Testing Documentation with “Low-Tech” Simulation

David G. Novick
European Institute of Cognitive Sciences and Engineering
4 Avenue Edouard Belin
31400 Toulouse, France
+33 (0) 5 62 17 38 38
novick@onecert.fr

ABSTRACT*

This paper introduces low-tech simulation as a technique for testing procedures and their documentation. The key idea is to test the interface-procedure-documentation set in the early stages of development, where changes can be still be made easily, using extremely low-cost simulation materials. Using this low-tech method, developers can test and improve the usability of documentation for user interfaces that are still being designed. An extended example, involving a new aircraft cockpit interface for text-based air-traffic control communication, presents a look at the low-tech approach in use. An evaluation of low-tech simulation shows that the approach was practical. Qualitative analysis indicates that low-tech evaluation produces results that are different from those obtained using the cognitive walkthrough for operating procedures and similar to those obtained using traditional high-tech simulation.

Keywords

Usability testing, simulation, user-centered design, operating procedures.

1. INTRODUCTION

How can you test the documentation before the underlying system is built? This paper reports the results of a feasibility test of using a "low-tech" approach to usability testing of documentation. The key idea is to test the combination of the interface, the operating procedures and the documentation in the early stages of development, where changes can be still be made easily, using extremely low-cost simulation materials. Using this low-tech method, developers can test and improve the usability of documentation for user interfaces that are still being designed.

This paper will introduce the application of low-tech techniques to the testing of operating procedures and their documentation, present an example of a low-tech simulation of a new aircraft cockpit interface for text-based air-traffic control communication, describe an experiment to validate the low-tech approach, and compare the results of the experiment with those from other techniques for testing documentation. The paper will conclude with a discussion on ways of improving the low-tech simulation and its use.

1.1 Early Testing of Documentation

The need for an evaluation method for documentation that takes the operation user interface into account but which does not require a highly developed, exhaustively defined specification or prototype arose out of my experience in developing and using the cognitive walkthrough for operating procedures (CW-OP) (Novick, 1999). Like low-tech simulation, the CW-OP was aimed at providing usability testing of documentation, especially for procedures, much earlier and at lower cost than the use of full-scale simulators. The CW-OP involved step-by-step cognitive analysis of possible problems in executing operating procedures based on their documentation.

While the CW-OP proved largely effective in providing lower-cost testing, it also needed an explicit “gold-standard” definition of the underlying hardware/software interface. In determining the correct action during evaluation of steps in the computer interface, the evaluators required exhaustive information on the interface's functionality. However, requiring a precisely defined interface makes it more difficult to develop operating procedures when the hardware and software for the interface have not yet been specified in complete detail. This tends to limit the method’s usefulness during early development phases, particularly if the developers wish to use a documentation-first approach.

What was needed, then, was a way of testing operating procedures and their documentation with an interface at an earlier stage of development, where the interface could easily be changed and, in order to obtain the fullest benefits of user-centered design, this quality of being easily changed is apparent to the user-testers. Low-tech simulation is a way of addressing this need.

1.2 Low-Tech Techniques

The direct inspiration for the low-tech approach came from PICTIVE (Muller, 1992), which is a technique for encouraging users to participate in design. In PICTIVE, the developers provide users with easily manipulable, low-tech materials such colored paper, pens, scissors, removable self-stick notes (such as Post-It™ notes), and transparent tape. The users create designs and scenarios using these materials, which tend to encourage thinking of the prototype user interfaces they create as inexpensive and plastic, thus in turn encouraging freedom to change the prototype.

Like other paper and cardboard prototyping methods (see, e.g., (Kyng, 1988; Wulff, Evenson & Reinfrank, 1990; Beyer & Holtzblatt, 1998), PICTIVE focused on the early design stages of user interface development. As adapted here for testing operating procedures and their documentation, the approach is aimed less at initial co-design but rather at a kind of usability testing of documentation in the early stages of interface development. Indeed, Beyer and Holtzblatt specifically distinguished their design approach from usability testing, which they viewed as a late-stage tune-up at the end of the design process. But for operating procedures and documentation, as opposed to the underlying user interface, some form of actual testing with users as early as possible might (1) lead to more usable documentation and (2) detect difficulties that are serious enough to warrant timely redesign of the user interface.

* Author’s current address: Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968-0518, USA, novick@cs.utep.edu

The low-tech simulation technique involves preparing rough paper or cardboard mock-ups of the basic interface background, along with sticky-note mock-ups of the interface content. This enables the tester to act as both the underlying system and the interface management system by swapping sticky notes into the interface, reflecting changes in the interface induced by external events or by users' actions. The characteristics of the "interface" can, if necessary, be adapted on the fly by the tester. In a sense, low-tech simulation is the opposite of the familiar Wizard-of-Oz approach (Dahlbäck., Jönsson & Ahrenberg, 1993), where the user is falsely led to believe that a computer is producing responses. Rather, in low-tech simulation the idea is to avoid giving the user the impression that the system actually works, encouraging—as in other paper and cardboard prototyping methods—users to believe that their design inputs can be taken seriously.

2. USING LOW-TECH SIMULATION

I now turn to an extended example of using low-tech simulation for testing documentation. The example involves the domain of operation of commercial aircraft and, in particular, a new text-based interface for air-traffic control (ATC) communications developed by Aerospatiale Matra Airbus. The new ATC interface, called DataLink, is slated to replace the current voice-based system, particularly for the "cruise" flight phase over oceans.

An aircraft's procedures and other documentation must meet standards of speed-of-operation, continuity-of-function, and safety. In all commercial aircraft, regardless of manufacturer, operation of the aircraft is specified in terms of procedures, which cover everything from control of the aircraft's flight surfaces and engines, to safety inspections, to air-traffic control communications and navigation. For complex systems with interfaces that give a large degree of flexibility to experienced users, procedures provide paths through possible uses of the system that are known to be safe and effective. They also serve to standardize activities across crew members, so that crew can be teamed without regard to whether they know one another. As a result, aircraft builders need means of developing and evaluating procedures that account not only for human-machine interaction but also for human-human-machine interaction.

The specific example presented here represents an intermediate state in the design process for the both the interface and its documentation. Accordingly, many aspects of the example differ significantly from the current design and documentation, and the example should not be taken as reflecting Aerospatiale or Airbus's design, documentation, standards or systems.

2.1 Aircraft Operating Manuals

The principal user documentation for an aircraft's pilots is the aircraft's flight crew operating manual (FCOM), which typically contains sets of:

_ Parameters, such as charts for take-off and landing speeds as a function of weight,

_ Explanations of the aircraft's systems and controls, such as the hydraulic system and the interface to the navigation computers,

_ Procedures, such as for normal, abnormal and emergency conditions, and

_ Checklists.

The documentation tested in this low-tech example was a prototype FCOM chapter for the DataLink interface. It included a descriptive section explaining the functions of the interface, and a prescriptive section containing the operating procedures associated with the use of the interface. Figure 1 presents one of the operating procedures contained in the prototype FCOM chapter.

Procedure: <u>Respond to a request to report</u>

- If you **need a short-term delay** to respond to the clearance, respond STBY.
- Respond ROGER.
- When the **report condition occurs**,
  - If the **report displayed by the DCDU is incorrect or incomplete**, MODIFY the report,
  - SEND the report.

Example

Aircraft receives message "REPORT PASSING ADSIN."
Aircraft sends message "ROGER".
When passing ADSIN, aircraft sends message "PASSING ADSIN."

Figure 1. Prototype DataLink operating Procedure "Respond to a request to report"

The example procedure in Figure 1 shows several conventions that were used consistently in the prototype documentation. These conventions included marking the title of the procedure with underlined text, marking conditions with bold text, marking actions that corresponded to labels in the interface with uppercase text, and providing an example for each procedure.

2.2 Simulated DataLink Interface

In this example, as in the experimental session validating the "low-tech" simulation, the main interface to be simulated is the DataLink Control and Display Unit (DCDU) in an Airbus A340 jumbo jet. The DCDU was represented by a green pasteboard sheet, approximately 18x24 cm, with the interface's screen and keys drawn with marker and pen. The interface's content elements were represented by whole or partial green sticky notes with hand-written labels. The area of the low-tech mock-up is about four times that of the actual DCDU.

Figure 2 shows the low-tech simulation of the DCDU in its state just before the aircraft receives a communication that should trigger the "Respond to a request to report" operating procedure. There are no messages, and the interface shows that the aircraft is monitoring the ATC center called KZAK.

The functions of the buttons were explained in the descriptive section of the prototype FCOM chapter. If there were multiple messages, these could be navigated using the MSG+ and MSG- buttons. If a message was too long to fit the screen, the different pages of the message could be displayed using the PGE+ and PGE- buttons. The DIM and BRT buttons controlled the brightness of the display. And the PRINT button would produce a hard copy of the current message. Finally, the four buttons with the "—" label are called line selection keys. They are familiar to Airbus pilots because similar keys are used in a standard cockpit interface. When pressed, the key activates a function displayed next to the key. Use of the line selection keys will be present in the example in Section 2.3.

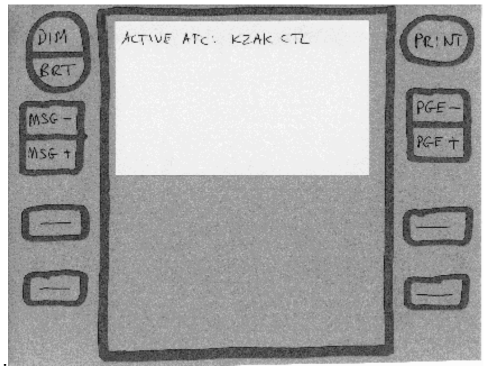

Figure 2. DCDU with no messages

In the figure, the sticky note appears much lighter than it did in the actual mock-up, where the green color of the sticky notes was only slightly lighter than the green color of the pasteboard background. The exaggerated color difference in the figure, however, makes it easier to discern the placement of the sticky notes.

Sets of prepared sticky notes representing task scenarios are arranged in folders. Figure 3 shows a representative page of one of the folders. As the hand-operated simulation progresses, sticky notes from the folders are placed on and removed from the interface background.

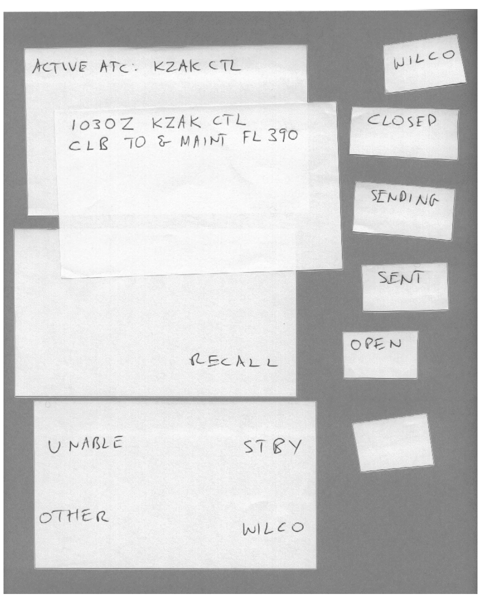

Figure 3. Folder of content elements for a task scenario

2.3 Steps in the Simulation

I turn now to the particular steps of simulating the operating procedure "Respond to a request to report." As presented in Figure 1, the crew's task, when receiving a request-to-report message from ATC, is to acknowledge the message and then, later, send another response when the aircraft reaches a specified navigation point. It should be recalled that this example presents an intermediate design of the DCDU and does not reflect the actual function or interface for the DataLink system.

The simulation begins with interface as presented in Figure 2: there is an active connection to an ATC center but no current message.

When a request-to-report message arrives from the ATC center, it is displayed in the DCDU, as shown in Figure 4. The DCDU displays the time of the message, the ATC center that sent it, and the text of the message. In this example, the text of the message is "REPORT PASSING NMEA." NMEA is understood as the identifier of a navigation point. The DCDU also displays the word "OPEN," indicating that the message has not yet been resolved. And in the bottom part of the display, the DCDU presents three actions that the crew can select with the line-selection keys. Changing the simulator interface from Figure 2 to Figure 4 simply involves removing one sticky note and placing three new sticky notes, which were ready in the folder associated with the scenario for this task.

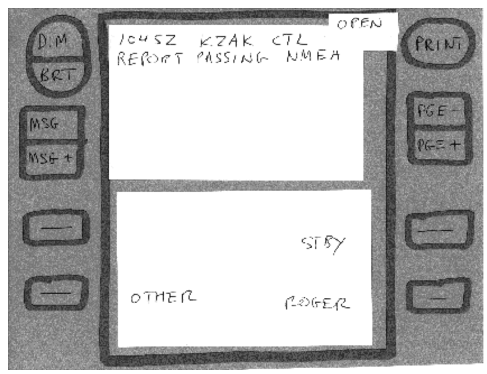

Figure 4. DCDU with an open message

The next step in the example depends on the crew's choice of action. Assuming that the crew had a normal situation and followed the operating procedure, they would select the key associated with the label "ROGER." To do this in the low-tech simulation, the user would touch the appropriate key drawn on the side of the interface.

The example now jumps ahead to four minutes later in the in scenario. The aircraft has reached the waypoint NMEA, and the crew have just taken the action of sending an appropriate message. This state of the interface is depicted in Figure 5. The top of the DCDU displays the time the crew's message was sent to the ATC center, the name of the center, and the content of the message. The bottom of the DCDU displays the actions that may be available to the crew. And in the center of the display, the status of the response communication is shown as "SENDING." All of these elements were placed on the low-tech interface from the sticky-note folder for this task scenario.

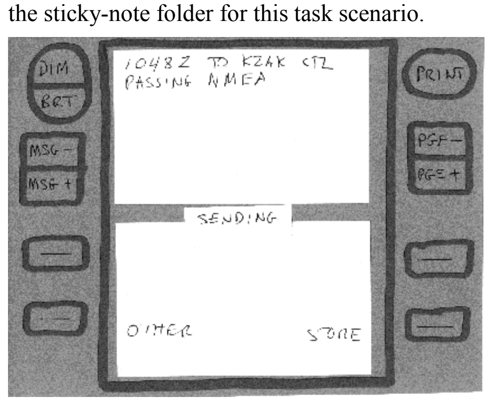

Figure 5. DCDU showing message being sent

The next step in the simulation is a relatively simple one. As shown in Figure 6, the DCDU changes the status of the aircraft's communication from “SENDING” to “SENT.” In the low-tech simulation, this step consists of substituting one small sticky note for another.

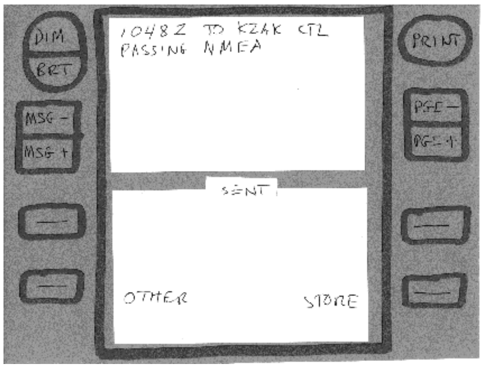

Figure 6. DCDU showing message sent

With their message now sent, the crew can select one of the actions associated with the line-selection keys in the lower part of the DCDU display. The normal action at this point would be to store the message, and so the user should again touch the lower right-hand key.

The result of the crew's action is shown in Figure 7. The DCDU again shows that the aircraft is monitoring the KZAK ATC center; in fact, this is the same sticky note as used in Figure 2. The DCDU also displays one available function that the crew could select: the RECALL function would enable them to display the last message stored. As in the prior steps, the simulation of this part of the interface involved a simple substitution of sticky notes from the task scenario folder. While the labels associated with the line-select keys could have been represented by individual, small sticky notes, they were normally presented together on a single large sticky note in order to keep the changes of display content relatively efficient. So in this case, even though there's only one available action, a large sticky note with the single action was used so that the display would retain a consistent appearance.

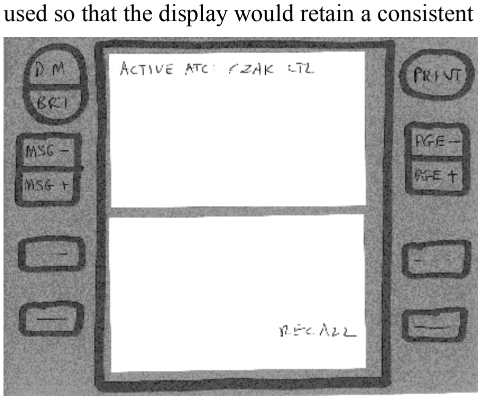

Figure 7. DCDU with no open message

In all, the complete scenario for this operating procedure requires eight large sticky notes and six small sticky notes.

3. EVALUATION OF FEASIBILITY

The feasibility of the low-tech simulation was tested using the prototype DataLink documentation I developed with the COHERE project’s documentation authoring tool (Novick & Juillet, 1998). The results of the simulation can be compared to evaluation of the same documentation (and especially the operating procedures) using CW-OP. The results can also be compared with those for a similar set of procedures evaluated in a full-scale A340 simulator.

The feasibility test sought to answer three questions:

1. Was the low-tech approach practical? That is, could the background/sticky-note system reasonably represent the interface and its responses to events and interactions?
2. How would the results of the low-tech approach compare to those from walkthroughs and full-scale "high-tech" simulation experiments?
3. How could the low-tech approach be improved?

Although the main focus of the feasibility evaluation was on operating procedures, the test did include both descriptive as well as prescriptive parts of the interface's user documentation. This was the case not only for the test of the low-tech approach but also for the CW-OP and the full simulator experiment.

The feasibility evaluation procedure consisted of three main phases: training, testing and debriefing. In the training phase, the subject, a former senior commercial test pilot and instructor, was asked to read the prototype DataLink documentation. The documentation was supplemented with responses by the experimenter to the subject's questions. The examples contained in the documentation were used as training materials for run-throughs of the procedures. The subject was advised that he could refer to the documentation during the evaluation.

In addition to the DCDU, the overall cockpit interface was represented by a photocopy of an existing FCOM page depicting the crew interface with the flight management and guidance system. Small blue sticky-note fields labeled “ATC” were added at the left and right sides of the printed instrument panel to represent push buttons.

Task scenario folders of sticky notes were prepared for two tasks. One task involved the operating procedure “Respond to a clearance;” the folder contained six large and five small sticky notes, most of which are shown in Figure 3. The second task involved the operating procedure “Respond to a request to report;” the corresponding folder contained eight large and six small sticky notes as described in the example in Section 2.3. Additionally, in both tasks the user followed a procedure-like “technique” called “Respond,” for which no additional sticky notes were required.

During the feasibility evaluation, I acted as the computer, changing the appearance of the interface depending on what the system was supposed to do and on what actions user took. The user read the documentation, and referred to it during the tasks. The user's actions led to situations that were all covered by the prepared content materials. If needed in response to possible excursions by the user from the normal task scenario, new sticky-note elements could have been created as appropriate.

The user produced unexpected behaviors for two steps of the “Respond to a clearance” procedure. First, the user thought that he would have to take a separate “SEND” action to send a “WILCO”" message, when actually selecting WILCO sent the message directly. Second, the user selected “STORE” when the system would have performed this action automatically. The user also produced two unexpected behaviors for the “Respond to a request to report” operating procedure. Both cases involved adding extra items to oral announcements made to his (presumed) co-crew member.

4. RESULTS

I now present the results of the feasibility evaluation, tracking the study's three main issues: the underlying practicality of the low-tech approach, comparison of the low-tech approach with alternative approaches, and possible improvements in the low-tech approach.

4.1 Feasibility

The low-tech simulation proved to be quite practical. The content folders for the two tasks were organized temporally within groups of similar items (see, e.g., Figure 2), which enabled quick and easy replacement of appropriate sticky notes on the interface background. That is, the background/sticky-note system reasonably represented the interface and its responses to events and interactions.

The experience showed that the simulation could, produce (1) the desired operating-procedure-following behaviors generally, and (2) elicit unexpected behaviors that shed light on ways to improve the procedures and their documentation.

As noted by the subject, the low-tech approach would not work well for determining the timing of operating procedures, but that the pauses in the interface's "response" due to the experimenter's having to work with the sticky notes corresponded well with his need to think things through; in other words, if the point of the test is the logic and sequencing of the procedures, then the method works well. However, the subject would have liked to have an additional background board and sticky-note content for other related interfaces. In particular, he would have liked to have tried out the interface and operating procedures for modifying or creating messages. After the evaluation, I checked on the feasibility of creating such an additional interface, which was significantly more complex than the DCDU. I was able to create the new interface background in about 10 minutes, and the creation of appropriate content took an additional 15 minutes.

During the evaluation, the subject suggested several changes to the interface and the operating procedures. This indicates that the approach did, in fact, present an interface with a sufficiently plastic quality so that users would not view the interface as a "done deal" that they could not easily affect.

4.2 Comparisons with Other Methods

The results of the "low-tech" approach were compared with those of the cognitive-walkthrough and "high-tech" simulation evaluations of the same or similar documentation.

4.2.1 CW-OP

The cognitive walkthrough for operating procedures (Novick, 1999) is a usability inspection method for interfaces that has been adapted to the evaluation of operating procedures and their documentation. The method leads the designer to consider factors such as users' backgrounds and mental effort. In brief, a task is decomposed into steps in the interface, which are analyzed individually with respect to connections between goals, artifacts, actions, and results. The CW-OP deals with procedural rather than interface steps. The evaluation process draws the evaluator's attention to the presentation of the procedure in the documentation, asks the evaluator to determine explicitly if training or experience are required for a particular step, looks at whether the procedure correctly implements the intended function with respect to the overall system, and determines whether errors are probable and, if so, serious. Each step is described as a "success story" or a "failure story," depending on the outcome of the analysis.

The low-tech simulation approach provided qualitatively different results than the walkthrough. For example, the subject did not have difficulty with an operating procedure that most of the CW-OP analysts identified as problematic. At the same time, the subject did produce unexpected behaviors in both tasks, where none of these effects had been predicted by the CW-OP.

The CW-OP identified several more potential difficulties in the draft procedures than was evidenced by the user in the low-tech simulation. In particular, the CW-OP analysts identified 11 areas of possible difficulty in the operating procedures and made 19 suggestions for redesigning the operating procedures. They also identified 14 possible difficulties in the underlying interface, and made three design suggestions. In contrast, the low-tech simulation identified a total of four potential difficulties in the draft operating procedures, and made six suggestions on improving the underlying interface. It should be born in mind, though, that the CW-OP involved six analysts, so the disparity in the number of issues identified and suggestions made does not appear to be unreasonable.

The CW-OP did require a precisely defined interface to test the documentation, while the low-tech approach showed that it could work well for testing documentation even when the interface is not entirely defined.

4.2.2 High-tech simulation

A second comparison involves the results of the low-tech approach with those from a full-fledged test on a high-tech simulator. The low-tech simulation was carried out with a single pilot, while the high-tech sessions required use of teams of pilots. In this case, six pairs of instructor pilots flew simulated flights of about two hours in Aerospatiale's A340 research simulator. They used a later version of the draft documentation that had been adapted on the basis of the results from the CW-OP evaluation.

Given the disparities between the low- and high-simulations in terms of time and number of subjects, and in terms of the number of operating procedures used, quantitative analyses would not be meaningful. In qualitative terms, though, the high-tech simulation approach provided results substantially similar to that from the low-tech simulation. In particular, the high-tech crews also produced unexpected behaviors with respect to (1) storing (or not storing) messages and (2) oral announcements to the other crew member. Just as in the low-tech simulation, the high-tech crews crew sometimes skipped required announces or added extra announces.

This relatively similar behavior with respect to announces is interesting because announces are among the "least-afforded" parts of the interaction, in the sense that they often do not have clearly labeled, selectable actions available in the interface. Rather, announces are something that depends almost entirely on the operating procedure rather than the interface for their triggering. This means that announces are a key indicator for distinguishing the effects of the manual from the effects of the interface. Consequently, it is particularly meaningful for evaluation of documentation that both the low- and high-tech simulations produced roughly similar results.

4.3 Possible Improvements

The low-tech approach could be improved in a couple of ways, mainly bearing on the appearance and management of the content elements.

As anticipated by the experimenter and noted by the subject, better use could be made of color. In the actual DCDU interface design, parts of messages appeared in different colors, where the color had standard meanings for pilots about things such as items that needed attention. Occasionally, a word or phrase would change color as a result of a system or ATC action. The evaluated documentation described these color-related aspects. Thus for more faithfully realistic interaction, perhaps the sticky notes could be in different colors, representing the colors of the underlying messages in the interface. Or the labels could be written in colored ink. In the latter case, though, it is not clear how to represent white text.

Repeated and extended tests should create a larger set of content elements, thus enabling more flexible "use" of the mock-up interface by the user. The main problem will be to organize the sticky notes; perhaps this could be done on the pages of a binder or notebook rather than in pasteboard folders. However, there was significant duplication among the content elements used in the two tasks of the feasibility test; this suggests that the growth in the number of sticky notes will be less than a linear function of the number of tasks.

5. DISCUSSION

While the low-tech approach to evaluating documentation proved feasible and useful, it has a number of inherent limitations that will be difficult or impossible to overcome.

Some of these limitations have to do with kind of interfaces associated with the documentation to be evaluated. For example, dynamic elements such as flashing push buttons or animated graphics are beyond the power of the pen-and-paper technology. A similar limitation is that management of screen area tends to be sloppy, as hand-written elements may not always fit exactly in the space available. However, this problem is unlikely to be serious unless information is actually hidden by another sticky note; the sloppy screen management probably simply reinforces the impression of informality and plasticity of the interface.

Other limitations involve the scope and role of the testing process. For example, testing the documentation in a more realistic setting—with both the pilot and co-pilot—would likely require more interfaces and hence more than one person to operate the simulation. For similar reasons, parts of the task may have to be elided; for example, the interface for creating new messages was not simulated in the feasibility study; the results of interaction with the message-creation interface were just introduced directly in the DCDU. Additionally, the low-tech approach may not be suitable for extended simulations. The low-tech feasibility simulations took about twenty minutes to complete, and were presented as unrelated tasks. The high-tech simulation experiments each lasted about two hours and involved a succession of naturally related tasks. Overall, then, low-tech simulation appears most appropriate for evaluating a prototype documentation in terms of its application to individual tasks.

It might be possible to use low-tech methods for an extended scenario, as was done in the high-tech experiment, but some instruments with more-or-less continuously changing displays would be difficult to simulate.

5.1 Eliciting Natural Expression

These differences between low-tech and high-tech simulation find their expression in the kinds of behaviors produced by users. The high-tech simulator's more realistic task environment, along with the presence of both a pilot and co-pilot, provided a basis for more naturalistic reactions. For example, in the high-tech simulation, the ATC center sent a message directing the aircraft to a navigation point, called MARAN, which was not in the aircraft's flight plan. This produced the following reaction from one crew member:

> It says here at maran, where the hell's this maran? Well we're not going to maran, so we'll just push unable for that.

It is hard to imagine the low-tech simulation having creating a level of involvement sufficient to produce this sort of reaction. Nevertheless, the overall information learned about the documentation appeared to have been relatively independent of the depth of the realism of the simulation.

5.2 Low-tech documentation?

Finally, I consider what would it mean to apply the low-tech simulation approach to the documentation itself. The low-tech method presented in this paper involved the interface used in the evaluation of the documentation, rather than for the documentation to be evaluated. Does documentation have an early-process equivalent of paper prototypes for interfaces?

This question is problematic because the effectiveness of documentation is so closely tied to presentation. In the case of operating procedures, one could argue that the operating procedure's steps are independently evaluable. But an operating procedure exists, outside of use, only in its realization in documentation. It is difficult to judge, *a priori*, the effects of presentation elements such as the conventions about underlined, bold and uppercase text presented in Figure 1.

In fact, much if not most documentation is on paper already, so it has both the tangibility of the low-tech interface and the fixed, non-plasticity of print. Ironically enough, an electronic version of a manual might be more effective in getting a user to appreciate the changeable nature of prototype documentation. This would lead to the curious combination of a paper prototype of the interface and an electronic prototype of its documentation.

5. CONCLUSION

This paper introduced low-tech simulation as a technique for testing operating procedures and their documentation. An extended example, involving a new aircraft cockpit interface for text-based air-traffic control communication, presented a practical look at the low-tech approach in use.

An evaluation of low-tech simulation showed that the approach was practical.. A comparative analysis indicated that low-tech evaluation it produces results that are qualitatively different from those obtained using the cognitive walkthrough for operating procedures and qualitatively similar to those obtained using traditional high-tech simulation.

The low-tech approach could be improved by increased use of color and by extending the simulation to more and longer examples. The approach also has some inherent limitations, particularly with respect to representing dynamic interfaces, simulating multiple interfaces simultaneously, and providing a level of user involvement that produces naturalistic, spontaneous behaviors.

6. ACKNOWLEDGMENTS

This research was supported by a contract from Aerospatiale Matra Airbus. Thanks to Jean Pinet for his participation in and contributions to the research. Thanks also to thank Florence Beaujard, Florence Buratto, Meriem Chater, Florence Reuzeau and Stéphane Sikorski for their work on the CW-OP and full simulator tests.

7. REFERENCES

Beyer, H., and K. Holtzblatt. *Contextual design: Defining customer-centered systems*. Morgan-Kaufman, San Francisco, CA, 1998.

Degani, A., and E. Wiener. Procedures in complex systems: The airline cockpit. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(3): 302-312, 1997

Dahlbäck, N., A. Jönsson and L. Ahrenberg. Wizard of Oz studies—why and how. *Knowledge-Based Systems,* 6(4): 258-266, 1993.

Kyng, M. Designing for a dollar a day. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 88)* (Portland, OR), 178-188, 1988.

Muller, M. Retrospective on a year of participatory design using the PICTIVE technique. *Proceedings of CHI 92* (Monterey, CA), ACM Press, 455-462, 1992.

Novick, D. Using the cognitive walkthrough for operating procedures. *Interactions*, 6(3):31-37, 1999.

Novick, D., and Juillet, J. (1998). Documentation integrity for safety-critical applications: The COHERE project, *Proceedings of SIGDOC 98*, Quebec, September, 1998, 51-57.

Wulff, W., S. Evenson and J. Reinfrank. Animating interfaces. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 90)* (Los Angeles, CA), 241-254, 1990.

David G. Novick earned his Ph.D. in Computer and Information Science at the University of Oregon in 1988 and his J.D. at Harvard University in 1977. He is professor and chair of Computer Science at the University of Texas at El Paso. His research focuses on interactive systems, especially development methods for interfaces and their documentation.