Devising instruction from errors in students' assignments: a case in usability engineering education

Elena García-Barriocanal, Miguel-Angel Sicilia, Salvador Sánchez-Alonso, Daniel Rodríguez

Information Engineering Research Unit
Computer Science Dept., University of Alcalá
Ctra. Barcelona km. 33.6 – 28871 Alcalá de Henares (Madrid), Spain
{elena.garciab, msicilia, salvador.sanchez, daniel.rodriguezg}@uah.es

Abstract

Problem-based learning emphasizes the role of problem solving as the main driver of instruction, thus critically relying on the quality of problem collections, which should exercise especially those aspects students find difficult to master. In areas in which extensive and tested problem collections are not available, it is useful to analyze the mistakes of students in previous year's assignments as a source of evidence for the design of new problems or tutorial resources. This paper reports preliminary results of that approach as applied to heuristic usability evaluation in the context of an introductory Human Computer Interaction course. The process is described, and some examples are reported as concrete cases in which assignment analysis lead to insights for devising instruction.

1. Introduction

Problem-based instruction has been subject to significant research interest in recent years (Merril, 2007). Problem-based strategies share as a central concern the design of problems that cover the complete spectrum of knowledge and abilities to be mastered, even though they vary in the amount of learner support that is provided and the degree of collaborative activity required. Problems can be defined as "questions rose for inquiry, consideration, or solution". Instructors facing problem design for their courses attempt to craft problems to support competency development, thus trying to anticipate the "problems in solving the problems", i.e. facing students with exercising most of the potential difficulties they would find in performing realistic work activities. This can be approached from the experience and sound knowledge of the instructions about the topics to be taught, self-reflecting on potential difficulties and pitfalls. In some disciplines, it is easy to find problem collections in textbooks or on-line learning resources that can be used as a form of sedimentation of practical experience in proposing and evaluating problems. However, there are topics or disciplines for which few or fragmented problems collections are available, and in general problem collections do not provide hints on the kind of learning difficulties and detailed topic coverage of each of the problems they contain. In those cases, an empirical approach would be helpful to continuously improve and evolve problem collections. Empirical data on problems faced by students can be gathered from the assignments in previous courses. The careful examination of assignment solutions given by students is the best source of evidence for devising new problems or updating existing ones, if we assume that student cohorts are reasonably homogeneous from one year to the next.

Human computer interaction (HCI) is a multi-disciplinary area of study that is essential in the education of engineers dealing with the construction of man-machine interfaces (Rozanski and Haake, 2003). One of the key competencies to be acquired in HCI is usability evaluation. Usability evaluation requires in general analysis skills and critical thinking, as usability problems are often a matter of degree, and knowledge on usability is described in terms of generic rules (usability guidelines), principles or heuristics that require judgment and practice to be mastered. There exist various techniques for evaluating usability depending on available resources (time facilities and resources), evaluator experience, ability and the stage of development of the software under review (Hartson, Andre and Williges, 2001). It seems apparent that problem-based approaches to instructional design may be adequate for teaching usability evaluation. However, such approaches require a carefully devised set of problems that provide the required progressive scaffolding (Simons & Klein, 2007). Skov and Stage (2005) reported a study comparing problems found by students using a conceptual tool with students not using it and with the evaluation outcome of the teachers. The use of the tool resulted in more problems found by students, which supports the idea that additional scaffolding elements are required for usability evaluation. The elaboration of case collections for usability engineering has been approached by Carrol and Rosson (2005). These are comprehensive cases that fit project-based education, but there is a need to understand and learn to apply the concrete guidelines and heuristics related to expert-based usability evaluation, as these can provide detailed insights on how students face usability problems.

This paper reports preliminary results of an ongoing comprehensive study on problems found by students when approaching inspection methods for usability evaluation, concretely, heuristic evaluation. Heuristic evaluation is a problem-oriented usability evaluation method (Nielsen, 1992). In its initial proposal by Nielsen and Molich (1990), it was found that it served to identify 55 to 90 percent of the known usability problems user interfaces, concluding that heuristic evaluation was a cheap and intuitive method for evaluating the user interface. Heuristic evaluation has an additional interesting property in the educational context: it forces students to classify usability problems, assess their importance and argument why they qualify as such. Consequently, the analysis of records of student heuristic evaluation has the potential to uncover underlying false assumptions, misunderstandings or in general difficulties in acquiring user interface evaluation abilities.

The rest of this paper is structured as follows. Section 2 describes the context of the case reported here and the data gathering procedures. Section 3 provides the results of evaluating a small sample of users and discusses these preliminary findings. An example of instruction design from the analysis of assignment data is provided in Section 4. Finally, conclusions and outlook is provided in Section 5.

2. Context and data gathering

The ACM/IEEE/AIS curricula recommendations for Computer Science include 8 core hours of Human-Computer Interaction, which is concerned with the required skill of “knowing how to create a usable interface and testing the usability of that interface”. In the detailed topics related to HCI, the recommendations include “evaluation without typical users”, including guidelines, heuristics and expert-based analysis. While user testing is considered the most reliable way of evaluating user interfaces, teaching guidelines and heuristic evaluation have the benefit of not requiring students to be provided with an observational setting, so that distance students are able to exercise that kind of evaluation. Further, such kinds of evaluations do not rely on the availability of users for testing, but on the application of theoretical elements and guides. If these elements are used for summative assessment of students, the student’s responses can be evaluated rather objectively by instructors.

The context of the present case was an elective course on Human Computer Interaction at the last year of a four-year degree in Computer Science. The authors had been teaching the course since 2004 following a continuous assessment method. The course starts with an HCI fundamentals module, followed by a module on user interface design and then a usability evaluation module. Students are taught about user testing, but also other methods including heuristic evaluation. One of the assignments included in the continuous assessment presents the students with concrete Web sites for heuristic evaluation, following Nielsen's heuristics and rating scales. Students have previously exercised the technique at an heuristic evaluation lab, and they are equipped with knowledge on guideline-based assessment as a supplementary tool[6].

The users have to report on problems found as exemplified by the Table 1 entry.

Table 1. Categories for problems, severity and error description in problems found.

Category	Severity					Error description
	0	1	2	3	4	
VSS		X				The Web site is too much interactive
UCF					X	Some pages linked from the homepage have no option to *go back* or this option is difficult to find.

Categories for problems are the following: visibility of system status (VSS), Match between system and the real world (MSRW), user control and freedom (UCF), consistency and standards (CS), error prevention (EP), recognition rather than recall (RRR), flexibility and efficiency of use (FEU), aesthetic and minimalist design (AMD), help users recognize, diagnose, and recover from errors (HURE) and help and documentation (HD). Hvannberg, Lai-Chong and Larusdottir (2008) found no significant differences between using Nielsen's heuristics and the cognitive principles of Gerhardt-Powals, and no difference was also found in either using a web tool or paper, so we have not initially not considered these variables in the design of the study.

The analysis of assignments requires a detailed representation of problems found versus actual problems (as identified by the instructors), an analysis of the appropriateness of the assessment of severity, and the qualitative examination of justifications in the problems found. The data analysis was done by creating a database following the schema depicted in the Fig. 1.

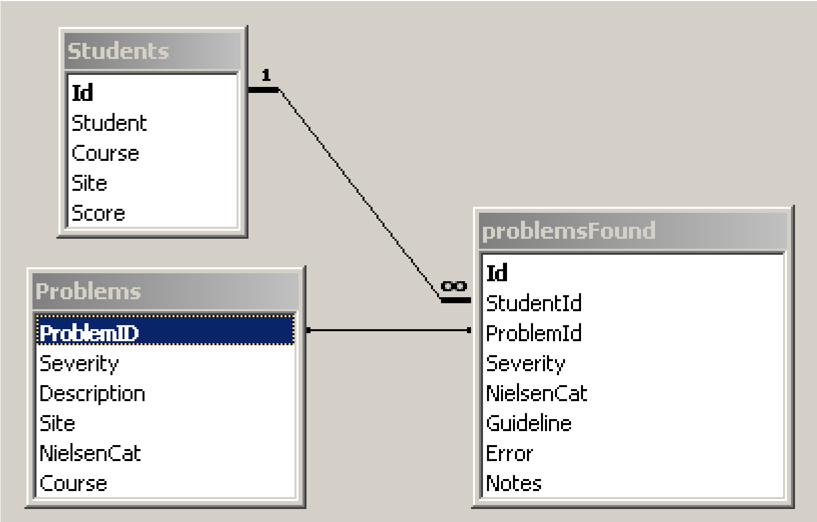

Fig. 1. Database created for data analysis

[6] Concretely, they have been introduced at the beginning of the course to guideline-based usability analysis using the research-based guidelines elaborated by the U.S. Department of Health and Human Services available at: http://www.usability.gov/pdfs/guidelines.html

Table `Students` recorded information on the students evaluated, the academic year in which they took the HCI course, the site they evaluated heuristically and their overall grade in the course (`score`). Then, each of the problems identified by students is recorded in the `problemsFound` table, including severity assigned, Nielsen category selected, and the related guideline(s) identified (if any). The instructors filled the `Error` field, classifying entries in which students failed to justify the reason of the problem, or simply reported a situation that was not actually a usability problem. Problems found that were actual problems are related to the `Problems` table, which stores the "correct" problems, categories and severity ratings. Matching problem found descriptions was done by the instructions, even though matching problems has been found to be controversial across matching techniques (Hornbaek and Frokjaer, 2008), here the group and criteria are homogeneous, and they are built by the same matcher.

3. Discussion

A sample of the problems found by five students (related to the same target web site) was used as a first application of the analysis method. A total of 82 problems were found by the students, of which 31 were correctly identified problems. Despite the apparent high error rate of this figure (below 50%), it represents a common situation, as there is in general a high dispersion in actual problems found, pointing out that there is some barrier for some students even to identify some prominent problems. This suggests general misunderstandings of usability problems, as confirmed by the qualitative analysis of justification reports. For example, the student with Id 5 reported a total of 31 problems, of which only 11 matched actual problems. Nonetheless, from the other 20 problems, 10 of them were incorrect because they are considered redundant statement (see section 4), that is, very related or identical to problems already detected. For example the student detected that there is no a link to the homepage in 4 pages and reported 4 different errors. There were other 3 reported errors that are considered as partially mistakes because they are very vague descriptions of errors or very subjective (and obvious) opinions (see in section 4 'vague statements').

Severity assessments in properly identified problems were identical to instructor assessments in only 13 cases, however, relaxing the coincidence in plus/minus one severity point, the figure was 21 out of 31, which represents a reasonable fit given that severity ratings entail a degree of subjectivity associated with the identification of priorities in fixing errors.

4. Elaborating problem statements from data

Evidence gathered from student assignments can be used for developing new assignments and also for the development of guided problems that are targeted to raise difficulties found by students in the past. This requires a categorization of mistakes found. In the sample discussed above, errors were classified in three categories: incorrect, redundant and vague statements.

Redundant statements entail the repetition of the same problem found in several sections of the Web site. This adds nothing to the heuristic evaluation report and should be avoided. This frequent error category reveals a methodological mistake, which can be avoided easily by emphasizing the unique nature of problem entries in the report. In this case, a clarification about redundancy was included in the heuristic evaluation report given to students, and an example emphasizing that aspect was also included in the case presented at the lab, which is discussed with the tutors.

Vague statements in contrast represent a very broad category, including statements that are to some extent correct, but fail in accurately describing the problem form a usability perspective or even use clichés and stereotypes. Examples of these statements are “the page is too interactive” (supposedly referring to the impossibility of avoiding animation, i.e. a lack of user control) or “the consistency of contents and interfaces is poor” (which fails in detailing the concrete inconsistent elements, e.g. navigation structures, general appearance, etc.).

Those reported errors that don’t reflect real usability problems are considered *incorrect* statements. Examples of incorrect statements are “The site is not WAI compliant” or “In page X there is no link to homepage” when it exists. Although statements like the last one can reflect usability problems, they are considered as errors in the assessment of a report on heuristic usability evaluation.

Once the errors reported by students have been analyzed, teachers can fine tune the problems (examples and lab practices) to reflect in usability evaluation case studies most common misunderstandings. To do so, found errors can be grouped by heuristic. After explanations of most common errors by heuristic, the students must evaluate examples and counter examples implemented *ad-hoc* as prototypes which reflect those errors. The learning process can be strengthened planning user interface design problems in which students had to reflect explicitly how to deal with that kind of errors.

5. Conclusions and outlook

Problem-based approaches to instruction rest in the careful design of problems or cases that are used both for tutoring and evaluating students. Some topics lack reliable, mature problem collections that could be used by instructors as a point of departure. In these cases, evidence can be gathered form student assignments involving problem solving, so that new or revised problems can integrate aspects known to have been difficult to master or sources of common errors in their statement and task design. This paper have reported on the preliminary results of that technique applied to gaining insight on student’s difficulties and pitfalls when confronting heuristic usability evaluation. The procedure entailed a detailed analysis of problem report entries submitted by students and their categorization, leading to three broad categories of error that entail different kinds of update of teaching material.

Future work will cover a detailed analysis of the database of at least two student cohorts in the HCI course, and a comprehensive account of potential sources of error and the problem design guidelines that follow from them.

Acknowledgements

The research is within the framework of the activity of the educational innovation project UAH/EV243, from University of Alcalá.

References

Carroll, J. M. and Rosson, M. B. 2005. A case library for teaching usability engineering: Design rationale, development, and classroom experience. J. Educ. Resour. Comput. 5, 1 (Mar. 2005), 3.

Hartson, H.R., Andre, T.S. and Williges, R.C. (2001) Criteria for evaluating usability evaluation methods, International Journal of Human–Computer Interaction 13 (4) (2001), pp. 373–410.

Hornbaek, K., Frokjaer, H. (2008) Comparison of techniques for matching of usability problem descriptions, Interacting with Computers, 20(6), pp. 505-514

Hvannberg, E.T., Lai-Chong, E., Larusdottir, M.K. (2008) Heuristic evaluation: Comparing ways of finding and reporting usability problems, Interacting with Computers, (19)2, pp. 225-240

Merrill, M.D. (2007) A Task-Centered Instructional Strategy. Journal of Research on Technology in Education, 40(1), pp. 33-50

Nielsen, J. 1992. Finding usability problems through heuristic evaluation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Monterey, California, United States, May 03 - 07, 1992). P. Bauersfeld, J. Bennett, and G. Lynch, Eds. CHI '92. ACM, New York, NY, 373-380.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proceedings of the Conference on Human Factors in Computing Systems (CHI 90), Seattle, WA, April 1-5, pp. 249-256.

Rozanski, E. P. and Haake, A. R. (2003). The many facets of HCI. In Proceedings of the 4th Conference on information Technology Curriculum (Lafayette, Indiana, USA, October 16 - 18, 2003). CITC4 '03. ACM, New York, NY, pp. 180-185.

Simons, K. and Klein, J. (2007). The Impact of Scaffolding and Student Achievement Levels in a Problem-based Learning Environment. Instructional Science, 35(1), pp. 41-72.

Skov, M. B. and Stage, J. (2005). Supporting problem identification in usability evaluations. In Proceedings of the 17th Australia Conference on Computer-Human interaction (Canberra, Australia, November 21 - 25, 2005). OZCHI, vol. 122. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, pp. 1-9.