

# Computer-Aided Usability Evaluation: A Questionnaire Case Study

*Elena García B.<sup>1</sup>*

*Miguel A. Sicilia<sup>2</sup>*

*José R. Hilera<sup>1</sup>*

*José A. Gutiérrez<sup>1</sup>*

<sup>1</sup>University of Alcalá

Cta. Barcelona km.33,600

Alcalá de Henares, Madrid. 28871. Spain

{elena.garciab, jose.hilera, jantonio.gutierrez}@uah.es

<sup>2</sup>Carlos III University

Av. Universidad, 30

Leganés, Madrid. 28911. Spain

msicilia@inf.uc3m.es

## SUMMARY

Computer-aided tools can be built to give support to different usability evaluation techniques, reducing some of their costs. These tools are complementary to existing fully automated ones, which are limited to the evaluation of external attributes. In this work, a generic model for questionnaire-based usability evaluation is described, along with the tool that implements it, which allows for fuzzy linguistic aggregation of opinions and provides support for results prediction based on similarity measures. Our tool is aimed at the exploitation of a growing database of evaluation facts with the help of various knowledge discovery and machine learning tools. Specifically, some preliminary applications of clustering and association mining techniques are described.

**KEYWORDS:** usability evaluation, computer aided engineering, machine learning, usability questionnaire.

## THE ROLE OF SEMI-AUTOMATED TOOLS IN USABILITY ENGINEERING

### Towards Computer-Aided Usability Evaluation Tools

Cost-justifying usability engineering, as one of the essential activities in software development, is a key topic in the adoption of a human-centred software design approach in business environments (a comprehensive discussion can be found in [2]). In many cases, the stringent requirements imposed on development cost and schedule result in usability as a non-first-priority requirement when developing software systems. One of the 'cost-items' that should be considered -at least in medium to large-scale usability evaluations- is that of the processing of the potentially large database of usability evaluation data. Given this scenario, the use of computer-aided tools would be helpful in the effort to promote usability (just as it is in other engineering areas) by providing a ready-to-use integrated set of tools and techniques.

We call those applications Computer-Aided Usability Evaluation (CAUE) tools (like in [4]). Questionnaires [19] drive the principal evaluation method that can be managed with the approach we present. The use of questionnaires requires user and expert intervention, and they can be performed in usability labs or conducted

remotely (with instrumented or semi-instrumented remote evaluation techniques [18]).

We think that some conditions are required if we want those kind of tools to become an important factor in the mainstream adoption of usability evaluation techniques, namely:

- Tools should be designed to be extensible for today and tomorrow usability practices.
- Tools should provide 'value-added' functionality that significantly increases the benefits of carrying out usability evaluations.
- Tools should enable the production of a baseline of usability facts that can be exploited by data analysis and knowledge management tools.

These premises are the main design objectives of our tool. We address extensibility defining abstract data models for evaluation artefacts (like questionnaires). Value-added tools are designed to make profit from existing artificial intelligence (AI) algorithms. And, finally, models are translated into conventional databases that are treated as operational data from which summaries and findings can be extracted.

## Related Work

We have labelled the tasks that can be carried out with our tool as semi-automatic, since they depend on user (and expert) intervention. Existing fully automated analysis techniques [16] can evaluate some internal attributes -following the definition in [3]- based on the parsing of Web pages or other kinds of media, but external attributes call for semi-automatic techniques that entail human intervention to some extent. Therefore, our model focuses in usability evaluation of external attributes, but it's capable of mixing the results of existing internal evaluation tools with external ones, (although this topic is not covered here). In this sense, it's complementary to other kinds of tools, like TANGO [15].

Some tools are available to prepare a usability questionnaire selecting it from a set of predefined ones (a very good example is [21]), but our approach, besides

this feature, enables the obtaining of results after the test itself.

### A Generic CAUE Tool Architecture

The central component of our generic architecture for an integrated CAUE tool is what we call Evaluation Repository (ER), and the following subsystems are built around it: Evaluation Performer (EP), Evaluation Miner (EM) and Evaluation Workbench (EW). The EP is responsible for the automated delivery of evaluations to evaluators and the gathering of their results, by using some kind of delivery mechanism like the Web. The EW is a graphical tool aimed at the efficient design of the evaluations that will be performed by the EP. Finally, the EM is a generic name for evaluation tracking and AI-based modules or data analysis tools. In this paper, we describe our first repository implementation for the just described generic architecture. The rest of this paper is organized as follows: the second section describes our current model and the first version of a CAUE tool for questionnaire evaluation. The third section describes some techniques used for the exploitation of evaluation results, and finally, conclusions and future work are presented in the last section.

## A QUESTIONNAIRE-BASED USABILITY EVALUATION MODEL AND IMPLEMENTATION

### A Generic Model for Questionnaire Evaluation

Usability evaluation questionnaires can be modelled by using a common abstract representation that can be easily implemented in a relational database. In what follows, we describe and reason about a model for questionnaire-gathered evaluation facts.

We'll define a set  $O = \{o_1, o_2, \dots, o_n\}$  of evaluation subjects (e. g. software systems, web sites). We consider these subjects as objects that can be further decomposed in aspects (e.g. the user registration section in a web portal), belonging to a set  $A = \{a_1, a_2, \dots, a_m\}$ . Aspects can be decomposed in lower granularity items, resembling the well-known *Composite* design pattern [11], which potentially represents a tree of objects (aspects in our case). An evaluation subject can be associated with zero, one or multiple 'root' aspects. A number of tasks can be suggested to evaluate each one of these aspects, for example, the task of creating a new web portal account.

A number of evaluators in  $E = \{e_1, e_2, \dots, e_s\}$  are asked for opinion about a number of aspects of a system (or about the system as a whole). Evaluators can be classified by expertise level or some other significant criteria belonging to their evaluator profile, denoted by  $\varepsilon(e_i)$ , which includes the history of previous evaluations. Their opinions are collected through questionnaires. These questionnaires are designed to evaluate some predefined criteria taken from a set of common ones  $C = \{c_1, c_2, \dots, c_r\}$ , including those directly extracted from

the ISO9126 usability definition, and others that can be found in usability literature.

A questionnaire is modelled as a hierarchically structured tree of questions in a set  $Q$ . Each question can be internally associated with one or more criteria. An evaluation fact is the response to a question, which internally can be characterized as a tuple:

$(e_i, o_j, a_k, q_h, \text{value}, \text{oid}, t) \in E \times O \times A \times Q \times V \times I \times D$ , where  $V$  represents a domain of evaluations (taken from one of the described below). It's assumed that a link between object  $o_j$  and aspect  $a_k$  exists. Since a questionnaire can be performed many times, on different evaluator subsets and applied to different subjects, the use of a field called *oid* is required to represent a specific application of a questionnaire. Note that the questionnaire is implicitly recorded due to the fact that, in our model, a question belongs to only one questionnaire.

Stored questions should be formulated in natural language in a context independent way. For that reason, when a questionnaire is delivered to evaluators, some re-writing could be needed.

### Types of Evaluation Fact Domains

In our system, we allow for the following evaluation domains: Likert, decimal and centesimal scales. Since Coleman's study [9] concludes that users prefer concrete adjectives for evaluation, we've included linguistic label sets as usability evaluation domain too. The former (domains in the form  $[a..b]$ , with  $a$  and  $b$  integer numbers and  $a < b$ ) covers most of the common cases of questionnaire design, and the latter case is provided for integration with fuzzy querying techniques. When using linguistic label sets, the questionnaire designer can define totally ordered label sets with odd cardinality  $T_g+1$ , in the form  $LS_g = \{s_i\}$ ,  $i \in \{0, \dots, T_g\}$  such that:

- the set is ordered:  $s_i \geq s_j$  if  $i \geq j$ ,
- there exists a negation operator:  $\text{Neg}(s_i) = s_j$ , such that  $j = T_g - i$ ,
- there exists maximization and minimization operators:  $\text{Max}(s_i, s_j) = s_i$  if  $s_i \geq s_j$ , and  $\text{Min}(s_i, s_j) = s_i$  if  $s_i \leq s_j$ .

After the evaluation, LWA and LOWA fuzzy linguistic operators are used for the (weighted or not) aggregation of linguistic evaluations to maintain the implicit vagueness of the labels. LOWA operator (and LWA, its weighted derivation, see [14]) has proven useful in practical decision-making situations. In addition, some properties about the rationality of its aggregation way are described in [13]. Usually, trapezoidal or triangular linear functions are used to capture the vagueness of the linguistic assessments. These functions are defined by fuzzy numbers (see, for example, [22]) in a somewhat

subjective way by the questionnaire designer. Of course, some rationality is assumed in the definitions.

### Case Study: Evaluation of Spanish Web Portals

We have conducted a usability evaluation of Spanish web portals as a proof of concept for our techniques (we have not attempted to be exhaustive nor to cover the whole Spanish portal industry and we haven't measured the validity and reliability of the new questionnaire we've designed since that's out of the scope of this work). We have studied the following portals: Inicia, Terra, Ya, Navegalia, Wanadoo, EresMas and MSN. The definition of the questions in the questionnaire was performed with the help of the Survey Designer, an instance of the previously mentioned Evaluation Workbench subsystem (see Figure 1). The questionnaire designer performs the following tasks:

- Define evaluation subjects. Since our evaluation domain is the Internet, URLs needs to be provided.
- Optionally, define a tree-like structure of aspects and associate it with subjects. In our case, we have taken into account the following aspects for each subject: Registry Process, Communities and Channels.
- Select usability evaluation criteria from predefined ones (or eventually, define new ones), and optionally, weight their relevance in the overall result of the evaluation. We have evaluated the above subjects according to four different usability factors: learnability, efficiency, user satisfaction and navigation control, all of them with the same weight in the overall result.
- Select an existing questionnaire or create a new one. In the latter case, the designer has to define the questions to be included in the questionnaire or select them from others that already exist in order to obtain the appropriate one. The expert can also define logical groups of questions to evaluate the different aspects of the subjects. In our example, we have made a new questionnaire with three logical groups of questions to evaluate the usability of the portals. A task is provided for each group of questions. The task is somewhat related to the questions, so that it helps the evaluator in answering. Tasks in our case study are the following: Create a new account in the portal, participate in a community about the Internet, and look for the weather in the south of our country. To reuse the questionnaire's items they must be written in a generic form. For example, the statement "I think these items are well-categorized" is written in a generic form and it's applicable to different aspects, like 'communities' or 'channels'. Although statements were formulated in a context independent way, they can be non-applicable to all the aspects.

The example above is not applicable to 'registry process'.

- Define the associations of each question with the criteria it evaluates, e.g., an assessment like "Using this web site for the first time is easy" is associated with the learnability criterion.
- Optionally, define the grade of similarity between the current questionnaire's items and other questionnaires' items, as described in the below section "A questionnaire outcome prediction tool".
- Select the target of evaluators. The expert can obtain a sample of representative evaluators using the appropriate tool option from the Evaluation Miner subsystem (see section "Clustering evaluators"), or retrieve by himself the evaluators from the database. If some expertise level or special characteristic is required, the designer can query for matching evaluator profiles. The test we have designed was answered by a wide range of users with different grades of expertise in Internet browsing.

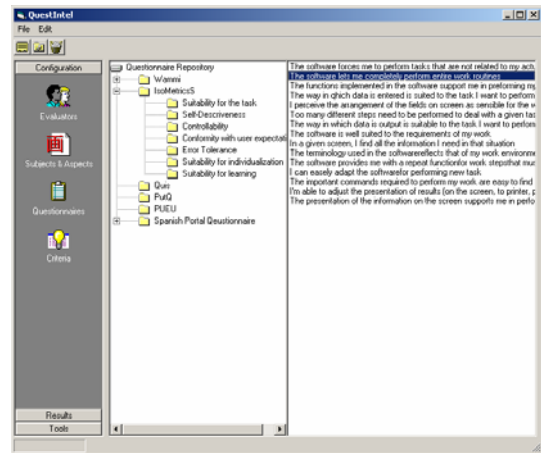


Figure 1: A Survey Designer screenshot

For the purpose of testing our assumptions, we stored versions of the following usability questionnaires<sup>1</sup>: QUIS [7], ISOMETRICS<sup>L</sup> and ISOMETRICS<sup>S</sup> [12], Purdue Usability Testing Questionnaire (PUTQ) [17], and Practical Heuristics for Usability Evaluation (PHUE) [20].

Once the parameters of the evaluation are defined, the Survey Performer (an instance of the generic Evaluation Performer) reads them and delivers the surveys to the previously defined evaluators. The delivery can be customized by specifying a start date and a deadline for each evaluator and subject, allowing for the change of sequence in the presentation of the questions for different individuals. Once completed, the expert can

<sup>1</sup> Although these instruments are available online, many are copyrighted and some require a fee for non-educational use.

examine the results and/or ask for additional facts extracted from the data of his/her current evaluation, or obtained by mining the entire evaluation baseline.

## EXPLOITING REPOSITORIES OF EVALUATION FACTS THROUGH AI TECHNIQUES

### A Questionnaire Outcome Prediction Tool

One common problem in the exploitation of the evaluation fact database is that results from different questionnaires are not directly comparable, due to the simple fact that they were designed to evaluate different criteria in different ways. Since our tool aims to reuse existing information, it partially overcomes this problem by providing a way to define similarities between some questions or sections (sets of questions) of two different questionnaires. Similarity measures are also useful in prediction of results according to previous questionnaire responses. Measures are often represented as values in [0,1] interval, which indicate a transitive degree of closeness (see [6]) between questions, and therefore enable the definition of imprecise relations between questions. Currently, our similarity measure for questions is defined by choosing one of the linguistic labels of a predefined label set (which includes terms like 'no resemblance', 'more or less the same' and 'completely equal'). Obviously, the quality of the prediction depends on the accuracy of the similarity grades defined by the expert. Two available tools are based on this information: (a) a user can ask for (partial) predicted results for another questionnaires, and (b) a questionnaire comparative table can be showed, which can be used as a 'synopsis' of questionnaires.

Note that scale conversions could be needed to accomplish this goal, and that losses of information are produced when conversions between scales of different granularity (different  $T_g$ ) are performed.

As an example, let's say we usually work with a questionnaire according to the Practical Heuristic for Usability Evaluation (PHUE) and we want to compare our results with others taken from Isometrics one. The expert only has to define a direct or transitive similarity measure between questions using linguistic labels, e.g. the statement "Learning to operate the system would be easy for me" (taken from PHUE) is 'completely equal' to "I don't need a long time to learn how to use the software" (taken from Isometrics). Given this, the tool can obtain a predicted outcome for the latter statement from the results of the former. In this case, the confidence in the prediction, that is implicit in the linguistic label, is high.

Confidence level can be aggregated if we want to predict the outcome according to a source set of previously answered questions that belongs to different questionnaires.

In order to maintain the implicit vagueness of the linguistic labels used for the comparison, we use the linguistic operators described above to perform aggregation.

### Applying Association Rules

The task of question selection for a usability questionnaire is driven by the previously defined criteria set  $C$ . Experts must decide and/or redefine the size of the set of questions related to any  $c_i$  in order to maintain the most suitable reliability/number of questions trade off. In this context, the identification of strong dependencies between question responses can be helpful in the decision of which questions can be removed from the questionnaire. We can say that a strong dependency exists between two questions  $q_i, q_j \in Q$  -taken from the same questionnaire- if:  $\forall e \in E, \forall o \in O (f_{e,o}^i \approx f_{e,o}^j)$ , where  $f_{e,o}^i$  stands for the list of evaluation fact values obtained from evaluator  $e$  by the application of question  $i$  (in a given questionnaire) to evaluate subject (or aspect)  $o$ . That definition can be extended to more than two questions.

A brute force approach for the extraction of these dependencies would require a nested iteration on evaluators and subjects, and the extraction of all the possible pairs of questions for each combination of them. The computational complexity of that approach is prohibitive, and the large volume of evaluation facts calls for some sort of exploratory analysis.

We have chosen Apriori algorithm (described in [1] and available in WEKA libraries [23]) for the extraction of association rules between questions. The first step before the mining process is the pre-processing phase. We first changed the relational database format of our evaluation facts to one in which each application of a questionnaire is represented in a tuple with questions as attributes. For example, a tuple taken from the application of a questionnaire with seven questions may look as follows: (low, very-low, low, high, high, low, low). Note that order is important, that is, the  $n^{th}$  attribute stores responses to the  $n^{th}$  question. Note also that evaluator and subject (and/or aspect) fields are omitted, since they're irrelevant for our purposes.

Then we need to carry out a second transformation to extract instances that can be directly taken as inputs by Apriori. For each of the just described tuples, we extract a set of 'boolean' tuples describing the different subsets of equal responses. For example, following the preceding example, responses one, three, six and seven would produce a tuple  $t1 = (true, false, true, false, false, true, true)$ , and responses four and five would produce another tuple  $t2 = (false, false, false, true, true, false, false)$ . A questionnaire with  $k$  questions would produce at most  $k/2$  tuples of booleans (with the symbol / standing for integer division), and each of these tuples

represents a positive fact about equality in subsets of the responses. Therefore, a table with boolean-valued attributes ( $q_1, q_2, \dots, q_k$ ) is given as input to Apriori. Since the number of tuples can be greater than the number of questionnaire responses, the support parameter<sup>2</sup> of the algorithm must be adjusted to a value slightly below:

$$\frac{1}{k/2}$$

Depending on the questionnaire, some adjustments could be needed to this parameter to allow for detection of noisy dependencies (as denoted by the symbol  $\approx$  above).

The algorithm produces a set of association rules in the form  $A \Rightarrow B$  where  $A$  and  $B$  are sets of pairs (attribute=value). Rules of arbitrary length are produced but we're only interested in those that contain exclusively 'true' values. That is, the rule  $q_1 = true, q_2 = false \Rightarrow q_5 = true$  has no meaning for our purposes (although it reflects a strong dependency in data). Therefore, we filter the output rule set to obtain a subset of rules in the form  $q_i \Rightarrow q_j$  where  $q_x$  stands for  $q_x = true$  for any  $x$ . Finally the extracted pairwise dependencies are made available to the expert through the *Survey Designer* interface, giving him/her the opportunity to discard or store the dependency as useful metadata about the questionnaire.

Note that redundancies are not the only information that can be discovered. We have chosen this one as a significant example, since our purpose here is only that of demonstrating the usefulness of our approach.

Note also that we can't use directly the result of the data mining process since it may contain association rules that have nothing to do with redundancy, and therefore, expert opinion is needed to assess the nature of the inferred rules. Despite this limitation, the automatic discovery of dependencies in databases of evaluation facts helps usability experts in gaining insight about the aspects their questionnaires are attempting to evaluate.

### Clustering evaluators

We've used clustering techniques to obtain natural groups of evaluators. Specifically, we used the implementation of COBWEB algorithm [10] available in WEKA libraries.

The attributes used to drive the clustering are extracted by aggregating the results of questions that are related to each criterion for each evaluator. We build a table with

---

<sup>2</sup> The support refers to the percentage of relevant tuples for which the pattern is true. Confidence, the other well-known Apriori parameter, can be set to a typical value, e.g. 90%

numeric attributes  $c_i$  obtained as summaries of the different evaluations. Each tuple is an instance that holds the summarized information of evaluation facts of a user in regard to each of the defined criteria. Linguistic summaries in  $T_g$  were converted into numeric values to allow the algorithm to take into account the order relation between the values. The outcome of the clustering process is a set of clusters that are usually not meaningful in a first attempt, in the sense that it is difficult to decide which were the clustering criteria in terms of domain knowledge, but the information extracted forms the basis for requests like "give me a neutral (non-biased) set of evaluators". An approach to answering this question is that of extracting a subset of evaluators by taking individuals for each of the clusters. COBWEB is based on the assumption that probability distributions on separate attributes are independent of one another, which could not be true in our case, and therefore, further experimentation is needed.

### CONCLUSIONS AND FUTURE WORK

Although questionnaires used in usability evaluation have many advantages, the process of elaborating and planning a questionnaire requires a lot of time and resources. The tool we have presented here enables the management of existing questionnaires and the design of new ones based on past experience. In addition, the questionnaires can be made available through the web, cutting off the costs associated with the manual gathering and processing of responses and the maintenance of usability lab facilities. Since the use of linguistic labels (that are closer to natural language expression than purely numeric scales) maintains an implicit vagueness level, we have used fuzzy aggregation operators in the data analysis process to reflect and propagate this vagueness of responses in the overall result. Similarity measures are useful when performing comparative studies on evaluation subjects, when reasoning about how different questionnaires address the same evaluation criteria, and when predicting a questionnaire result without responses for it. Those similarity measures, refined by the feedback of usability experts, are expected to provide the information fusion criteria for future applications of data mining techniques that operate on the results of different questionnaires at a time. Evaluator clustering helps in gaining insight about evaluators, and may provide useful data in outlier detection or in the assessment of possible biases in the evaluations. Although artificial intelligence has been used in usability evaluation [5] and a survey of application of machine learning techniques in human computer interaction can be found in [9], the topics covered here were not considered in that paper as future research areas. Nonetheless, we believe that our approach can provide useful extracted knowledge, which can be used in forthcoming evaluations, but to accomplish this goal, in addition to a well-structured repository to store the evaluations, a larger number of

them need to be performed. We're currently in the process of obtaining a very large base of evaluation facts, and testing other machine learning approaches (e. g. fuzzy clustering schemes). Future work will also extend the scope of evaluation techniques supported and the richness of value-added tools.

## References

1. Agrawal, R., Srikant, R. Fast Algorithms for mining association rules. In *Proc. of International Conference Very Large Databases VLDB'94* (Sept, 1994, Santiago de Chile), pp. 487- 499.
2. Bias, R. G., Mayhew, D. J. *Cost-justifying usability*. Academic Press, 1994.
3. Brajnik, G. Automatic web usability evaluation: what needs to be done?. In *Proc. 6th conference on Human Factors & the Web* ( June, 2000, Austin).
4. Chang, E., Dillon, T. S. Computer aided usability evaluation. In *Proc. of the International Conference on Interface and Virtual Reality INTERFACE'97* (May, 1997, Montpellier), pp. 56-62.
5. Chang, E., Dillon, T. S., Cook, D. An Intelligent system based usability evaluation metric. In *Proc. of IEEE Conference on Intelligent Information Systems IIS '97* (December, 1997, Bahamas), pp. 218-226.
6. Chen, G. *Fuzzy Logic in Data Modeling: Semantics, Constraints, and Database Design*. The Kluwer International Series on Advances in Database Systems. Kluwer Academic Pub, 1998.
7. Chin, J. P., Diehl, V. A., Norman, K. L. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proc. ACM Human Factors in Computing System Conference CHI'88* (May, 1988, Washington), pp. 213-218.
8. Coleman, W. D., Williges, R. C., Wixon, D. R. Collecting Detailed user evaluations of software interfaces. In *Proc. of the 29th Human Factors Society Annual Meeting* (1985), pp. 240-244.
9. Finlay, J. Machine learning: a tool to support improved usability?. In *Proc. Workshop Machine Learning meets Human-Computer Interaction ICML'96*. (1996), V. Moustakis and J. Herrmann, eds, pp. 17-28.
10. Fisher, D. *Knowledge acquisition via incremental conceptual clustering*. Machine Learning, Vol. 2, No. 2, , 1987, pp. 139 –172.
11. Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns. Elements of Reusable Object Oriented Design*. Addison Wesley Pub Co, 1995.
12. Gediga, G., Hamborg, K.C., Düntsch, I. *The IsoMetrics usability inventory: an operationalisation of ISO 9241/10*. Behavior and Information Technology, Vol. 18, 1999, pp. 151-164.
13. Herrera, F., Herrera-Viedma, E.,Verdegay, J. L. Aggregating Linguistic Preferences: Properties of LOWA Operator. In *Proc. 5th IFSA World Congress* (1995, Sao Paulo), pp. 153-156.
14. Herrera F., Herrera-Viedma, E. *On the linguistic OWA operator and extensions*. The ordered weighted averaging operators: Theory, Methodology, and Applications. R. R. Yager and J. Kacprzyk eds., Kluwer Academic Publishers, 1997
15. Ivory, M.Y. Web TANGO: Towards Automated Comparison of Information-centric Web Site Designs. In *Proc. of ACM Conference on Human Factors in Computing Systems CHI'00* (April, 2000, The Hague Netherlands), Poster Session.
16. Ivory, M., Hearst, M. *State of the art in automated usability evaluations of user interfaces*. Available from <http://www.cs.berkeley.edu/~ivory/research/web/survey.ps>
17. Lin, H. X., Choong, Y. Y., Salvendy G. *A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems Usability Evaluation Methods*. Behaviour and Information Technology, Vol. 16, No. 4/5, 1997, pp. 267-278.
18. Mayhew, D. J. *The Usability Engineering Lifecycle*. Morgan Kaufmann, 1999.
19. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, 1993.
20. Perlman, G. Practical Usability Evaluation. In *Proc. of ACM Conference on Human Factors in Computing Systems CHI'97* (March, 1997, Atlanta), pp.168-169.
21. Perlman, G. *Web-Based User Interface Evaluation with Questionnaires*". Available from <http://acm.org/~perlman/question.html>
22. Tsoukalas, L., Uhrig, R. E. *Fuzzy and Neural Approaches in Engineering*. John Wiley and Sons, 1996.
23. Witten I. H., Frank E. *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000.