The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure

David E. Rowley and David G. Rhoades

Varian Associates
2700 Mitchell Drive
Walnut Creek, CA 94598
510-945-2275
rowley@varian.com

ABSTRACT

Walkthrough techniques have been shown to be an effective supplement to empirical testing methods for evaluating the usability of software systems [3, 4]. Unfortunately, structured walkthrough procedures tend to be time-consuming and unpopular with evaluators when used on substantial tasks. To maximize the useful information obtained from walkthroughs while minimizing the overhead of the procedure itself, a fast-paced methodology was developed and used within the constraints of a real-world product development environment. By using video recording equipment and an informal, interactive evaluation session, the "cognitive jogthrough" procedure revealed significant user interface problems that could then be studied using other techniques.

KEYWORDS: User interface evaluation techniques, structured walkthroughs, design methodologies.

INTRODUCTION

This paper describes the in-house development of a user interface evaluation procedure at Varian Chromatography Systems. This procedure was used to evaluate the proposed user interface of a software application package being developed for an existing pc-based chromatography workstation that we manufacture.

Our chromatography workstation is used by industry and regulatory agencies to determine and control the composition of materials such as pharmaceuticals, pesticides, and petrochemicals, and to monitor for the presence of compounds of concern in the environment. It consists of a set of software applications based on Microsoft® Windows 3.0™. These applications control instruments that perform automated experiments to separate and quantify the individual components of the chemical samples, process the raw data, and generate calculated results in a form which is useful to the chromatographer. Each iteration of this process is termed a *chromatographic run.* In addition to the software that performs these core workstation tasks, application packages may be utilized to perform calculations and generate results that are required by a particular market segment. These application-specific programs may be invoked automatically at the end of each chromatographic run, or interactively by the operator.

One of the primary tasks of the chromatographer is to develop methods that specify and control the actions described above (see figure 1). A *method* is the set of instructions required to perform a single chromatographic run. In a typical methods development scenario, the chromatographer first determines the instrument hardware conditions which result in a satisfactory chromatographic separation of the sample components. Once this has been achieved, the chromatographer specifies the data handling parameters which control processing of the raw data and the basic results calculations. Finally, parameters which control any application-specific calculations and report generation are specified. As shown in the figure, each of these steps consists of specifying the instructions, invoking the actions, and inspecting the textual and graphical results. Chromatographers utilize heuristics derived from knowledge of the application domain to determine whether the results are satisfactory, or whether additional iterations are required.

It has been our experience that users prefer to do methods development in an interactive software environment in which the chromatographer can rapidly perform these separate but related actions in an iterative manner until satisfied with the results. This is facilitated when the individual applications that provide the various chromatography workstation services are easily accessible, interact seamlessly, and are consistent in their look and feel. We have established a user interface

evaluation team at our site as a means of working towards these goals. The team is composed of user interface designers, software engineers and in-house "users" who are knowledgeable in the tasks of our target markets. We feel the multi-disciplinary nature of the team is important to provide both a customer perspective and experience in graphical user interface design.

The team's roles are to maintain and to further develop interface consistency in the workstation software, and to ensure that the user interface of the system does not interfere with the work flow of the user. This is especially important when individual applications are developed by different software engineers, and when new applications are added to an existing product. The team develops project-wide guidelines, templates, and examples, and it evaluates interface proposals for new applications and modifications to existing ones. It also serves as a mechanism for ongoing communication during the development cycle between the software developers and the in-house representatives of the end users.

Since the notion of Human Factors activities as an integral part of the software development process is relatively new in our organization, no full time Human Factors specialists exist. User interface evaluation is a supplementary activity for all of the participants on the team, and therefore the time that can be devoted to evaluations is limited. This turns out to be a key constraint on the activities of the group and has had a great influence on our evaluation technique.

EVALUATING A PROPOSED USER INTERFACE

We set out to perform user interface evaluations with several goals in mind. Primarily, we wanted to identify problems in the user interface which could either be resolved immediately during the evaluation session, or designated as a separate design task. We wanted to develop a procedure that would be suitable both for evaluators with little understanding of the application, and users with little understanding of evaluation techniques. It was important for the procedure to provide a mechanism for a thorough review, with enough structure to avoid tangential diversions, but flexible enough to allow us to skip areas beyond the scope of our influence. The documentation of the procedure was also important as a way to capture significant information brought out by the

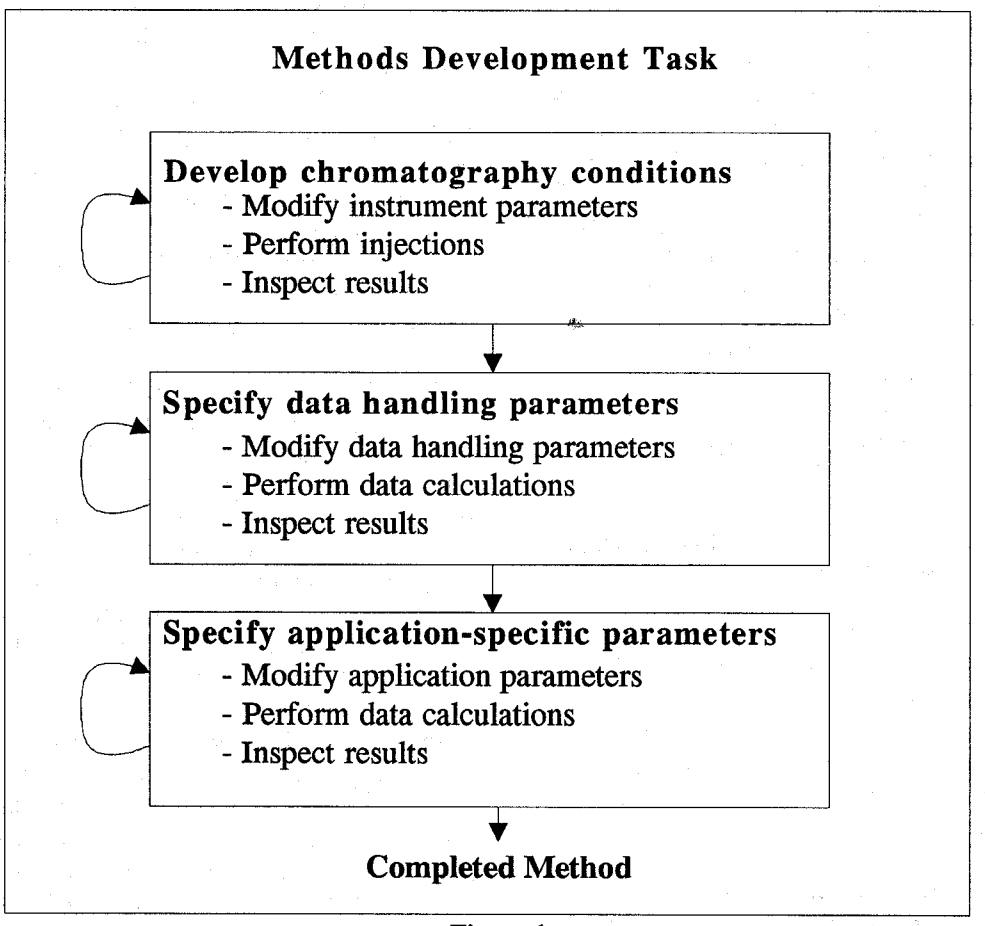

Figure 1

evaluation, but we didn't want the process of documentation to impede the evaluation.

Due to time and resource constraints placed on the project, empirical usability testing on interface prototypes was not considered to be a feasible evaluation technique for the situation. Co-operative Evaluation had been practiced in-house to gain some insight on what was required to use this procedure successfully [2, 6]. Based on our experiences, we felt that in order to do meaningful evaluation using this procedure, a prototype modeling the system behavior in a rather convincing manner was needed. We also believed that we needed ready access to subjects who were familiar with the application and who adequately represented our customer profile. We had attempted to use in-house customer representatives, but found that the bias they had developed while working intimately with our products influenced the way in which they performed user tasks. Time and resource constraints prevented us from developing a working prototype and doing the field visits that would be necessary to satisfy our criteria.

We reviewed several papers suggesting alternative interface evaluation procedures, including Lewis and Polson's Cognitive Walkthrough procedure [1, 4, 5]. The Cognitive Walkthrough approach appealed to many of the interface evaluators with a background in software engineering who have had good experiences with software design walkthroughs. It provided a satisfactory theory-based mechanism for finding problem areas in the user interface, and because of the familiar nature of the procedure, it seemed to be a good technique to use in an organization that had relatively little prior experience with formal user interface evaluation.

Task Selection

The user interface evaluation team spent time identifying core tasks of our customers, based primarily on experience in the field and knowledge of the market. Two major modes of usage that exist with our product are high-throughput automated analysis and methods development. Automated analysis requires little participation of the user beyond setting up the initial conditions for the sequence of separations to be performed. Methods development is a far more interactive task, requiring constant attention of the user. The burden on the user interface is greater for this task since the "modify-perform-inspect" loop needs to be optimized to the users' work flow. The chromatographer does not wish to be burdened by a system whose user interface gets in the way while performing this task.

The user interface evaluation team selected methods development for the new application as the task to be evaluated. This selection was made because of the interactive nature of the interface, because the methods development task is considered to be a core task of our chromatography workstation, and because it was the portion of the new application interface that had the most unresolved issues. Since the new application was still early in the design stage, the opportunity of "doing it right the first time" was there.

Before the evaluation session began, it was recognized that the entire methods development task was too long to be covered in one sitting. Therefore, only a subset of the task, the specification and verification of the application-specific parameters, was selected for the first evaluation. We hoped (perhaps naively) to cover a substantial portion of this task subset in a single ninety minute session.

The Walkthrough

The ground rules for the walkthrough were derived not only from the Lewis and Polson material, but also from internally developed walkthrough procedures used on software architecture designs [7]. In this procedure, the evaluation participants play one (or two) of four roles: Presenter, Evaluator, Moderator, and Recorder. Moderators and Recorders may act also as Evaluators.

The participants arranged themselves in a meeting room before an overhead projector and a large computer monitor. Drawings of the proposed screens were displayed on the overhead projector while the monitor displayed existing software which could be used to demonstrate the system's proposed behavior. The Presenter described to all participants the task to be evaluated, and the preferred path to the goal state. This was done in a session preceding the actual evaluation session. During the walkthrough, the Presenter traversed the preferred path by executing all the actions required to navigate through it. The Presenter identified alternative paths, and the consequences of pursuing them. Any uncertainties in the proposed design were noted. The Moderator ran the meeting, keeping everyone focused on the walkthrough, and resolving any issues that arose regarding the process. The Recorder wrote down each step taken during the walkthrough on an evaluation sheet. We used a modified version of the evaluation sheet presented in the Lewis/Polson paper (see figure 2). In order to help ease the burden of recording all the evaluators' comments, the Recorder used an electronic template of the evaluation sheet that reduced the amount of repetitive typing required. The Evaluators answered each of the questions on the evaluation sheet for each step taken towards the goal state. Actions and choices were ranked according to the percentage of potential users that were expected to have problems. As likely actions were

identified, the Presenter executed the action and the system response was recorded.

After the first immediate goal was presented, the group proceeded to identify the first atomic action that was likely to be taken by most users. The action was executed and the system response recorded. The evaluation proceeded along these lines and several problem areas in the user interface were identified. Suggestions for alternate design approaches were tabled by the Moderator so that the evaluation could continue within the time limits that had been defined. While the Recorder was able to transcribe comments during the evaluation by typing them into a template, the evaluation team spent much of the time waiting for the Recorder to catch up with the action.

At the end of the ninety minute period, only ten atomic actions were covered. This is roughly equivalent to opening a single file and displaying a windowful of application parameters to edit (a minor portion of the task subset that we wanted to evaluate). While the participants were satisfied with the quality of the evaluation, they were disappointed with the small amount of material covered and the pace with which it proceeded. The thought of repeating the process several more times in order to complete the evaluation for the task subset was discouraging to all of the participants.

The Jogthrough

To address the problems associated with the pace of the walkthrough, we decided to modify the procedure for future evaluation sessions. Instead of transcribing the comments manually into an electronic template, we recorded the evaluation session on videotape. We had already developed a test logging software package in-house that is used during videotaped usability testing sessions. We were able to make use of this test logging software during the evaluation in a similar manner -- to log significant events in real time. The role of the Recorder, therefore, becomes one of camera operator and event logger. The test logging software allows the Recorder to quickly enter notes about the events of the evaluation and stamp them with a time value that is synchronized with the video camera's timer. The test log

Walkthrough Evaluation Sheet

Actions/choices should be ranked according to what percentage of potential users are expected to have problems: 0 = none; 1 = some; 2 = more than half; 3 = most.

1. Description of user's immediate *goal*:

2. (First/next) atomic *action* user should take:
   - 2a. Obvious that action is *available*? Why/Why not?
   - 2b. Obvious that action is *appropriate to goal*? Why/Why not?

3. All other available actions *less appropriate*? For each, why/why not?

4. How will the user *execute* the action?
   - 4a. Problems? Why/Why not?

5. Execute the action. Describe system response:
   - 5a. Obvious *progress* has been made toward goal? Why/Why not?
   - 5b. User can access needed *information in system response*? Why/Why not?

6. Describe appropriate *modified goal*, if any:
   - 6a. Obvious that *goal should change*? Why/Why not?
   - 6b. If task *completed*, is it obvious? Why/Why not?

Figure 2

is then later used to index into the video tape and review segments of interest.

We noted that during the walkthrough, several good design suggestions were raised, but they were not well accommodated by its structure. The walkthrough, after all, is an interface evaluation procedure and does not lend itself to the design process. For the most part, discussions along these lines were suppressed by the Moderator. In the modified procedure, we wanted to encourage these suggestions, as long as they were appropriate, and explore the alternate paths the user might choose as a result of their implementation. To accomplish this, we changed the ground rules of the evaluation to allow more free discussion and to allow the discussion to stray from the questions on the evaluation sheet when we thought it might provide fruitful suggestions. Questions like "what if it were implemented this way" would spawn discussions about the possibilities of various other approaches to the design problem.

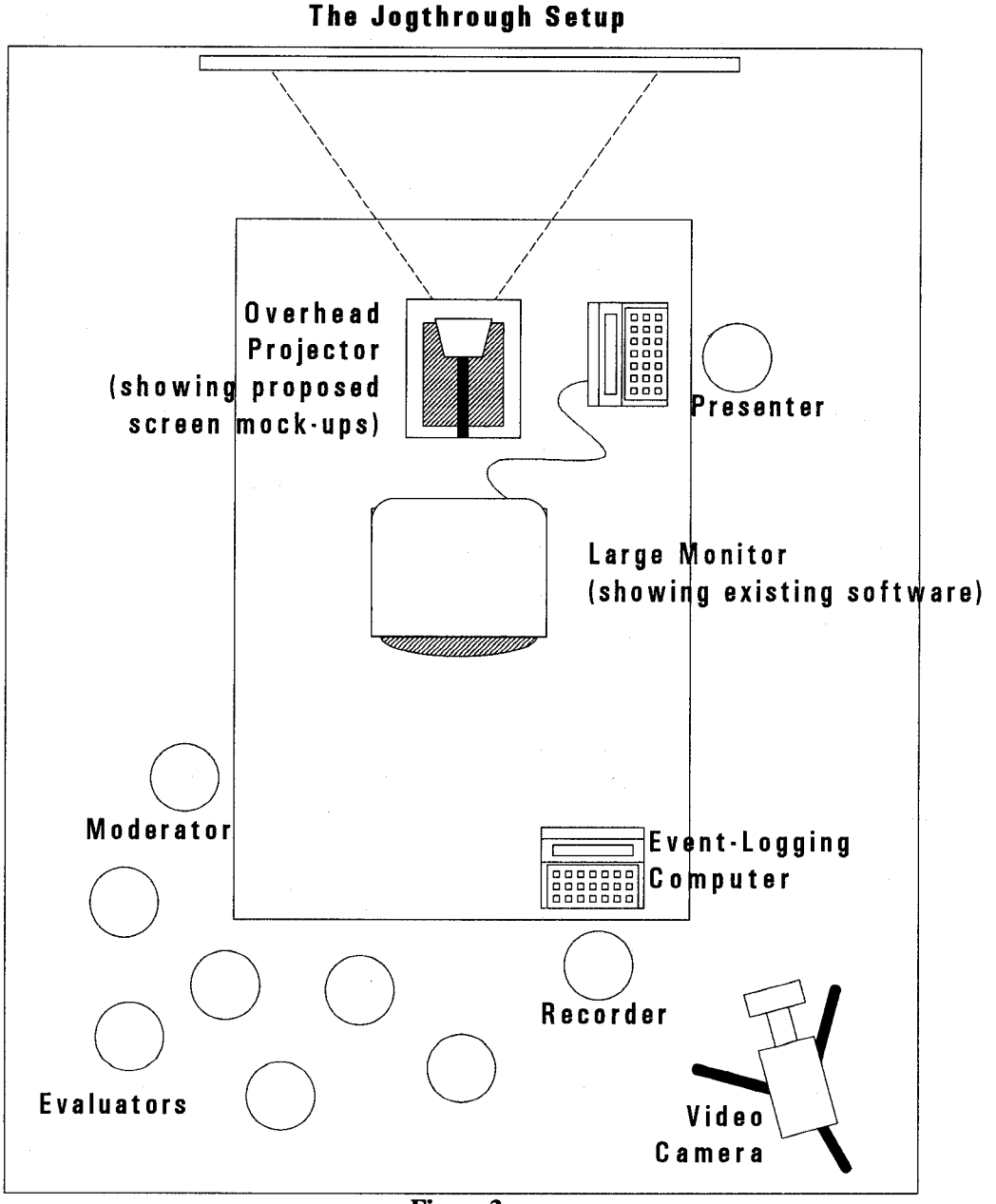

Figure 3

The criterion by which the Moderator judged discussions inappropriate for the scope of the evaluation became more subjective with this procedure, since we were no longer bound to adhere strictly to the questions on the evaluation sheet. It was up to both the Moderator and the Presenter

to decide when it was appropriate to combine several atomic actions into a single step, and when discussion of a particular design issue was beyond the scope of the evaluation. In some cases it was agreed that certain concerns about the interface could not be resolved in the evaluation session and would require further empirical study.

In the ninety minute jogthrough, roughly thirty user actions were covered. Atomic actions that would have been covered individually in the walkthrough procedure were combined when the evaluators felt that individual examination of the actions would not provide any useful information. Certain areas of the proposed design that had not been fully developed were discussed and various alternative design suggestions were made. The focus of this procedure, as with the Cognitive Walkthrough, is to evaluate how the user might choose from the available options the one considered most likely to result in a state closer to the goal state. When design suggestions were made, they were immediately evaluated with this in mind.

RESULTS

Since the granularity of the atomic actions evaluated in the walkthrough sometimes differed from those evaluated in the jogthrough, it is difficult to make quantitative comparisons of the number of steps evaluated. A rough estimate can be made that approximately three times as much material was covered in the jogthrough than in the walkthrough. The section of the task that we intended to review, however, was not fully covered in either session.

The interaction provided by the jogthrough brought about some modified behavior of the participants, especially the application experts and in-house users who were not familiar with formal review techniques. Since a certain amount of discussion was encouraged, they became more vocal and played a bigger role in the evaluation than they had during the walkthrough. When application-specific terminology or conventions were in question, they were able to immediately share their insights. This increased level of participation was a welcomed benefit of the jogthrough technique.

The documentation of the walkthrough session was definitely the pacing factor during the evaluation. The Recorder was forced to omit certain comments simply to keep up with the flow of the discussion. This resulted in frustration during the session as the participants constantly had to be told to slow down, and frustration later on when the interface designers had to decipher the cryptic abbreviations that the Recorder resorted to under pressure. The video tape generated during the jogthrough, on the other hand, captured all the comments and discussions made by the participants and required little maintenance during the session. The Recorder was able to log the evaluation of major steps throughout the task without affecting the tempo of the discussion.

The evaluators responded favorably to the modifications made to the walkthrough procedure resulting in the jogthrough. There was a general feeling of accomplishment after the jogthrough that was in marked contrast to the feeling of frustration after the walkthrough. Although we did not cover all that we had anticipated with either technique, there was more willingness to continue the jogthrough than there was with the walkthrough.

CONCLUSIONS

The modified walkthrough, the jogthrough, satisfied our evaluation objectives and provided design inputs that were not afforded by the structured walkthrough procedure. The jogthrough procedure seems particularly well suited to a product development environment where multi-disciplinary teams evaluate proposed interfaces, but have primary responsibilities other than user interface design and evaluation. The informal, fast-paced feel of the evaluation encourages participants to make design suggestions, but provides a theory-based framework that can be used to immediately screen them. The active participation of people with different areas of expertise has the additional benefit of increasing each person's understanding of the overall project, and in improving communication and consensus-building across product development teams.

The structure of the walkthrough enforces a rigorous review of all steps necessary to navigate through the proposed system, and therefore makes it difficult to overlook hidden usability errors (although the tedium of the procedure itself may lead to some oversights). This may not be the case with the jogthrough if the evaluation team chooses to skip over a step that is considered insignificant. A greater burden is placed on the Moderator and Presenter during the jogthrough since the criteria used to determine when it is appropriate to discuss design alternatives are left up to them.

Although it was our experience that the jogthrough allowed us to evaluate significantly more of the proposed interface than the walkthrough, it is still a time consuming activity. In fact, it is probably not feasible to evaluate all core tasks in our system in this manner. Therefore, task selection becomes a very important issue, and sometimes decisions have to be made to neglect areas that may present usability problems in the product. These are never easy decisions to make, but the jogthrough procedure provides a mechanism for identifying aspects of the interface that require further attention, and allows

evaluators to find these areas of concern faster than with the walkthrough.

By allowing design alternatives to be proposed and discussed during the evaluation session, the jogthrough procedure becomes as much a technique for collaborative design as it is a structured evaluation. By combining these two processes, a more formal approach to the user interface development process was employed than may have been otherwise, given the time constraints on the project. The tight coupling of design and evaluation promotes the rapid development of working breadboards that can be further evaluated using empirical techniques. Problem areas identified in the jogthrough can be focused upon in usability tests when the time and availability of subjects is often limited.

SUMMARY

Under strict schedule constraints, a proposed user interface was evaluated using several techniques. A Cognitive Walkthrough approach was used as an alternative to empirical usability testing due to both schedule and resource constraints. The Cognitive Walkthrough proved laborious and did not make the best use of the evaluators' limited time. A modified walkthrough procedure, a "Cognitive Jogthrough", was developed to gain valuable usability feedback from evaluators in a fast-paced session. The results of the two approaches showed that much the same type of feedback was made available by the jogthrough as with the walkthrough, but far more actions were evaluated and design suggestions were made that were not well accommodated by the rigid structure of the walkthrough procedure.

The Cognitive Jogthrough technique took advantage of the diverse background of our evaluation team members. Application experts and in-house users were given the opportunity to make design suggestions and voice their opinions on the usability of the proposed interface. Design discussions allowed us to resolve some concerns within the structure of the evaluation session, and allowed us to target others as areas for further investigation. We feel that we retained enough structure to provide a thorough review of the interface, but enough flexibility to change the focus of our attention during the session when we felt it appropriate. By using a video camera in conjunction with test logging software, we were able to fully document the session without hampering the pace of the evaluation. The test log provides a convenient way to quickly access information on the tape at a later time. These aspects of the jogthrough process helped fulfill our goal of employing a user interface evaluation methodology that worked within the framework of our development environment.

ACKNOWLEDGEMENTS

The authors wish to thank Lisa Breslow, Gary Burce, Duncan Carmichael, Jean-Louis Excoffier, Susan Finkelman, Tim Medlin, Bob Palin, Sue Ann Scheppers, and John Sullivan for their participation in the user interface evaluation process. Additional thanks to Susan Finkelman, Marge Levin and Tim Medlin for their valued editing and review inputs.

REFERENCES

1. Bell, B., Rieman, J., and Lewis, C. Usability Testing of a Graphical Programming System: Things We Missed in a Programming Walkthrough. In Proceedings of CHI 1991 Conference (New Orleans, LA, Apr. 27 - May 2). ACM, New York, 1991, pp. 7-12.

2. Haber, J. Proposing Usability Testing to Management - An "It works therefore it's truth" approach. Presented at CHI 1991 Conference (New Orleans, LA, Apr. 27 - May 2) by Lora Davenport.

3. Jeffries, R., Miller, J.R., Wharton, C., and Uyeda, K. User Interface Evaluation in the Real World: A Comparison of Four Techniques.. In Proceedings of CHI 1991 Conference (New Orleans, LA, Apr. 27 - May 2). ACM, New York, 1991, pp. 119-124.

4. Lewis, C., Polson, P., Wharton, C., and Rieman, J. Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. In Proceedings of CHI 1990 Conference (Seattle, WA, Apr. 1-5). ACM, New York, 1990, pp. 235-242.

5. Nielsen, J., Molich, R. Heuristic Evaluation of User Interfaces. In Proceedings of CHI 1990 Conference (Seattle, WA, Apr. 1-5). ACM, New York, 1990, pp. 249-256.

6. Wright, P. and Monk, A. Co-Operative Evaluation - The York Manual. University of York, York , UK, 1991.

7. Yourdon, E. Structured Walkthroughs. Yourdon Press, New York, 1978.