# Vizability: A Tool for Usability Engineering Process Improvement through the Visualization of Usability Problem Data

Pardha S. Pyla, Jonathan R. Howarth, Chris Catanzaro, Chris North
Center for Human-Computer Interaction and Dept. of Computer Science
Virginia Tech
Blacksburg VA, USA
{ppyla, jhowarth, ccatanza, north}@vt.edu

## ABSTRACT
We present Vizability, a usability engineering tool that is motivated by the variability of analysis processes and facilitates the analysis of existing problem data with the goal of process improvement. Usability engineers spend considerable resources performing usability testing and analyzing the resulting data, but their work is often limited to a single development effort or a small set of similar efforts. To extend the analysis process to multiple diverse efforts, it is necessary to categorize and store data in a consistent manner and have techniques for discovering patterns in that data. Our tool is designed to work with problem data that has been organized according to a hierarchical framework of usability concepts, which ensures consistency through completeness and precision. In addition, our tool helps engineers discover weaknesses in their process through exploratory browsing of a visualization of the tree structure and visual filtering based on cost, importance, and keywords.

## Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Evaluation/methodology, Screen design (e.g., text, graphics, color), Prototyping, and User-centered design

## General Terms
Design, Human Factors

## Keywords
Process Improvement, Usability Testing and Evaluation, Usability Research

## 1. INTRODUCTION
In this paper, we present a tool for analyzing usability data. A main component of the tool is the visualization of a tree structure of usability concepts, but our work is primarily intended to be in the area of usability and not visualization. In particular, our tool is used to analyze a collection of usability problem data that has been organized according to a hierarchical framework of usability problem concepts. We do not present this tool as a substitute for existing tools, but instead intend it as a complement to be used for additional analysis.


*ACMSE'06*, March 10–12, 2006, Melbourne, Florida, USA.


Before discussing the tool in any detail, we present background information to explain our motivation and position our work with respect to previous work. In the *Process Improvement* section we introduce our concept of process improvement and the key motivation behind development of the tool. Only with an understanding of their existing usability engineering process can engineers refine and improve it in future efforts. In the *Variability of the Analysis Process* section we discuss the ad hoc nature of existing usability engineering processes and the limitations of the resulting data. We discuss how our tool directly addresses this variability and the potential benefits offered as a result. In the *Tree Structure* section we describe our motivations for using a tree structure of usability concepts and give an example of diagnosing problems with it. In the *Visualization* section we address the use of visualization and how our approach differs with respect to previous approaches. In the remaining sections, we discuss the tool; we first provide an overview of the main areas of the tool and their uses in the *Vizability* section and then discuss an application of the tool to usability data from a documented evaluation of a digital library system in the *Exploratory Example* section. Finally, in the *Future Work* section we describe our plans for this tool.

## 2. PROCESS IMPROVEMENT
Our tool for analyzing usability data can be used not only to improve the usability of a specific development effort, but more importantly to track and guide the entire usability engineering process as it grows and changes across multiple efforts. A variety of tools already exist for doing analysis, so why is another one necessary? We do not promote our tool as a replacement for existing tools, but instead as a complement that provides a more long term approach to usability engineering. Andre, et al. identify the ad hoc nature of processes that are currently in use to analyze and report usability problems [2]. These processes may be effective for improving a given development effort, but because they are not consistent and do not facilitate reuse by cataloging data, they are not useful for monitoring trends across multiple efforts.

Having an understanding of usability methods as observed across multiple efforts is essential for refining, improving, and calibrating the usability engineering process. Without a complete understanding of existing methods, their advantages, and shortcomings, usability engineers may continually attempt to develop new practices for analyzing usability data without knowledge of whether these methods are more or less effective than previous ones. In addition, cataloging of usability process data, when shared, will benefit the usability industry as a whole by serving as reference and knowledge store.

# 3. VARIABILITY OF THE ANALYSIS PROCESS

In its current state, usability analysis is not precise, and the recommendations that result from an analysis are not always easy to understand or apply. As an example, in the first comparative usability evaluation study Molich, et al. describe the usability analysis of an application as performed by four different labs [9]. The variance in problems found and documentation methods used led to the production of four different reports, which had different interpretations of the usability of the application.

Our tool is motivated by the need to limit the variability of the analysis process; it focuses on the visualization of usability problems, which have been associated with nodes in a tree structure of usability concepts. It is only through limiting variability that the results of different usability engineering efforts can be combined to provide information on how a usability engineering process is developing over time. With our tool, we target three main issues that we believe contribute significantly to the reduction of variability in the analysis process: systematically analyzing problems, identifying pertinent problems, and improving communication between usability engineers and software developers.

## 3.1 Systematically Analyzing Problems

Nayak, Mrazek, and Smith emphasize the need for a systematic process for analyzing data because such a process improves consistency and understanding [10]. Redish, et al. discuss reasons why individuals, software developers in particular, sometimes dismiss usability engineering as deceptively easy and minimally effective [14]. A main reason for this lack of credibility is the often divergent interpretations of a given problem and its solutions produced by the variety of existing discount methods, guidelines, and rules of thumb. A consistently applied approach would limit the number of interpretations, thereby making them more credible. A more systematic process would also promote a better understanding of usability problems. Different analysis methods may be better suited for different types of applications, but a unifying method is necessary for extending analysis beyond a single application to collections of applications. A systematic process would improve understanding over time by allowing application producers to track their development efforts and determine which problems reoccur and which problems have been corrected. Our tool depends on the structuring of diagnoses according to a hierarchical framework of usability concepts, which limits interpretations and extends diagnoses across applications.

## 3.2 Identifying Pertinent Problems

Determining what problems are pertinent and by whom they would be best addressed are important problems for usability analysis. Different users are interested in different aspects of a system; consequently, it is important to allow users to quickly locate information that is relevant to them. In [14], Redish, et al. discuss some shortcomings of traditional usability reports. In many cases, the members of the target audience will simply ignore a report because it is too long or too dense or even because it identifies too many insignificant problems. The visualization functionality of our tool allows users to quickly browse for information that they need and ignore information that is irrelevant or impossible to fix given time and resource limitations.

## 3.3 Improving Communication between Usability Engineers and Software Developers

Another issue is improving communication between usability engineers and software developers. This communication link is essential to improving the precision of the analysis process. As Pyla, et al. and Pyla, et al. discuss, communication between these two groups is essential to the development of a system that is pleasing and useful to the user as well as efficient [12, 13]. In addition, effective communication saves time by reducing inconsistencies in understanding thereby reducing the number of areas for debate. Oftentimes usability engineers, lacking a thorough understanding of the functional system, suggest fixes for problems that developers cannot implement within budget, time, quality, or architectural constraints. The opposite situation occurs when developers create systems that have poor usability or fix non-critical aspects of the interface because they do not understand the recommendations provided by the usability engineers. Our tool brings together the skills and knowledge of both parties through cost and importance rating systems and promotes precision and completeness as a result of the diagnoses process.

# 4. TREE STRUCTURE

Our tool is designed to work with a tree structure that has as its nodes usability concepts; usability problems are diagnosed and associated with these nodes. We use the User Action Framework (UAF) [4], which is based on Norman's Stages-of-Action Model [11], as the basis for the prototype tool discussed in this paper. Tree structures provide precision and completeness for the diagnosis process; it is through a consistent diagnosis process that the variability of usability analysis can be reduced.

## 4.1 The Multidimensional Space of Problem Data

The space for usability problems can be described as multidimensional. A tree structure allows a user to navigate the dimensions of the space, ultimately arriving at a specific dimension. Each level of the tree structure maps to a dimension, and each node in the tree maps to an attribute. Selecting one of the nodes at given level is equivalent to removing unnecessary attributes, thereby filtering or removing dimensions. Traversing the tree is equivalent to repeated node selection and results in the identification of a specific dimension that contains the problem.

## 4.2 Completeness and Precision

The diagnosis process consists of navigating the tree to find a node that maps to a given problem description. Once a problem has been associated with a node, the path to that node contains all the information needed to specifically identify the problem. Completeness is ensured because only one path leads to a given node. Precision is also ensured because other problems that have the same attributes will be placed in the same node.

## 4.3 A Natural Organization Device

In addition to providing completeness and precision, a tree is a natural way to organize usability problem data. Existing techniques for understanding data include affinity diagrams, priority ranking, and Pareto diagrams; such techniques require grouping data and have the ultimate goal of organization [10]. Trees provide the same functionality, but do so with a structure that can be reused in future development efforts.

## 4.4 Facilitating Redesign

A final benefit of trees is that they organize problems in a way that facilitates the identification of design changes. Nayak, Mrazek, and Smith discuss how techniques that are easy to translate to solutions increase team acceptance [10]. By completely specifying a problem, the tree allows developers to understand the specific causes of the problem and the changes necessary to correct it. Through time, developers can associate generic solutions with nodes and increase the speed of the correction process.

## 4.5 An Example Diagnosis

We give an example of diagnosing a problem using the UAF to illustrate our motivation for using a tree structure. A detailed discussion of the background of the UAF and using it for problem diagnosis can be found in [2, 4, 8], but a simple example will suffice for this paper. The following problem description was extracted from a critical incident report for a database system that was evaluated in our institution:

"A database user accidentally deleted a number of related records. The user knew that it was possible to back out of this operation and correct the error, but the system did not help in finding a way to do it. There was a button, labeled 'Back' for recovery from deletion, but the user was looking for something like 'Undelete' or 'Undo' and did not make the connection."

In particular, the UAF is based on user's interaction behavior. The diagnosis would place this problem in the node at the end of the following path, where each node along the path represents a usability concept:

- Translation (design helping a user know what to do)
- Content, Meaning (of a cognitive affordance)
- Clarity
- Precise Use of Words
- Labels for Buttons, Menus

The diagnosis completely specifies the dimension that contains the problem, and because the structure of the UAF is constant, other problems in the same dimension will be associated with the same node.

# 5. VISUALIZATION

In this section, we discuss our use of visualization. We begin by describing previous work on visualization for the usability analysis process and discuss why we believe our Vizability tool is unique. We then describe the benefits of visualization as they relate to our work.

## 5.1 Previous Work with Visualization

Visualization has been used in the usability analysis domain, but not at the level we propose. Previous efforts have focused on very low-level data, but our approach visualizes information that is essentially at a higher, more abstract level. It is not the methods used in our visualization that are novel, but the target of those methods.

Previous work with low-level data has focused on visualizing events such as keystrokes and mouse clicks that are performed by users as they interact with an interface. Such events certainly have the potential to be useful in evaluating an interface, but they may be difficult to interpret correctly. Ivory and Hearst discuss the three common activities of usability evaluation: capture, analysis, and critique [7]. Data capture may be partially or completely automated, but accurate analysis and critique often require a thorough review of the data by a usability engineer. However, the volume of data produced by low-level events, particularly when capture is automated, can be overwhelming. Visualization tools such as those presented in [3], [6], and [15] are necessary because they help order events, associate user-initiated events with system events, and highlight patterns. Using Ivory and Hearst's classification, these visualizations focus on the data capture part of the usability evaluation process.

With our work, we focus on the visualization of usability problem diagnoses made by usability engineers. A diagnosis involves isolating a distinct usability problem from observational data and documenting that problem according to a predefined method or system. The tree structure provides a way of systematizing the diagnosis process. A usability problem diagnosis represents a higher level of understanding of a problem and is not as prone to the incorrect inferences or contextualization that may complicate usability studies focusing on low-level data [3]. Diagnoses, particularly when performed by usability engineers using a tree structure, promote consistency. In addition, because they are at a higher level, usability problem diagnoses encourage knowledge sharing and reuse.

## 5.2 Why Visualization

The visualization of problem diagnoses is a central component of our tool. We emphasize its role because it offers unique capabilities and opportunities for both the analysis process and the process of making recommendations to fix problems.

The analysis process consists of understanding the quantity and type of existing problems as well as their effects on the overall usability. Traditionally, tools such as cost-importance tables have been used for the analysis process. Such tools benefit a single development effort and help with the creation of a more usable product by identifying what to fix given a limited amount of resources. However, these tools do not necessarily aid in the discovery of trends or reoccurring problem areas across multiple development efforts. Visualization of a tree structure in which the nodes contain problem diagnoses complements these tools by facilitating the rapid identification of patterns in a given development effort or across a group of efforts [15].

Instead of directly querying a set of usability data, users progressively refine the data set and can easily identify outliers and selectively view details [1]. Such functionality is desirable, particularly in larger development efforts that contain a large number of problem diagnoses. Through the identification of clusters of related problems, users can better understand the strengths and weaknesses of their development effort. Visualization allows users to better understand the development effort by providing an overall picture. Having a better understanding is critical in making recommendations to fix problems.

Visualization helps users target the problem areas that are most relevant to their specific needs or talents. For example, a graphic designer can filter a data set to determine the number and concentration of problems related to contrast or layout issues and make recommendations for fixing those problems. Visualization also helps to identify and address problem areas in the development process. For example, users may discover that a large percentage of their diagnoses are clustered in a branch of the tree that addresses wording. With this understanding, a technical writer can be hired to fix the current problems and to assist in future efforts.

# 6. VIZABILITY

The Vizability tool is a Java application that uses as its data source a Microsoft Access database containing the UAF and usability problems. Figure 1 is a screenshot of the tool. The tool has three basic areas: the selection or criteria area on the left, the tree view area in the center, and the problem list area on the right.

## 6.1 Selection Area

The selection area is the left pane and contains range selecting sliders for cost and importance and checkboxes for keywords and evaluators. Each of these selection controls can be used to query the set of usability problems. The user is provided immediate feedback via the percentage bars associated with each of these controls that show the percentage and the fraction of total problems that match the selected values or value range of the control.

Cost represents the person hours necessary to fix a usability problem in terms of software design and implementation. The screenshot shows a scale of 1 to 20 hours, but this scale can be customized based on the set of usability problems. Importance represents the benefit of fixing a problem to the overall system in terms of usability. An importance of 1 indicates minimal benefit, while an importance of 5 indicates that a fix is required. The cost and importance ratings require communication between software developers and usability engineers. In particular, software developers familiar with the design and implementation of the system are responsible for supplying the cost ratings, and usability engineers are responsible for supplying the importance ratings.

Keywords represent another way of understanding data in the tree. Keywords are associated with nodes in the UAF; selecting a keyword will select all usability problems in nodes that contain the keyword. Possible keywords include wording, layout, and system model.

Evaluators are individuals that have analyzed data from usability

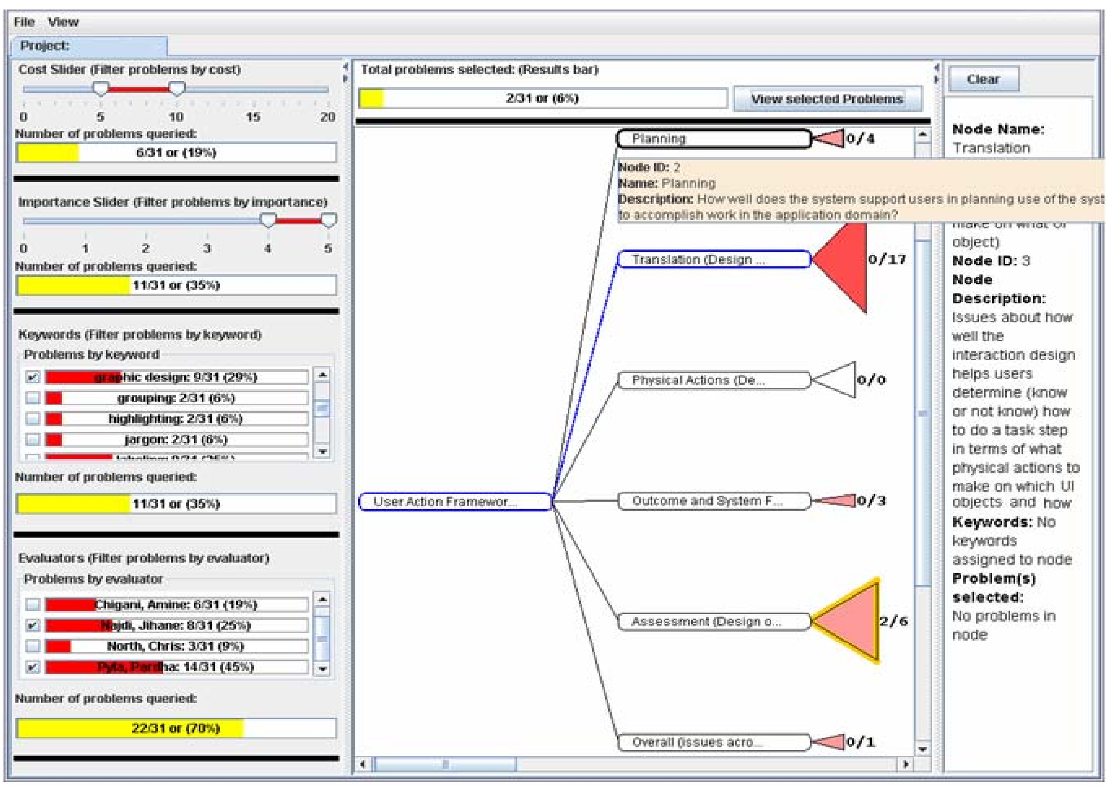


**Figure 1. Screenshot of Vizability**

evaluations and extracted and diagnosed problems. Selecting an evaluator will select all usability problems diagnosed by that evaluator.

An "and" operation is implied among the fields in the selection area; selecting certain combinations of values will allow a user to filter the set of usability problems. The selection area in Figure 1 shows a complex query to search for problems which take between 5 and 10 person hours to fix, are of high importance (4 and 5), are related to "graphic design" and "support for user goals" (not shown in figure), and have been identified by evaluators "Jihane Najdi" and "Pardha Pyla".

### 6.2 Tree View Area

The tree view area is a visualization of the UAF tree; each node is represented by a rounded rectangular object, and parent nodes are connected to their child nodes with thin black lines. Clicking on a node will expand the node to reveal its children or collapse the node to hide its children. Such a feature is useful for focusing on a particular aspect of user interaction. One or more nodes in the tree can be selected by shift-clicking on them. Clicking on a node highlights the path from the root node to the clicked node. In Figure 1, "Translation (Design helping user know what physical action to make on what UI object)" is selected as shown by the blue highlighting.

In a collapsed state, each node is shown with a triangle on the right hand side. The height of the triangle represents the depth of the sub-tree and the width represents the total number of nodes in the sub-tree beneath that node. The color intensity of the triangle encodes the number of problems in the sub-tree; the darker the color, the greater the number of problems. The fraction M/N at the base of the triangle provides an exact picture of the problems in the sub-tree; M shows the number of usability problems matching the query in the selection area and N shows the total number of usability problems in the sub-tree. The triangles are highlighted in yellow if a query selects one or more nodes in the sub-tree. For example, in Figure 1, the triangle next to the "Assessment" node is highlighted to show that two problems match the query described in the selection area. The top part of the tree view area has an aggregate query results percentage bar. This bar shows the fraction and percentage of the total problems from the tree that match the query in the selection area. In Figure 1, only two out of 31 possible problems match the query.

The description of any node that is currently displayed in the tree view area can be accessed via tool-tips. Figure 1 shows the tool-tip for the Planning node. When the mouse pointer is placed over a node (without clicking), the node is highlighted using a thick black border. This secondary highlight is only for accessing the tool-tip but not for selecting the node.

### 6.3 Problem List Area

The problem list area on the right hand side is a detail view that contains textual descriptions of the problems that are present in nodes selected in the tree view. The problem list area is useful for understanding a collection of problems at a lower, more detailed level. Information about the path and the actual node is essential for an understanding of deficiencies in a given effort or collection of efforts, but the problem entries are essential for understanding errors at the level of detail necessary to fix them.

## 7. EXPLORATORY EXAMPLE

As an exploratory evaluation of our tool, we used real usability problem data from a digital library case study documented in [5]. The authors collected a set of usability problems while performing a usability inspection of the web-based system. They used cost-importance analysis to determine which problems could be fixed given a limited amount of time and resources. Because there were a limited number of problems, they were able to manually group and visualize the type and distribution of those problems. Had the system been larger or had a number of problems in the hundreds or thousands, manually grouping and visualizing the problems would have been prohibitively expensive.

The study as performed by the authors was useful in determining how to fix the most pertinent problems, but it gave little information as to overall trends. We decided to use our tool to visualize the data so as to gain a better understanding of the big picture. We diagnosed the problems from the case study according the usability concepts of the UAF; there were 31 problems in total. We assigned costs and importance to the problems based on information in the problem descriptions themselves as well as our knowledge of web-based systems. When we visualized the information in our tool, clusters of nodes helped us easily identify deficiencies in the system. In particular, we noticed that the majority of the problems were clustered under a node that addressed the content and meaning of cognitive affordances.

Additional exploratory browsing with filters on cost, importance, and keywords helped us to discover that the majority of problems were of low cost and medium to high importance. We were also able to discover which problems were absolutely necessary to fix and which problems were too expensive. For example, we discovered that the 4 problems in nodes that were children of the Planning node were necessary to fix to ensure that the user understood the system model. Also, we discovered that the majority of problems dealing with wording were of relatively high importance with low cost to fix. By using our tool, we realized that future efforts by the developers of this digital library system would benefit from the inclusion of a technical writer or an easily accessible style guide containing naming conventions for interface elements such as labels and buttons.

## 8. FUTURE WORK

We have taken an iterative approach to the development of the tool. As of the submission of this paper, we have created six versions of the tool. Once we resolve outstanding issues, we plan to submit the tool to review by graduate usability engineering students. Finally, we plan to submit the tool to testing by professional usability engineering practitioners as one of a suite of usability engineering tools being developed in our institution.

## 9. CONCLUSION

We have presented a tool for visualizing usability problem data that is organized according to a hierarchical framework of usability concepts. Our tool focuses on process improvement by enabling the exploratory browsing of results from usability evaluations of multiple development efforts. The tool facilitates the identification of patterns and trends so as to increase understanding of strengths and weaknesses in the usability engineering process.

## 10. REFERENCES


[1] Ahlberg, C. and Wistrand, E. IVEE: An Information Visualization and Exploration Environment. *IEEE InfoVis*, 1995, 66-73.

[2] Andre, T.S., Hartson, H.R., Belz, S.M. and McCreary, F.A. The User Action Framework: A reliable foundation for usability engineering support tools. *International Journal of Human-Computer Studies*, 54, 1, 2001, 107-136.

[3] Gray, M., Badre, A. and Guzdial, M. Visualizing usability log data. *IEEE Symposium on Information Visualization*, 1996, 93-99.

[4] Hartson, H.R., Andre, T.S., Williges, R.C. and Rens, L.v. The User Action Framework: A theory-based foundation for inspection and classification of usability problems. *HCI International (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Ergonomics and User Interfaces*, Lawrence Erlbaum Associates, Inc., 1999, 1058-1062.

[5] Hartson, H.R., Shivakumar, P. and Pérez-Quiñones, M.A. Usability inspection of digital libraries: A case study. *Journal of Digital Libraries Special Issue on Usability of Digital Libraries*, 4, 2, 2004, 108-123.

[6] Holm, R., Priglinger, M., Stauder, E., Volkert, J. and Wagner, R. Automatic data acquisition and visualization for usability evaluation of VR systems. *Eurographics*, 2002

[7] Ivory, M.Y. and Hearst, M.A. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33, 4, 2001, 470-516.

[8] Keenan, S.L., Hartson, H.R., Kafura, D.G. and Schulman, R.S. The Usability Problem Taxonomy: A framework for classification and analysis. *Empirical Software Engineering*, 4, 1, 1999, 71-104.

[9] Molich, R., Bevan, N., Curson, I., Butler, S., Kindlund, E., Miller, D. and Kirakowski, J. Comparative evaluation of usability tests. *Usability Professionals Association 1998 Conference*, 1998.

[10] Nayak, N.P., Mrazek, D. and Smith, D.R. Analyzing and communicating usability data: now that you have the data what do you do? *ACM SIGCHI Bulletin*, 1995, 22-30.

[11] Norman, D.A. Cognitive engineering. In Norman, D.A. and Draper, S.W. eds. *User centered system design: New perspectives on human-computer interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986, 31-61.

[12] Pyla, P.S., Pérez-Quiñones, M.A., Arthur, J.D. and Hartson, H.R. Towards a model-based framework for integrating usability and software engineering life cycles. *Interact 2003 Workshop on "Closing the Gaps: Software Engineering and Human Computer Interaction"*, Université catholique de Louvain, Institut d' Administration et de Gestion (IAG) on behalf of the International Federation for Information Processing (IFIP), 2004, 67-74.

[13] Pyla, P.S., Pérez-Quiñones, M.A., Arthur, J.D. and Hartson, H.R. Ripple: An event driven design representation framework for integrating usability and software engineering life cycles. In Seffah, A., Gulliksen, J. and Desmarais, M. eds. *Human-Centered Software Engineering: Integrating Usability in the Development Process*, Kluwer HCI series, 2005.

[14] Redish, J.G., Bias, R.G., Bailey, R., Molich, R., Dumas, J. and Spool, J.M. Usability in practice: formative usability evaluations - evolution and revolution. *CHI '02 Extended Abstracts on Human Factors in Computer Systems*, ACM Press, 2002, 885-890.

[15] Wesson, J. and VanGreunen, D. Visualisation of usability data: measuring task efficiency. *2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology*, South African Institute for Computer Scientists and Information Technologists, 2002, 11-18.