Sharing Usability Problem Sets within and between Groups

Gudmundur Freyr Jonasson, Ebba Thora Hvannberg

University of Iceland, Hjardarhaga 2-6, 107, Reykjavik, Iceland
gfj@hi.is, ebba@hi.is

Abstract. Merging similar usability problems is often a time consuming step in the process of usability evaluation. We have developed a tool which can aid with merging duplicate problem descriptions. The tool enables evaluators to define constructs which describe a usability problem and store them in a database. The evaluator can search for existing constructs, problem descriptions and problem sets using the query language XQuery. The XQuery search function also enables the evaluator to compare usability problems. The tool is designed to be flexible and the exchange of data between evaluators is simple exploiting the compatibility of XML (Extensible Markup Language) and a well defined data structure. We have performed an analysis on a problem set to check for the comparison quality of the merging function in the tool. This has allowed us to suggest further ways to compare usability problems.

Keywords: Usability, problem, construct definition, merging, exchange, relevance.

1 Introduction

Merging similar usability problems is often a time consuming step in the process of usability evaluation. The complexity lies in the data which comes from different users and also that evaluations of the same system may be done at different places or times and sometimes with different constructs [1]. Usability problem sets are compared not only for practical reasons but also for research by meta-analysis as several examples show [2, 3]. Our goal is to design and implement a tool which enables usability testers to record usability problems into a database using different usability evaluation methods, and to allow evaluators to search the database for similar problems, exchange data sets and to perform a meta-analysis of data sets. By using the tool the analysis following user tests should be less time consuming and the exchange of data between evaluators made easier. The tool uses methods to diagnose what problem descriptions are unique and what descriptions are duplicates by comparing constructs within problem descriptions. If there is a match according to a predefined relevance scale the descriptions are grouped together. The user then makes the final call whether or not the two problem descriptions are duplicates. If the user decides that the descriptions are duplicates then those two problem descriptions can be merged into one description.

The tool is designed to be flexible with the use of XML. Therefore, data exchange between different users is easy, given the good compatibility of XML. The tool runs on internet browsers and connects to an XML database called eXist (http://exist.sourceforge.net/) and uses XQuery to search the database.

2 Requirements and implementation of the tool

In this section we will describe three main use cases of the tool: defining a construct to describe a usability problem, searching the problem database and matching usability problems. First, the tool allows the user to flexibly define constructs to describe usability problems. At the onset of recording usability problems from an evaluation in the tool, the user can select a set of constructs already defined in the tool, and add new constructs. To enable exchange of data sets and meta-analysis of data sets, it is necessary to record meta-data for individual constructs. Cockton [4] has suggested a coding dimension format which includes: Name of construct, Definition, Operationalization (explains what the coder should do), Evidence Requirements (describes how to code values/levels), Sources of Bias (describes possible flaws in coding), Motivation and Relevance (explains why this construct matters), Used in (references) and Examples (from existing case studies).

Second, after having recorded a set of usability problems, the user can search for problems with the use of XQueries. Given the power of the XQuery language, users can make very precise searches on the whole database, within usability problems of a particular usability group or usability problems recorded by single authors. Queries can be stored and recalled in the tool, allowing the user to use common queries, such as ‘Find all problems of highest severity level’. Thus expert users are provided the power to enter any query but novice users of the tool, can apply predefined queries.

Third, the tool, upon request, notifies the user of similar usability problem descriptions. The similarity is based on relevance, defined by a set of constructs, which can be defined by the user. The tool compares constructs of problems and if they satisfy a given relevance the tool groups those problem descriptions together and displays the results to the user. For example, a user defined relevance-scale TEF can be defined for the variables Trigger, Expected Phase and Failure Qualifier. Trigger describes what evaluators are doing when he/she discovers the problem, Expected Phase indicates in which phase of a software development life cycle the developer is expected to be able to fix a usability problem and Failure Qualifier which is used to capture more meaning about the usability problem. If, say, four usability problem descriptions in a given set have exactly the same values in those three constructs, they are grouped together and the user makes the final call by evaluating if the descriptions are describing the same problem or not. Problems which the user has acknowledged to be the same are marked as such in the database by putting them in the same equivalence group. The equivalent groups, i.e. the mergeable problems, appear as output in alternate colours, grey or white.

3 Quality analysis of merging

A quality of a merging tool is rated by how accurately it can identify mergeable problems. More specifically we want to find out whether the merging tool could correctly identify set of problems which a human agrees are mergeable. There are several possibilities: a) The tool suggests a match between problems but a human not b) the tool misses a match, which a human has identified c) the tool and the human agree that there is a match. In this pilot research study, we addressed options a) and c), labeling the former a 'False match' and the latter 'Mergable'. In order to inspect the quality of the merging tool, we examined a problem set resulting from a usability evaluation of a learning management system (OWL). The problem set has 71 problems, which have already been consolidated, and thus the problems in the set have been filtered and merged by a usability expert. Despite this earlier merging, the merging tool suggested seven groups of usability problems. We analyzed the quality of the tool by rating the tool's suggestions either 'False match or 'Mergeable' in column two of Table 1.

Three merging suggestions of seven were correct. This analysis shows that the tool found opportunities for merging problems, even though they had been consolidated earlier manually. Several problems became evident during the merging exercise. First, some of the constructs, i.e. Context has too many values. This can be remedied by suggesting ontology of values within a variable by creating a few subsets of values, where the values within each subset are equal. Second, some of the constructs are textual and hence a text matching algorithm needs to be implemented. We have not yet analyzed the remainder of the problems in the OWL set to see if any problems were missed by the merging function in the tool, i.e. option b) above.

Table 1. Outcomes of matches using the relevance scale TEF

Matched Group ID	Rating	Matched UPs	Similarities
1	False match	1 and 27	The UPs describe two different use cases.
2	False match	14 and 61	
3	Mergeable	16, 56, 57 and 64	UP 16 and 56 could be merged.
4	Mergeable	20 and 40	Both users are trying to change previously stored text.
5	False match	21 and 59	UP 59 should be put in group 4
6	False match	26 and 66	
7	Mergeable	32 and 45	Both users are having difficulties identifying hyperlinks.

4 Future work

The development and analysis tool described in this paper marks a first step to reach the goal of providing evaluators with an automatic technique to merge problem sets and allow for their exchange. Although the tool itself is quite usable, immediate next steps will include easier access to search functions through an enhanced user interface and import/export functionality from and to Excel. In this paper, we have focused on one of three use cases, i.e. matching usability problems. As we mentioned above, further work is needed to understand the automatic matching of usability problems, through the recoding of variables which have a large range of values such as the variable Context or free text variables such as Description. It will also be interesting to address textual comparison further with advanced algorithms such as text mining or latent semantic analysis. Other use cases await further studies, such as exploring the tools' usefulness for exchanging problem sets with different construct definitions between evaluators. As future work it will be interesting to explore how relationships between the constructs definitions can be made by expressing that a construct can be derived from another; a construct can be equal to another; a construct can be an aggregation of several other constructs, a construct can determine another etc. Thus, the tool can also have a potential for researchers in performing a meta-analysis of problem sets.

Acknowledgement

This work is in part supported by the Icelandic National Research Fund, RANNÍS. We thank the reviewers for their useful comments.

References

1. Law, E. L-C., Hvannberg, E.T., Consolidating usability problems with novice evaluators, Proceedings of the 5th Nordic conference on Human-computer interaction, Lund, Sweden, p. 495-498, ACM (2008)
2. Hornbæk, K., Law, E. L-C., Meta-analysis of correlations among usability measures, Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human Factors in computing systems, p. 617-626, ACM (2007)
3. Vermeeren, A.P.O.S., Attema, J., Akar, E., Ridder H. de, Doorn, A.J. van, Ergub C., Berkman, A.E., Maguire M.C. Usability Problem Reports for Comparative Studies: Consistency and Inspectability. Human-Computer Interaction 23(4), p. 329-380 (2008).
4. Cockton, G. & MAUSE. COST294-MAUSE workshop on coding constructs definitions and coding problem sets, 7th June 2007, Salzburg, Austria http://www.cost294.org (2007)