A Hybrid Approach for Modelling Early Prototype Evaluation Under User-Centred Design Through Association Rules

María Paula González, Toni Granollers, and Jesús Lorés

Departament of Computer Science – Universitat de Lleida
C/Jaume II, 69 – 25001 Lleida, Spain
{mpg,toni,jesus}@diei.UdL.es

Abstract. One of the main activities in User Centred Design (UCD) is prototype evaluation, which is traditionally performed by means of an Evaluation Stage that looks for the redefinition of the prototype requirements, involving quantitative and qualitative usability testing techniques. This paper describes a new approach in which the traditional methodology for performing the Evaluation Stage under UCD is embedded in a framework with capabilities for mining association rules. This allows to minimise the impact of the interpretation bias of the evaluation team when analysing ambiguous user statements in natural language.

Keywords: UCD – Evaluation Stage – Data mining – Association Rules.

1 Introduction and Motivations

During the last decade, User-Centred Design (UCD) principles have proven to be successful cutting costs in software development and increasing user satisfaction and productivity [1]. When a software system is iteratively produced under UCD a number of intermediate prototypes $\mathcal{P}_1, \ldots, \mathcal{P}_n$ are developed. Between a prototype $\mathcal{P}_i$ and its successor $\mathcal{P}_{i+1}$ several intermediate stages are involved, such as the *Evaluation Stage*(ES)[1] whose main goal consists in contrasting the usability[2] features of the prototype $\mathcal{P}_i$ against its requirements in order to refine them. The main components for performing the ES for a prototype P_i are shown in full lines in Fig. 1. Given the design of P_i and its requirement specification R_i, different usability evaluation techniques are applied by the software development team (DT) with an active user role. The aim of such techniques is to refine the current set of requirements R_i transforming it into a new requirement definition R_{i+1} for a new (still undeveloped) prototype P_{i+1}. This redefinition

[1] In what follows we assume that the Evaluation Stage is always performed early during the software development process (i.e. we focus on early prototypes) and under a UCD perspective.

[2] Usability is defined by the ISO 9241-11 norm as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

G. Doherty and A. Blandford (Eds.): DSVIS 2006, LNCS 4323, pp. 213–219, 2007.

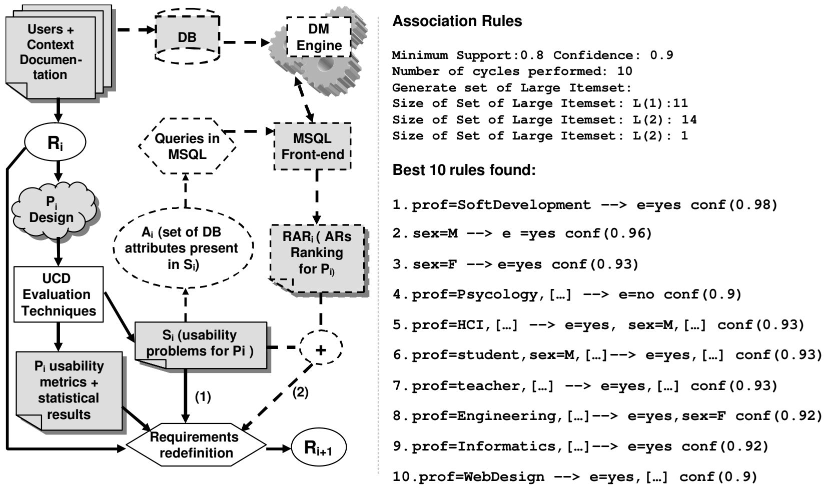

Fig. 1. Left: Schema of the Evaluation Stage for a prototype P_i (full lines represent the traditional approach whereas dotted lines stand for the new components introduced for integrating association rule mining). **Right:** Ranked list RAR_1 of association rules (best 10 out of 50 found) for the experiment in Section 4 ([...] means that more attributes were present in the rules, but without direct relation with this experiment).

is carried out by performing both a *quantitative* and a *qualitative usability estimation* of P_i to make explicit its usability features and problems, which will be then contrasted against R_i [1].

Qualitative usability estimation of a prototype P_i is commonly carried out by techniques based on the interpretation of the DT when analysing user statements expressed in natural language [7]. The output of such techniques is a a set $S_i = \{s_1, \ldots, s_k\}$ of usability problems (statements in natural language) that are used to contrast the current requirements of P_i against its design (as the usability problems in S_i express the problematic features in the design of P_i) in order to redefine those current requirements [1] (arrow (1) in Fig. 1). As a result, a new requirement definition R_{i+1} is generated to be considered in the next cycle of the iterative software production, where a new prototype P_{i+1} will be designed on the basis of R_{i+1}. The interpretation bias of the DT when analysing ambiguous user assertions leads to some lack of objectiveness. For this reason, one of the most significative challenges in the ES is to define a usability testing process that combines high expressiveness, minimum risk and a maximal objectiveness level.

This paper describes the main features of a new, hybrid approach in which the traditional methodology of the ES for prototypes is embedded in a framework with capabilities for mining *association rules*. These rules will allow to automatically find unbiased cause-effect relationships in data coming from the definition of the requirements of $\mathcal{P}_i$. As a result, the DT will be able to use these relationships to reinforce their understanding of qualitative usability problems detected during the ES of $\mathcal{P}_i$ in order to improve the requirement definition for the next prototype $\mathcal{P}_{i+1}$.

2 Mining Association Rules: Fundamentals

Datamining involves a collection of techniques for the analysis of large data sets in order to discover patterns of interest and extract potentially useful information [6]. This extraction can be formalised as a process in which data from (possibly) heterogenous sources are normalised and combined into a transactional database, in which features or *attributes* are identified, and onto which different techniques are performed to mine the available data.

Association rule mining [6] is a powerful datamining technique which allows to find hidden relationships among attributes in a transactional database. Every transaction consists of a set of items $I = \{i_1, \ldots, i_m\}$ and a transaction identifier. Association rules (AR) are implications of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$. In addition to the antecedent A (the "if" part) and the consequent B (the "then" part), an AR has several interestingness measures that express the quality of the rule. One relevant measure is called the *support* for the rule, which is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. (i.e., the percentage of transactions in D that contain $A \cup B$). Other important measure is known as the *confidence* of the rule, and corresponds to the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent. (i.e. the percentage of transactions in D containing A that also contain B). Computing ARs is computationally complex task, and several efficient algorithms (e.g. Apriori or FP-Growth [6]) have been developed. The AR mining process generates usually a huge number of rules, making it necessary to provide powerful query primitives to post-process the generated rulebase, as well as for performing selective, query based generation [14]. Several specialised query languages have been proposed in the literature such as MSQL [14] and DMQL [6], among others.

3 Enhancing Early Prototype Evaluation Through ARs

We have developed an hybrid approach for modelling Qualitative Evaluation in which the elements detailed in Section 1 are integrated with new components for carrying out association rule mining. These new components are depicted in dotted lines in Fig. 1 (left). The traditional use of the results coming from the qualitative usability testing of the prototype (arrow 1 in Fig. 1) is complemented by an AR-based approach (arrow 2). This new hybrid approach intents to generate a more objective and expressive output set of usability problems by minimizing the impact of the interpretation bias of the DT when interpreting ambiguous user statements expressed in natural language that were generated during the Qualitative Evaluation process.

In the proposed framework the input for the process is similar as before (an early prototype design P_i, and its requirement specification R_i). Both quantitative and qualitative usability problems are measured by UCD Evaluation Techniques. Qualitative results are compiled in a set $S_i = \{s_1, \ldots, s_k\}$ of usability problems expressed as natural language sentences. At the same time, all the heterogeneous information coming from the development of the current prototype (e.g. data from the definition of context of use, user profiles or organisational requirements specifications for P_1) are collected in a database DB.[3]

In the new, extended framework the DT can now formulate queries which will provide them with additional information (a list of ranked association rules RAR_i). These ARs express hidden cause-effect relationships among attributes present not only in the set S_i of usability problems but also in the information provided by users during the specification of R_i, stored in DB. This new ranked list RAR_i will help to cope with the impact of the interpretation bias of the DT when analysing ambiguous elements in S_i to redefine R_i into R_{i+1}. In particular, those attributes from DB present in S_i are included in a set A_i and used by the DT members to pose queries expressed in a MSQL language. These queries are processed by a front-end module linked with with the actual DM engine which provides the algorithms required for processing the database DB and obtaining ARs from it. As an output, the front-end module provides a ranked list RAR_i for the prototype P_i. Every item in RAR_i represents a non trivial and hidden cause–effect relationship present in the DB. Adding RAR_i to the original usability problem list S_i provides a new perspective to interpret ambiguous user statements in natural language when contrasting the design of P_i with the conclusions coming from the qualitative usability measure. It must be noted that as the DB collects information coming from users, the UCD perspective of our hybrid approach is reinforced.

4 Experimental Results

In this Section we will summarise the results obtained after using the proposed approach for the development of a web site under a UCD perspective.[4] The web site is intended for educational purposes. The development team DT was formed by two CS students, one usability expert, one CS professional with experience in DM techniques and an interdisciplinary group of six university students (final users). The WEKA [4] platform was available for performing datamining, and its interface was used as a basic MSQL front-end for posing queries.

Experiment (sketch): Following the first part of the methodology shown in Fig. 1, (dotted lines) a first prototype P_1 was developed. An exhaustive poll was carried out on the Web to specify the context of use and the user profiles.

[3] Notice that spreadsheets and databases used to compute the quantitative usability can be used when constructing the database DB.

[4] The final version of this web can be found at http://www.mpiua.net. It was developed in the GRIHO Labs at the Universitat de Lleida (Spain).

More than 200 results were compiled in a relational database and used to define the requirement specifications R_1 for P_1. Then the prototype P_1 was designed, including a wide spectrum of resources (such as information, bibliography, etc.), the possibility of registering as a user and having a discussion forum, and restricted access to examples (depending on features provided in the registration, e.g. profession). Next the ES was performed. The design of P_1 was confronted against its requirements R_1 by evaluating the usability of the design P_1. The extended approach proposed in this paper was used to perform the redefinition of requirements in R_1. Quantitative results were summarised into statistical graphics [5]. Qualitative results were compiled as a set $S_1 = \{s_1, \ldots, s_5\}$ generated by means of a Focus-Group [9] and a Stakeholders Meeting [7]. Five elements in S_1 were problematic as different members in the DT had conflicting opinions about their relevance and no clear evidence could be found based on the statistical information. Four of these problems (80%) could be treated with our approach. We will summarise the analysis made by the DT during the contrast between the requirement r_1 = {The web must include a resource called "examples"} and the usability problem s_4 = {There is no agreement with the necessity of having an hyperlink called "examples": should it be available only for determinate users?}. As the most relevant issue present in the usability problem s_1 was the DB attribute "examples" and this attribute was one of the elements in S_1, the methodology presented in Section 3 could be used this case. The DT defined the following query in MSQL language: GetRulesDB where [Antecedent has {prof=* or sex=*} and Consequent has {e=*} and support $>$ 0.8 and confidence $>$ 0.9] In this query "prof" and "e" stand for "profession" and "examples", resp. The query was adapted for the basic front-end interface provided by the WEKA platform and solved by the WEKA datamining engine using the APRIORI Algorithm [6]. As shown in the above query, the ranking function was a combination of a particular threshold for support and confidence. The final result was the list RAR_1 depicted in Fig. 1 (right). Finally, the DT confronted the usability problem s_4 as well as the elements of the ranked list RAR_1 against the requirement r_1. The evidence provided by RAR_1 was strong enough to disregard the usability problem s_4 and avoid the redefinition of the requirement r_1. The iterative development of the final web continued with the design of next prototype P_2 on the basis of R_2.

5 Related Work. Conclusions

To the best of our knowledge, there is no similar approach to integrate datamining techniques (such as association rules) for performing the Evaluation Stage within other models for software development related to UCD [1,2,9,10,11,12,13] as presented in this paper. Indeed, there are other proposals where association rules are used for assessing usability, but always on fully developed, executable software products (i.e., when the prototype design is no longer an issue), particularly in the context of user interfaces [8]. Datamining techniques have been successfully integrated for interface development in web sites. For example, the

[5] To consult quantitative results see http://www.alzado.org/articulo.php?id_art=417).

AWUSA framework [15] presents an automatic tool for evaluating usability in web sites by combining logging techniques and datamining along with the static structure of the web site. Another example is described in [5], where logging techniques are applied based on browsing activities performed by users.

In this paper we have presented a novel approach for integrating association rule mining with the traditional formulation for Prototype Evaluation under UCD. Our experiments have shown that our proposal can be successfully applied to early stages in the life cycle of software developmentcontributing to a better understanding of results from qualitative usability testing for P_i, allowing a more accurate redefinition of its requirements for the next prototype P_{i+1}. Part of our future work is focused on using the proposed approach in professional usability evaluations. In this respect, a prototype of an electronic newspaper related with the ELIN Project[6] and a prototype of a new website for the EPS of the Universitat de Lleida are under evaluation. We are also interested on testing different ranking functions for association rules, evaluating their applicability in early prototyping. In particular, we are interested in rule prioritization by taking into account the cost associated with software development, as suggested in [3]. Research in this direction is currently being pursued.

Acknowledgments. We thank the reviewers for their comments which helped to improve the original version of this paper. This work was partially supported by Cicyt Project ADACO (TIN2004-08000-C03-03), and Project SGR-00881 (Generalitat de Catalunya, Spain).

References

1. Sutcliffe A. *User-Centred Requirements Engineering.* Springer, 2002.
2. Sydner C. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces.* M. Kaufmann, 2003.
3. Choi D., Ahn B., and Kim S. Prioritization of association rules in DM: Multiple criteria decision approach. *Expert Systems with Applications*, 29:867–878, 2005.
4. Witten I. and Eibe F. *Data Mining: Practical machine learning tools and techniques.* M. Kaufmann, 2005.
5. Alipio J., Poças J., and Azevedo P. Recommendation with Association Rules: a web mining application. *Data Mining and Warehouses Conf. IS-2002*, 2002.
6. Han J. and Kamber M. *Data Mining.* M. Kaufmann, 2001.
7. Dumas J.S. and Redish J. C. *A Practical Guide to Usability Testing.* Intl. Specialized Book Service Inc, 2000.
8. Hornbaek K. Current practice in measuring usability: Challenges to usability studies and research. *Int. Journal of Human-Computer Studies*, page (in press), 2005.
9. Constantine L. and Lockwood L. *Software for Use. A practical Guide to the Models and Methods of Usage-Centered Design.* Addison-Wesley, 1999.
10. Ivory M. Y. and Hearst M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33:470–516, 2001.
11. D. J. Mayhew. *The Usability Engineering Lifecicle. A practioner's handbook for user interface desing.* M. Kaufmann, 1999.

[6] See European IST-2000-30188 ELIN Project at http://elin.grupoalamo.com/

12. M. Rosson and J. Carroll. *Usability Engineering: scenario-based developement of HCI*. M. Kaufmann, 2002.
13. Granollers T., Perdrix F., and Lores J. Usability Engineering Process Model. Integration with Software Engineering. *ACM HCI'03*, 2003.
14. Imielinski T. and Virmani A. Msql: A query language for database mining. In Springer Science and Business Media B.V., editors, *Data Mining and Knowledge Discovery*, volume 3, pages 373–408, 1999.
15. Tiedtke T., Märtin C., and Gerth N. Awusa. a tool for automated website usability anlaysis. *9th Int. Workshop DSVIS*, 2002.