience.

Discussion technique ranges from strong questioning to nondirective class discussion. In the first case instructor asks probing questions and students analyse the problem and try to provide answers. In nondirective discussion instructor starts with minimal introduction and then acts more as a facilitator than a dominant questioner. During seminars mostly a middle approach is used.[4] It combines an introduction, case materials, directive but not dominating questioning. The class ends with an appropriate summary.

In their studies students are also encouraged to suggest topics of their interests for both one of two debates that take place at the end of semester. Students are also encouraged to suggest topics of their interests for both discussion and debate. A course reading list suggests the main

3. DEBATES IN GRADUATE SEMINARS
In this section we analyse playing the pro and contra roles on the debate in graduate seminars.

HCID course has been introduced for masters since 2006. The course is delivered each year for approximately 50 students. Master courses in the Faculty of Mathematics and Informatics of Vilnius University are taught in traditional lectures and seminars format. Despite its engineering nature, HCID course deals with social aspects that need another teaching approach as we realised delivering course for the first students’ crop.

In the first year seminars students analysed the examples and applied facts and principles to small assignments. A part of seminars was devoted to explore the research papers that dealt with innovative designs and interesting interaction design cases. In many cases the presentations did not encourage discussions and seminars were tiring lecture continuation. After the course in the
We observed that debates engage to discussion not only the pro and contra student teams. Audience also takes active part by asking both sides. At the end of a class they are asked to write what position and arguments were more convincing and whether they changed their attitude. The discussions start just from the beginning and are more intensive than on other seminars.

The grading is done evaluating the quality of preparation, amount and quality of involved sources as well as engagement of the audience (low, medium, high). After second course delivery we can conclude that discussion quality mainly depends on the quality of preparation. Most of master students in Lithuania already have several years of working experience. Apart the publications they augment the materials with examples from their professional experience. This experience is also asserted as a very valuable source of information.

We have had various seminars: more active and less active. But debates all were active and intensive. Pro and con role play improves the discussion and overall seminar quality. After the course students commented that despite the fact that debates require more preparation this form is more engaging than other seminars. Students stated that topics discussed on debates were easily recalled on exam. Other topics remained not so markedly.

4. CONCLUSIONS

In this paper we aim to illustrate how role play addresses the comprehension of topics related to social aspects. These topics are asserted as obscure by computing students because they are trained to use structured methods in analysis and development. We have presented our teaching approach for undergraduate and graduate HCI courses in software engineering curriculum.

Role play activities are useful in appreciating the social aspects of computing. Acting as a user before the software development activities encourage thinking from users’ perspective. The introduced discussion on irrelevant design examples and the ways how to make them good improved the comprehension of the user needs and usability goals. Discussion about inconvenient interfaces is perceived by students as an amusing activity. Earlier the first assignment was user needs analysis, followed by definition of usability goals. This assignment was asserted by students as complicated because it was not easy to start thinking from user perspective without any preparation.

The user needs assignment requires sudden transition from usual developer’s perspective to user’s point of view. The additional homework is a more gradual way to immerse to user’s view: we suggest acting as a user, trying to detect the inconveniences in the already used interactive devices. These examples also encourage the discussion how to improve inconvenient solutions from user’s perspective. At the beginning of semester students do not have enough skills to provide more competent argumentation for their solutions. When personal use cases are discussed on the seminar it is easier to further specify the potential user problems and vision of the future product that should address these problems.

Role play in the undergraduate HCI course allows students to act as users also while testing peer’s projects. In other assignments they play a role of developers and programmers. We think that various roles allow them to avoid the tiring monotony in practical classes.

Debates and discussions in graduate course enrich teaching of non-technical HCI aspects. We think that techniques established in social sciences are useful for dealing with social aspects in the graduate HCID course. For example, after the debate about the viability of on-line communities, students discovered that technological part of on-line communities is a significant but not the most important factor.

Pro and contra role play encourage engagement not only the speaker teams but also the audience. The main conclusions made in debates were better recalled on the exam than in the other seminar forms. Debates aid the ability to present and defend personal viewpoints. In our opinion, the skills of non-technical aspects are better assimilated in this class format.

As a conclusion, we state that it is not easy to introduce playful activities in rigid and hard engineering studies. Assignment grades, exam results and student opinions suggest that such attempts are truly noteworthy and will be elaborated in the future.

5. REFERENCES


