

Assessing Usability Problems in Latin-American Academic Webpages with Cognitive Walkthroughs and Datamining Techniques

María Paula González^{1,2}, Jesús Lorés*, and Antoni Granollers¹

¹ Grup GRIHO – Universitat de Lleida - c/Jaume II, 69 –25001 Lérida, Spain
{mpg,tonig}@diei.udl.es

² Dept. of Computer Science and Engineering – Universidad Nacional del Sur
Av. Alem 1253 – 8000 Bahía Blanca, Argentina
mpg@cs.uns.edu.ar

Abstract. Qualitative usability evaluation is usually included within the Evaluation Stage in Usability Engineering through a *Qualitative Usability Testing process QUT*. This *QUT* process includes the application of methods that have been defined focusing on the evaluation of a particular interactive system, becoming highly expensive when a context of use has to be evaluated (by analyzing a large number of interfaces belonging to that context) in order to detect common usability problems from a qualitative viewpoint. This paper presents the *QUTC_{KDD}* methodology which incorporate techniques from Knowledge Discovery in Databases, extending the existing *QUT* process in order to solve the above situation. To illustrate the *QUTC_{KDD}* methodology, an experimentation related to a particular Latin-American context of use is also discussed.

1 Introduction and Motivations

Usability is a software attribute associated with the “ease of use and to learn” of a given interactive system [7]. Nowadays usability evaluation is becoming an important part of software development based on Usability Engineering (*UE*) [11,14]. During usability evaluation a *qualitative usability estimation* is performed, since no quantitative measure can be expressive enough to represent something so complex as a usability problem [4,7]. Indeed, qualitative results are collected through a *Qualitative Usability Testing process (QUT)* which includes a number of different methods (such as Cognitive Walkthrough (CW) [21], etc.) focused on analyzing the interface of a particular system [2,7,13,18,20]. In spite of the differences present among these methods, there exist a number of common elements (see dashed box in Fig. 1) that can be identified in any *QUT* process: an *interactive system S* which is being assessed using the *QUT* process; a *generic evaluation method M* for qualitative usability testing (selected among the possible methods mentioned above); a *particularization* (or instance) *M'* of the method *M*, defined for applying *M* to the particular usability evaluation of

* In memoriam.

the system S ; a dataset representing the *results* obtained from the application of M' over S ; a number of *visual representations* (visualization) of the above results which facilitate their processing and analysis; a final *usability report* containing the results coming from the whole qualitative usability evaluation process. Besides, some typical steps are carried out when applying a *QUT* process to a generic system S , namely:¹

- *Planning.* A particular method M is selected by the evaluation team, depending on several aspects (e.g., its adequacy to the system S at issue). According to M , sometimes a redefinition of the team should be done. Besides, the conditions to perform M (timetable, physical environment, equipment, etc.) should be stated. Note that most *QUT* methods are defined in a general way. Consequently, many times a specialization M' of M should be defined in order to adequate M to the particular process in progress. The definition of M' includes the selection of appropriate ideal values for the context of use involved, and a more specific definition of the main features of M (e.g., the definition of tasks in CW). Additionally, the selection of specialized software may be required.
- *Execution.* This step materializes the real execution of M' , which may require the use of a specialized software with adequate parameters. The members of the evaluation team will apply M' to test the system S under evaluation, carrying out actions such as task completion, time capture, collecting usability data, etc. Depending on the features of M' , the process will be carried out different numbers of times. It must be stressed that due to the nature of *QUT* methods, qualitative information is collected and complemented with natural language explanations.
- *Processing and analyzing results.* A crucial part of the *QUT* process is the processing and the analysis of the obtained results. In the worst case, the final usability problem set is created just by compiling the output data from M' without any other consideration. In an ideal situation, the above output data are processed and analyzed to achieve a general diagnosis of the usability problems in S . To do this, the members of the evaluation team should compare those data in order to detect agreements and disagreements between the different applications of M' during the previous *QUT* step. Any usability problem should be carefully examined to decide if its relevance is strong enough. Frequently, this examination implies the consideration of natural language explanations that complement the alphanumeric data available. The use of different kinds of graphical representations (e.g., diagrams) can enhance the visualization and processing of the obtained results.
- *Reporting conclusions.* As a final output, the *QUT* process returns a set of usability problems detected in the system S under evaluation. To report this set there are different standards and reporting sheets [7].

Most *QUT* methods are focused on analyzing the interface of a particular system S , becoming complex when a large number of real cases from the same

¹ We are aware that in actual software development projects under *UE* some parts of these steps might overlap.

context of use have to be jointly considered to provide a general diagnosis, since a considerable amount of qualitative information must be treated simultaneously. In this situation, it would be desirable that the evaluation team be able to consider all such information with a reasonable cost and without losing the richness of the qualitative perspective. Note that identifying the most general usability problem patterns of a context of use from a qualitative viewpoint can help not only to evaluate a new interface belonging to it, but also to prevent usability errors when a novel interactive system is being developed.

To cope with the above problem, this paper presents a new methodology called $QUTC_{KDD}$ in which association rules (a technique from *Knowledge Discovery in Databases (KDD)*) are used to extend the existing QUT process in order to provide a general usability diagnosis of a given context of use C . Starting from the data obtained by applying a traditional QUT method to a sample group of real interactive systems from C , the goal is to automatically detect qualitative usability problem patterns, each of them describing a relevant usability feature of C from a qualitative viewpoint.

2 Knowledge Discovering in Databases: Overview

KDD is a discipline which involves “the nontrivial extraction of implicit, previously unknown, and potentially useful information from data”. This extraction can be modelled as a process in which different steps [10,16] can be identified: 1) *Data preprocessing*: This first step condenses a number of intermediate steps, such as data cleaning, integration, selection and transformation. Data cleaning routines attempt to fill in missing values, smooth out noise, and correct inconsistencies in the data. If required, multiple data sources may be combined in a single database, and additional selection and transformation operations may be performed to unify and consolidate data formats. 2) *Datamining (DM)*: Once data has been preprocessed and condensed in a database, datamining techniques can be applied in order to extract relevant patterns from it. There exist several datamining techniques, such as association rules, decision trees, etc. (for a detailed discussion see [10,15]). 3) *Pattern evaluation and knowledge representation*: Different criteria (e.g. interestingness measures) are applied to evaluate the patterns obtained as an output of the datamining algorithms. 4) *Visualization*: Finally, visualization and representation techniques are used to present the mined knowledge to the user.

Based on the previous steps, a typical software platform for *KDD* would involve the following major components: a) a database or information repository, consisting of databases or spreadsheets; b) a *KDD* engine, which consists of a set of functional modules for accomplishing datamining algorithms, providing as well modules for pattern evaluation and visualization; c) a Front-End module which allows the final user to interact with the *KDD* engine, typically by means of a specialized *datamining query language (DMQL)* [10]. Different general-purpose platforms like WEKA [22] or Orange [6] have been developed to carry out the *KDD* process, each of them with their specific DMQL language.

Such query-oriented languages range from very basic command-line interpreters (as the Command-line Interface provided by the WEKA platform [22]) up to more sophisticated tools such as specialized scripting languages (as the one provided by the module `orngMySQL` in the Orange platform [6]), MSQL [12], or more recently KDDML [17], among others.

Association rule mining [10] is a powerful datamining technique which allows to find hidden relationships among attributes in a transactional database D . Every transaction consists of a set of items $I = \{i_1, \dots, i_m\}$. An association rule (AR) is an implication 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 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 association rules (ARs) is a computationally complex task, and several algorithms (e.g. APRIORI or FP-GROWTH [10]) have been developed. The AR mining process generates usually a huge number of rules, making it necessary to provide powerful query primitives for performing selective, query based generation [12].

3 Our Proposal. The $QUTC_{KDD}$ Process

In our proposal, KDD is used to gather common usability features of a large group of systems $S_C = \{S_1 \dots S_n\}$ belonging to the same context of use C in order to achieve a general diagnosis of C . In what follows we will call this new approach $QUTC_{KDD}$. By performing $QUTC_{KDD}$, the members of the evaluation team will be able to formulate *queries* to an underlying KDD engine, getting as an output additional, qualitative information that will condense usability problem patterns of C . The result of every query will be expressed in terms of a ranked list of *association rules* where every AR will express a hidden cause-effect relationship among usability features of S_C .

Fig. 1 shows a schematic view of the framework for $QUTC_{KDD}$, which is based on extending the traditional QUT approach by adding the following elements: a *transactional database* DB which stores all the results obtained from applying the traditional QUT process to S_C ; a *KDD engine* capable of automatically computing ARs ; a *DMQL Front-end* for the KDD engine, which should provide the interface for the evaluation team in order to pose queries (using the DMQL associated with the KDD platform); a *visualization module* (in most cases provided by the KDD engine itself) which provides different alternative graphical representations of the patterns found; and a *collection of usability problems patterns* (ranked lists of ARs) related with C . As before, we will summarize next the different steps related to applying a $QUTC_{KDD}$ process to S_C :

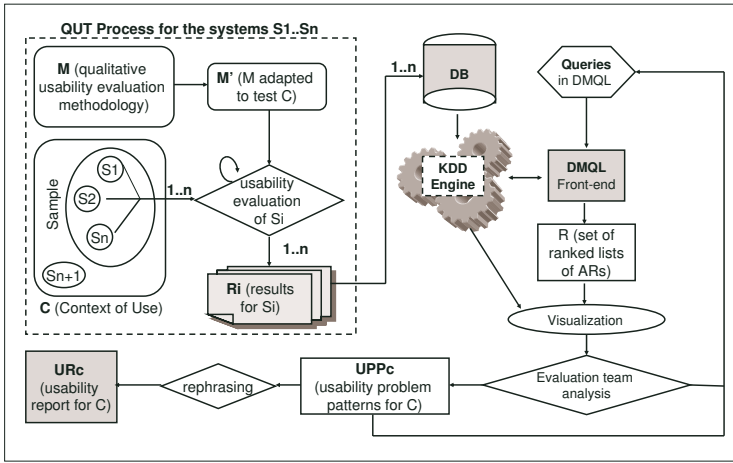


Fig. 1. Schema of the proposed Qualitative Usability Evaluation ($QUTC_{KDD}$) Process for a context of use C of interactive systems

- *Planning.* Planning the $QUTC_{KDD}$ process will involve similar steps as the ones described for the QUT process (see Section 1). Besides, a sample group of systems $S_C = \{S_1 \dots S_n\}$ must be selected to represent the context C . Choosing this sample group is not a trivial task, and the use of the selection criteria will depend on the nature of C [16]. In addition, the selection of a particular software platform and the associated DMQL must be considered.
- *Collecting the data.* Performing the method M' within all the elements in S_C is a central task. As the evaluation team carries out the usability evaluation of the elements in S_C , the obtained results (qualitative data) must be collected in an information repository (a database, a spreadsheet, a data warehouse, etc.).
- *Applying KDD techniques.* Once the data related with qualitative usability problems has been collected, the next step in the $QUTC_{KDD}$ process will consist in discovering patterns in that data by means of KDD techniques. The data repository obtained in the previous step has to be automatically transform in a transactional database [16]. The following steps will be carried out, as described in Section 2:
 - *Preprocessing the qualitative usability data.* There does not exist a fully automatic system capable to preprocess the data to clean and integrate them [16]. In the case of the $QUTC_{KDD}$ process cleaning and integrating the data is worthwhile, as the result can be used to obtain statistical information.
 - *Mining the data.* Within this step, the evaluation team can use the selected KDD software platform and the associated DMQL in order to pose queries using the DMQL Front-end. In a first stage, the evaluation team will post a starting query just to detect the most relevant cause-effect relationships (expressed as ARs) among the attributes in the database

DB on the basis of a threshold value for support, confidence and a maximum number of rules to be computed. As a result, a ranked list RAR_{i1} of ARs will be automatically obtained, containing the most relevant information (with possibly hidden cause-effect relationships) that needed to be considered by the evaluation team. By performing the next step (*Visualizing and analyzing $QUTC_{KDD}$ results*) the evaluation team will be able to analyze the relationships present in RAR_{i1} . Next, on the basis of RAR_{i1} information, more refined queries can be posted. This time, each new query for mining ARs will produce additional and more focused information, characterized as new ranked lists $RAR_{i2}, \dots, RAR_{ik}$.

- *Processing and analyzing results* The use of visualization tools provided for most KDD platforms [6,22] can help members of the evaluation team to observe different graphics and charts representing the obtained patterns. For the elements in each ranked list, results could be displayed as a sequence of ARs or as a group of Cartesian charts (each of them depicting the relationships between attributes present in the ARs in the ranked list). The goal of the evaluation team analysis will be to decide whether every pattern found depicts a usability problem of C . Indeed, each detected pattern will be discussed on the basis of its relevance and significance. For example, the relative position of every AR in a ranked list expresses these features, jointly with its interestingness (support and confidence).
- *Reporting conclusions.* The $QUTC_{KDD}$ process returns as an output a set of justified usability problems patterns detected for C . each pattern has to be rephrased as a usability problem expressed in some standard format (similar to the one shown in [7]). If necessary, different visualizations of the usability problem patterns generated during the $QUTC_{KDD}$ process can be added.

Note that during the traditional Evaluation Stage in UE different data generated during the evaluation processes are usually compiled into databases required for the statistical analysis associated with quantitative usability evaluation. The re-utilization of these databases minimizes the cost associated with the generation of the documentation needed to carry out $QUTC_{KDD}$. Note also that the computation of KDD algorithms should be performed only one time to achieve satisfactory results. Finally, we would like to remark the high level of automation present in the KDD stage within the $QUTC_{KDD}$ process, which is a highly desirable feature (as pointed out in [13]). In that respect the proposed approach emphasizes the role of automation, thus enhancing the predictability in the findings of this usability evaluation. However, it must be noted that the evaluation team is always in charge of controlling the $QUTC_{KDD}$ process as a whole. Even when the $QUTC_{KDD}$ methodology enhances the decision making capabilities of the evaluation team, it does not replace them.

4 Experimentation in a Latin-American Context of Use

In this Section we will summarize the results obtained after using the proposed approach to evaluate the usability of the context of use formed by academic

web sites in Spanish-speaking countries in Latin America. In particular, the $QUTC_{KDD}$ process was carried out by processing qualitative information stored in 20,400 records collected by means of seven CWs. The experimentation was carried out within the 3rd Stage of the *UsabAIPO Project*,² a project focused on usability research which involved the participation of more than 15 university research groups specialized in HCI, in which four categories were considered (Design Category, Content Category, Navigation Category and Search Category). The evaluation team was formed by three usability experts and two CS advanced students, all of them frequent users of the context of use under evaluation.

First, a sample group of the context of use was selected by considering the web sites of the 69 universities listed in the *Universia* portal,³ a widely used portal about universities available for Spanish-speaking countries in Latin America. Second, CW [21] was selected as the *QUT* method M to be applied. Following the 2nd Stage of the *UsabAIPO Project* [9], three user profiles were considered to define M' : student, professor and administrative profiles. Besides, seven CWs involving 37 tasks were defined on the basis of an exhaustive poll carried out during four months between 400 different users of academic web sites of Latin-American universities. Eight questions related with qualitative features to be tested were linked to each task (two question for each Category). Next the definition of the seven CWs and the tasks associated with the CW #1 is shown:

- CW#1 (student profile): locate and visualize the study plan corresponding to a given undergraduate degree offered by the university under study./ CW#2 (student profile): locate and visualize information concerning academic regulations (enrollment information, etc.)/ CW #3 (administrative profile): locate and visualize information about a training course oriented towards the administrative staff and offered by the university/ CW #4 (administrative profile): locate and visualize data for contacting a person belonging to the University administration using the option of People Search/ CW #5 (professor profile): locate and visualize information about a posgraduate course or seminar offered by the university/ CW #6 (all profiles): locate and visualize email access facilities/ CW #7 (all profiles): locate and visualize a particular news made by the university.
- CW#1 (task description): *Task 1*: Visualization of types of degrees offered by the University (starting from the homepage); *Task 2*: Visualization of undergraduate degrees offered by the University (starting from the visualization of type of degree offered); *Task 3*: Visualization of study plan (starting from visualizing undergraduate degrees); *Task 4*: Short walkthrough using the study plan (courses, etc.).

Ranked values for users' cognitive effort were selected, namely (from bottom to top) *non_mesurable*, *insignificant*, *low*, *normal*, *high* and *very_high*. Besides

² See www.aipo.es (webpage of the Asociación Interacción Persona Ordenador – AIPO).

³ See www.universia.es

ideal values between *insignificant* and *normal* were related to every task. The WEKA platform [22] was adopted, as it provides the implementation for mining of ARs, along with an embedded Front-End with a very simple Command-line interface for posing queries and a visualization module to improve the interpretation of final results. Next, the evaluation team was divided in two sub-teams in order to carry out two independent usability evaluation of the 69 websites considered by applying the method M' described before. As a result, each sub-team obtained 10,212 qualitative answers that were stored in two temporary spreadsheets. During the preprocessing of the data stored in the two spreadsheets, each evaluation sub-team controlled the data produced by the other sub-team, and 24 data records were rejected as they had missing values. Afterwards both spreadsheets were condensed in a final spreadsheet called `Usability_Universities` which was automatically transformed into the Attribute-Relation File Format (`arff`)⁴ suitable for being processed by WEKA. Then, different queries were posted by the evaluation team to mine the data in `Usability_Universities`. General ranked list of ARs were computed using the Apriori algorithm provided by WEKA (with a support of 60%, a confidence of 90% and the selection of the best five rules obtained), each of them including in the consequent a high value associated with the general assessment of the usability problems involved in each CW. As an example, Figure 2 depicts the ranked list R#1 corresponding to the next query ($TxD-y$ represents the question y used to test the task x of the Design Category (D)):

```
GetRules (Usability_Universities) where
  [Consequent has {RC1_GeneralVal=very_high or RC1_GeneralVal=high}
  and support > 0.8 and confidence > 0.9]
```

Note that according to data in `Usability_Universities`, the information in R#1 shows: 1) the heterogeneous style in the webpages increments seriously the users' cognitive effort (AR#1), 2) the incorrect design of pull-down menus causes considerable visual noise rising the users' cognitive effort (AR#2), 3) the links' style is critic to define the users' cognitive effort (AR#3 and AR#4), 4) Information in the antecedent of AR#5 provides unbiased evidence showing that excessive animation (`T1D_PageStyle=low` and `T1D_Animation=very_high`) increases significantly users' cognitive effort and decreases heterogeneity in webpages' style. 5) In general, information in R#1 shows objectively that when values measuring users' cognitive effort increase in the last tasks of CW#1, the general assessing for CW#1 rises up to the highest possible ponderable values. Note that all the above knowledge (from 1) to 5)) was obtained from automatically mining the information stored in `Usability_Universities` by applying the APRIORI algorithm within the proposed $QUTC_{KDD}$ process.

At this stage the evaluation team was able to process the results obtained so far using the WEKA visualization module. Relationships among association rules in the ranked lists were visualized as 2-dimensional graphics. To give a formal account of the usability problem patterns that were obtained, the evaluation team rephrased them using a format similar to the one shown in Fig. 3.

⁴ See <http://www.cs.waikato.ac.nz/ml/weka/arff.html>


```

Minimum Support: 0,5
Minimum metric (confidence): 0.9
... ..
Best rules found:
1. T1D_MenuDesign=normal T3D_HomogeneousStyle=high 35==>RC1_GeneralVal=very_high 35 conf:(1)
2. T1D_Links=normal T1D_PullDownMenus=normal T2D_MenuDesign=high 40 ==> T2D_VisualNoise=high
   RC1_GeneralVal=very_high 42 conf:(0.95)
3. T1D_VisualNoise=normal T2D_PullDownMenus=high T3D_LinkStyle=high 34
   ==> RC1_GeneralVal=very_high 36 conf:(0.94)
4. T2D_LinkStyle=high T4D_PageStyle=high 34 ==> GeneralVal=very_high 37 conf:(0.91)
5. T1D_PageStyle=low T1D_Animation=very_high ==> T1D_HomogeneousStyle=high
   RC1_GeneralVal=high 22 conf:(0.90)

```

Fig. 2. Ranked list R#1 corresponding to Cognitive Walkthrough #1 and the Design Category. Third Stage of *UsabAIPO Project* (visualization provided by WEKA).

USAB. PROB PATTERN #1	Lack of an homogeneous style in different pages
FREQUENCY (1 TO 5)	4
JUSTIFICATION	AR#1, AR#4 and AR#5 in ranked list R#1
EV. TEAM COMMENT	Usability problem pattern observed in 35 webs
RECOMMENDATIONS	Include a study of the corporative image of the university and use results as guidelines for the webpage graphic design.

Fig. 3. Report of usability problem pattern. Third Stage of *UsabAIPO Project*.

Similar association rules were grouped and jointly rephrased. The frequency of every pattern was calculated on the basis of a weighted percentage which represents the number of rules supporting the pattern adjusted in function of the support and confidence of the rule. The explanation was stated by a list of alphanumeric characters identifying *ARs* and ranked lists supporting the usability problem pattern, as well as the corresponding comment of the evaluation team. An internal report describing the results was also written [8].

5 Conclusions. Related and Future Work

Identifying usability problem patterns in a given context of use is a challenging problem. However, *QUT* methods for qualitative usability evaluation are focused on considering particular systems, becoming inappropriate when analyzing a given context of use *C* as a whole on the basis of the joint analysis of a large number of systems in *C*. To cope with this situation, this paper presents the *QUTC_{KDD}* methodology, which is based on the integration of the traditional *QUT* methods and *KDD* techniques. Note that in *QUTC_{KDD}* intuitions that were usually informally stated by the evaluation team during the *QUT* process can now be appropriately analyzed in the context of *KDD*-based techniques. Note also that the evaluation can be enriched with the detection of hidden relationships among qualitative data that are documented with a formal basis.

To the best of our knowledge, there are no other approaches to extend *QUT* by incorporating *KDD* techniques as presented here. Typical models for usability evaluation under *UE* are not focused on the evaluation of contexts of use as a whole [2,4,7,11,13,14,18,20]. However, there exist some approaches which include

AR mining for assessing usability oriented towards the evaluation of a single system. For example, the AWUSA framework [19] presents an automatic tool for evaluating the usability of a web site by combining logging techniques and datamining, along with the static structure of the web site. Another approach where the *ARs* are used to test the usability of a single system is described in [1]. In this case, logging techniques are based on browsing activities performed by users. Techniques for statistical analysis (such as correlation) can also be applied to find relationships in usability testing on the basis of the available data. However, note that correlation refers to the departure of a group of variables from independence, but does not imply causality, as in the case of *ARs*. In addition, note that in our proposal the variables to be analyzed do not need to be identified explicitly, as they are found automatically by the *AR* mining algorithm.

Part of our future work is focused on testing different ranking functions for *ARs*, as suggested in [3]. To achieve this, the use of more powerful *KDD* platforms (e.g., KDDML platform [17]) should be crucial, as this kind of platform can be seen as an evolution of *KDD* engines like WEKA [17]. Another research line currently explored is the development of a usability evaluation module based on *QUTC_{KDD}* capable to be integrated in a *Usability Evaluation Management System*. Particularly, an integration with SketchiXML [5] is under consideration.

Acknowledgments. This work was supported by Projects TIN2004-08000-C03-03 (FEDER, European Union) and SGR-00881 (DURSI, Catalonian Government, Spain).

References

1. J. Alipio, J. Poças, and P. Azevedo. Recommendation with Association Rules: a web mining application. *Data Mining and Warehouses Conf. IS-2002*, 2002.
2. R. Bailey, R. Molich, J.Dumas, and J.M. Spool. Usability in Practice: Formative Usability Evaluations. *ACM CHI2002*, 2002.
3. D. Choir, B. Ahn, and S. Kim. Priorization of association rules in DM: Multiple criteria decision approach. *Expert System with Applications*, 29:867–878, 2005.
4. L. Constantine and L. Lockwood. *Software for Use. A practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley, 1999.
5. A. Coyette and J. Vanderdonckt. A Sketching Tool for Designing Anyuser, Anyplatform, Anywhere User Interfaces. In *Lecture Notes in Computer Science. Proc INTERACT 2005*, volume 3585, pages 550–564. Springer Verlag, 2005.
6. J. Demsar, B. Zupan, and G. Leban. Orange: From experimental machine learning to interactive data mining. Technical report, University of Ljubljana, 2004.
7. J. S. Dumas and J. C. Redish. *A Practical Guide to Usability Testing*. Intl. Specialized Book Service Inc, 2000.
8. M. P. González, J. Lorés, R. Gabandé, and T. Granollers. UsabAIPO Project. 3rd Stage. Technical report, AIPO Association, 2006.
9. M. P. González, J. Lorés, A. Pascual, and T. Granollers. Evaluación Heurística de Sitios Web Académicos Latinoamericanos dentro de la Iniciativa UsabAIPO. In *Proc. del VII Int. Interacción Persona-Ordenador*, 2006.
10. J. Han and M. Kamber. *Datamining: Concepts & Techniques*. M. Kaufmann, 2000.

11. Andreas Holzinger. Usability engineering for software developers. *Communications of the ACM*, 48(1):71–74, 2005.
12. T. Imielinski and A. Virmani. MSQL: A Query Language for Database Mining. *Data Min. Knowl. Discov.*, 3(4):373–408, 1999.
13. M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33:470–516, 2001.
14. D. J. Mayhew. *The Usability Engineering Lifecycle. A practitioner's handbook for user interface desing*. M. Kaufmann, 1999.
15. T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
16. D. Pyle. *Data Preparation for Data Mining*. M. Kaufmann, 1999.
17. A. Romei, S. Ruggieri, and F. Turini. KDDML: a middleware language and system for knowledge discovery in DB. *Data and Knowledge Eng.*, 57(2):179–220, 2006.
18. J. Scholtz. Adaptation of traditional usability testing methods for remote testing. In *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 5*, page 5030. IEEE Computer Society, 2001.
19. T. Tiedtke, C. Märtin, and N. Gerth. Awusa. A tool for automated website usability anlaysis. *9th Int. Workshop DSVIS*, 2002.
20. E. van Veenendaal. Low-cost usability testing. *Software Quality and Software Testing in Internet Times*, pages 153–164, 2002.
21. C. Wharton, J. Rieman, C. Lewis, and P. Polson. *Usability Inspection Methods*, chapter The Cognitive Walkthrough Method: A practitioners guide, pages 105–140. John Wiley & Sons Inc, 1994.
22. Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. M. Kaufmann, 2005.