INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, *14*(3&4), 489–502

Remote Usability Evaluation: Overview and Case Studies

Edward McFadden
Danny R. Hager
Claude J. Elie
Jason M. Blackwell
User-Centered Design Services, IBM Global Services

Traditional laboratory-based usability evaluations have many costs associated with them: financial costs, temporary loss in participant productivity, travel issues. For this reason, many human–computer interaction practitioners cannot conduct as many test iterations as they would like or may fail to secure representative samples, especially as user groups become more global. In recent years, we have begun to use less expensive remote evaluation techniques to complement traditional methods. This article describes some collaborative software that allows remote evaluations, along with the necessary hardware, to reproduce, as closely as possible, traditional evaluation techniques. It also describes the types of activities that lend themselves to this sort of approach and provides tips for conducting these sorts of sessions. Finally, this article provides data that suggest remote techniques are not only effective at improving interface designs, but also produce results comparable to traditional approaches for typical measures.

1. ANALYSIS OF REMOTE USABILITY TESTING IN PRACTICE: BENEFITS AND LIMITATIONS

The audience for many types of software continues to broaden to include users from a wide variety of geographies. As the user audience becomes more diverse, new challenges for user-centered design (UCD) are introduced, not the least of which is increased cost. Although the cost of following a UCD process is a very wise investment (Karat, 1993), a practitioner is not always in a position to appropriate the funds necessary for carrying out a full suite of usability activities.

The traditional UCD approach, which typically involves either bringing participants into the laboratory or traveling to user locations, entails a number of costs, some of which are the following.

Request for reprints should be sent to Edward McFadden, User-Centered Design Services, IBM Global Services, Mail Drop 272, Southbury, CT 06488. E-mail: emcfadd@us.ibm.com

Travel costs. Paying for travel, particularly when that travel may be transcontinental, requires a substantial level of funding. Even if only a couple of participants from a few geographies are included, the costs can quickly reach tens of thousands of dollars. Travel costs may also be exacerbated when the user audience consists of a very specialized population, in that they may be geographically disparate.

Productivity costs. Requiring users to be away from their jobs for 2 to 3 days can be quite burdensome. This is especially true for positions such as sales, where participants may actually lose income by attending the test, and their managers are usually somewhat reluctant to allow them to miss work for a usability activity.

Opportunity costs. Planning and scheduling of activities can be difficult and time consuming, given that users need a substantial amount of notice if they will need to travel for the session. This increases the difficulty of following an iterative design process, because it is not always known whether another iteration will be necessary.

These costs can lead to difficult choices for the practitioner. For example, do you accept a less representative user group (e.g., eliminate geographic participation) to increase the number of iterations that can be performed? Do you modify your usability goals because you will not be able to perform additional iterations? Because of difficulties such as these, we have explored the usage of collaborative software (detailed in a later section), which allows the practitioner and participant to "share" their computer screens. Using such software, an administrator in New York can view, in real time, the performance of a participant in Melbourne, Australia or Paris, France. Representative users are able to participate in usability activities without leaving their desks, resulting in no travel costs and minimal loss of productivity. Scheduling participants is greatly simplified (although the administrator must exercise some flexibility due to time zone differences), and we have found that users are much more willing to volunteer. Furthermore, participants perform the evaluation in their own environment with their own equipment, eliminating any questions of how well the test environment matches the real-world environment. In short, using this approach allows the practitioner to perform a substantial number of iterations in less time and at a significantly reduced cost, and thus provides a valuable complement to traditional testing. Subsequent sections discuss common usability activities along with the strengths and weaknesses of doing them remotely. Finally, data from some case studies are presented that speak to the effectiveness of remote testing.

2. CURRENT USE OF THE REMOTE APPROACH

Remote usability evaluation is a relatively new technique and thus lacks a long history of basic research. In 1994, Hammontree, Weiler, and Nayak (1994) described a method for viewing a user's screen over the network to conduct a usability test. By 1996, there were a variety of methodologies and technologies available to conduct a various types of usability evaluations. Some examples include remote questionnaires and surveys (Hartson, Castillo, Kelso, Kamler, & Neal, 1996), video conferencing (Hartson et al., 1996), and user-reported critical incident method (Castillo, Hartson, & Hix, 1997).

Hartson et al. (1996) provided an examination of a number of remote evaluation techniques. In addition to local evaluation (i.e., traditional usability evaluation), described as the benchmark with which to compare, they described local evaluation at remote locations, remote questionnaire–survey, the use of collaborative software for evaluations where the administrator is remotely located from the participant (as described in this article), video conferencing as an extension of the usability laboratory, instrumented remote evaluation, and semi-instrumented remote evaluation (in which the application itself includes the ability to collect and record usage data). Although they described two case studies evaluating the feasibility of video conferencing and semi-instrumented remote evaluation, there are no case studies of the techniques examined in this article.

Whereas Preston (1999) described nine tools that can be used as remote-control software (of these nine tools, only Microsoft NetMeeting is described in this article), her review of these products is limited, and no data are referenced indicating how well-suited these tools are to usability evaluation. She does, however, touch on a significant problem affecting remote usability evaluation: the corporate firewall. We describe this problem, along with workarounds in a subsequent section of this article.

Similar to prior works, this article describes a variety of remote usability evaluation techniques and three of the tools we used. Unlike some of the earlier works, in addition to describing the software, hardware, and methodologies implemented, preliminary data are presented that suggest remote activities are comparable to traditional lab techniques in relevant ways. Finally, the appendixes provide practical information on how to prepare for and conduct remote sessions.

3. SOFTWARE AND HARDWARE

The following sections describe, in some detail, software and hardware options for performing remote usability evaluations.

3.1 Software

A variety of applications are currently available for remote collaboration. We have, for the most part, used three products: Microsoft NetMeeting, Lotus SameTime, and PlaceWare Web Conferencing. Each of the products has advantages and disadvantages, depending on the testing context and the usability activity being performed.

Microsoft NetMeeting. This is a free product that can be downloaded by participants, and which comes bundled with recent versions of Windows (e.g., Windows 2000, Windows XP). NetMeeting includes the ability to video conference (i.e., view video of one or both locations), to share screens (i.e., allow one user to view the screen of another user from their own workstation), to collaborate (i.e., allow one user to take control of an application on the connected user's system), to use a communal whiteboard (i.e., allow users to draw and write on a virtual whiteboard), and to chat (similar to Internet Relay Chat and instant messaging software).

The primary advantage of NetMeeting is that one user's workstation passes operating system calls to the other user's workstation, meaning that very little data are sent over the network. This results in quick response time: When the participant moves their mouse cursor to a button on their system, the administrator can view that activity almost instantly. The primary disadvantages of this approach are that the administrator and all participants must be running a Windows operating system, must be behind the same (or no) firewall, and have the client software (NetMeeting) installed. (Appendix A provides some specific information for dealing with some vagaries of NetMeeting.)

Lotus SameTime. Lotus SameTime is a commercial product that is currently being used as a collaborative tool and instant messaging application throughout IBM. It includes the ability to send audio and video over the network, as well as whiteboard, chat, and other collaborative functions. Although SameTime includes very similar functionality as NetMeeting, it works in a significantly different manner. Instead of sending Application Programming Interface calls from one system to the other, *screenshots* of one system are sent across the network to the other. An updated screenshot is sent with each change of the screen, resulting in generally slower performance than NetMeeting. It is typically not possible to view mouse or typing behavior as it occurs; for example, the user could move the mouse and click a button without the administrator seeing either action.

SameTime does have two important advantages over NetMeeting. First, installation of SameTime is extremely simple for the participant, as it is can be configured as a browser plug-in. For example, IBM currently runs an "e-Meetings" site where customers and business partners can "connect" with an IBM representative, as long as they have a browser that supports Java. In other words, a participant with a Web browser and an Internet connection can utilize all of the functions contained within the product. Second, although SameTime does require its own server, this makes it easier to work with participants outside the administrator's corporate firewall. Specifically, a SameTime server can be set up outside of a corporation's firewall allowing users from outside the corporation to interact with users within it. The firewall may still pose a problem if the user is behind a second firewall (e.g., their own company's or a personal firewall).

PlaceWare Web Conferencing. PlaceWare is a Web browser plug-in that allows participants to view PowerPoint presentations and Web pages at the presenter's direction. Whereas most Web-based presentation tools are self-paced (e.g., using an application such as Freelance Graphics or PowerPoint to create a "portable" HTML presentation), in that the participant indicates when he or she is ready to proceed, the presenter controls the pace in PlaceWare. Participants cannot "skip ahead," a significant feature when conducting a remote walkthrough with a group of participants.

The primary advantage of PlaceWare over the other tools presented here is that it can be used to perform real-time polling with groups of participants. The administrator is able to present multiple-choice questions to participants; free response questions can also be presented by using either forms on Web pages (similar to a Web-based survey) or by using the built-in chat feature. In other words, you can present a screen and then have an entire group of participants respond electronically to questions about that screen.

3.2. Summary of Software Advantages and Disadvantages

Deciding which of these applications to use for a particular usability activity depends both on the software's inherent advantages and disadvantages as well as on the type of activity being performed. A summary of the more important advantages and disadvantages of the software mentioned previously is provided in Table 1.

Table 1: Advantages and Disadvantages of Three Remote Evaluation Tools

Tool	*Advantages*	*Disadvantages*
Microsoft NetMeeting	It is free Excellent for viewing real time events Can connect computers directly (no server needed)	All meeting participants must have NetMeeting installed Must be running a Windows operating system (Windows 95 or greater) There is a limit to the number of users that can connect, depending on the operating system being used (although you can daisy chain as a workaround)
Lotus SameTime	Easy client set up (browser plug-in) Can support large groups May avoid firewall issues	Commercial product Not practical for viewing real time Must be running Window operating system (Windows 95 or greater) Requires SameTime server
PlaceWare	Can support large groups Easy client setup (browser plug-in) Works on both Windows and Sun Solaris operating systems Built-in data collection via multiple choice polls only Can be set up to allow connection through firewalls	Commercial product–service (expensive) Cannot collaborate (share control across administrator and participant) of application Presentation limited to display frame in browser (i.e., not full screen)

Some general recommendations and advice for conducting remote evaluations can be found in Appendix A.

3.3. Hardware

Voice communication. In most situations, it is highly desirable that the participant and administrator need to communicate verbally. Although NetMeeting and SameTime include the ability to transfer voice over the network, this is rarely feasible. To do so, both the participant and administrator would need quality microphones and speakers, and it appears that using the network for data and voice simultaneously can hamper performance of the system. The simpler and more effective choice is to use the telephone, preferably with hands-free functionality. Both participant and administrator, then, must have an Internet connection (preferably local area network [LAN] or broadband) and a separate voice line.

Videotaping the session. A traditional usability lab includes hardware for recording the participant's screen. This may also be done remotely. If the administrator has access to a traditional lab, they may reap most of the benefits of lab testing without the participant being physically present. Most labs have the necessary requirements (e.g., Internet connectivity, telephones, scan converters) for recording a remote session, thus avoiding the need to invest in additional equipment. However, setup costs for recording remote sessions are relatively low even if a lab is not available.

A standard VCR and scan converter connected to the administrator's workstation can be used to record the participant's on-screen actions (a "double deck" VCR is an economical way to perform limited video editing). The primary disadvantage is that the low resolution of a television and VCR may make it difficult to read or recognize button labels and other small details. If this presents too much of a problem, one could invest in better recording equipment (comparable to what is used in traditional usability labs) or record the session using software like Lotus ScreenCam or TechSmith Camtasia, given sufficient processing speed and storage space (e.g., a large hard drive).

For audio, a recording device can be inserted between the phone console and handset. The lead from the recording device plugs into an RCA converter, which then connects to the audio input of the VCR. One possible setup is shown in the Figure 1. Currently, this setup (double-deck VCR, TV, scan converter, telephone recording device) costs approximately $600.

Logging the test. Use of logging software on the administrator's system is possible, provided that there is enough screen "real estate." Alternatively, using a second workstation for logging the session is an effective approach. The latter approach is particularly important when the participant will be viewing the administrator's screen.

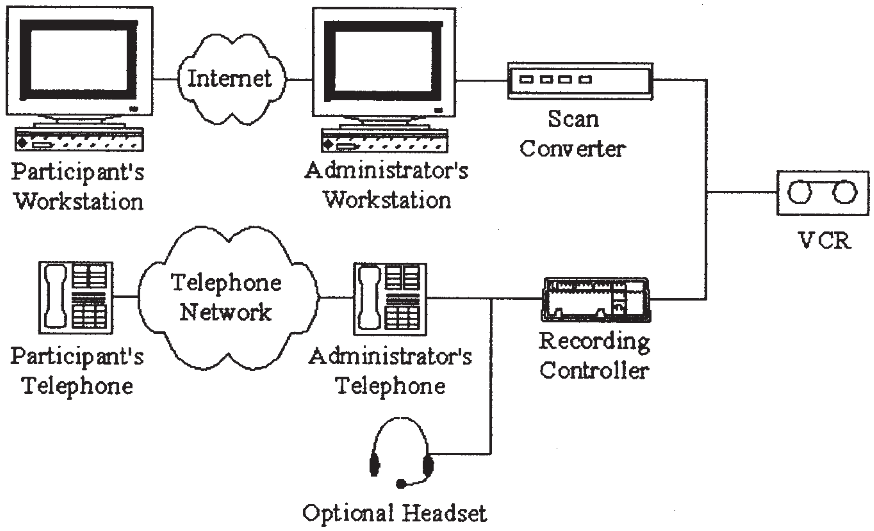

FIGURE 1 Remote usability test.

4. USABILITY ACTIVITIES

In this section we outline several common usability activities and present the advantages and disadvantages of performing them remotely (for a review of these and other traditional usability activities, see Dumas & Redish, 1993; Nielsen, 1995; Nielson & Mack, 1994). Specifically, we have found that participative design sessions, walkthroughs, and usability tests lend themselves to the approach and are covered in this article. Other activities, such as some methods of task analysis, are more likely to require that the participant and administrator be co-located, and are not discussed in this article.

4.1. Participative Design

Participative design typically involves gathering a sample of users in one location and presenting them with low-fidelity prototypes (which may be at the paper-and-pencil level, although more advanced prototypes are not precluded). User feedback is collected on the appearance and function (as well as task flow, etc.) of each screen, and changes based on that feedback might be incorporated "on the fly." Feedback and design iterations are performed within and between sessions until predefined usability goals are achieved. This activity usually occurs very early in the development cycle, often before any coding begins.

Unlike the traditional approach, remote sessions are generally, although not always, one-on-one, in which the participant and administrator connect using applications described in the previous sections. The participant is shown mockups that were either scanned from pencil and paper or created via a graphical editing program, such as Adobe PhotoShop. The design can be modified on the fly utilizing the "virtual whiteboard" included with applications such as NetMeeting and SameTime. That is, the administrator and participant can "draw" and type text on the mockup so that modifications are made immediately (the whiteboard can be saved for future reference).

The participative design activity demonstrates that the remote versus traditional approaches are not mutually exclusive. An effective strategy that we have used is to conduct remote participative design sessions as a follow up to traditional activities. If problems are found with the interface in one activity, changes can be made with user involvement in remote sessions. This alleviates the cost of bringing users back in for follow-up sessions, while ensuring that changes to the interface are validated with users.

4.2. Walkthroughs

The traditional usability walkthrough typically involves the presentation of low-fidelity prototypes to representative users. The prototypes are rarely functional, so users require little or no interaction with them. Instead, the screen mockups are presented in sequence by task flow, whereas the users provide feedback on each screen in turn (e.g., indicating the next action they would take given a particular scenario).

Because the methodology principally involves the viewing of screen mockups (which lack functionality), the walkthrough, more than any other UCD activity, lends itself to the remote approach. The bulk of the work involved in preparing a remote walkthrough is the creation of screens in a graphical program such as Adobe PhotoShop. This may take more time than the preparation of pencil-and-paper mockups, depending on the complexity of the screen and the experience of the test administrator, but creation of the graphical prototypes results in several additional benefits (e.g., the screens can be handed over to the development team as part of requirements).

Once the screens are created, the administrator presents them to the participant using collaborative software; specifically by sharing an application on his or her workstation with the participant. For example, one can create a series of Lotus Freelance or Microsoft PowerPoint slides, and then share that application with the participant. The participant simply views the screens and responds to administrator questions.

4.3. Usability Testing

Traditional usability testing, an activity typically performed late in the UCD process, involves having participants individually attempt to complete realistic scenarios on high-fidelity prototype, beta, or finished application. Testing is usually performed in a usability lab, which gives the administrator great control over the environment (e.g., a particular workstation with specific software installed) and the ability to collect a wealth of data (e.g., recording a participant's voice, facial expressions, mouse and keyboard behavior, on-screen actions, etc.). Ubiquitous measures reported from such studies include objective measures such as error rate (for both critical and noncritical errors), time on task, as well as some subjective measures (e.g., satisfaction or ease of use ratings).

The remote approach involves using collaborative software's ability either to (a) share an application on the participant's workstation or to (b) grant control of an application on the administrator's workstation to the participant (i.e., collaborate). The latter approach is particularly desirable when the application being evaluated cannot reasonably be installed, accessed, or run on the participant's system. Also, if a participant cannot directly access the application to be evaluated, an effective strategy is to grant them control of the application on the administrator's workstation. For evaluating Web sites, the former method is preferable in that the participant is able to use their own browser with the settings they typically use.

5. SUPPORT FOR THE REMOTE APPROACH

Support for the remote approach can be provided in two ways: by demonstrating that the results of an evaluation do not meaningfully differ based on whether the methodology was traditional or remote, as well as by showing that the remote approach can lead to substantial improvement in the usability of an application. Although we have been utilizing the remote approach, out of necessity, since approximately 1999, rarely have we had the opportunity to compare the traditional and remote approaches.

Recently, however, we performed two usability tests in which there were a sufficient number of both local and remote participants to conduct an initial comparison. For both studies, complex applications were tested. The following sections provide details about the two tests, along with an exploration of the correlation between testing methodology (traditional vs. remote) and several common measures (number of critical errors, number of noncritical errors, and time on task).

Note that these studies were not performed with the goal of comparing remote versus traditional methodologies. Instead, administrators tested participants locally where possible, and used the remote methodology (NetMeeting, in these cases) to ensure appropriate coverage of user groups and geographies. The user groups were large enough in these studies that we tested a fairly large sample (as opposed to the more typical four to eight participants), which provided enough data to make comparisons between the testing approaches. In other words, there was no attempt to counterbalance variables such as administrator across the testing conditions.

5.1. Case Study 1

The first test involved a baseline test of new e-mail client interface. For this test, three administrators tested 22 participants. Seven of the sessions were conducted in the laboratory, with the remaining 15 conducted remotely. Table 2 summarizes the results of the two methods across participants and scenarios for mean critical

Table 2: Case Study 1 Summary Data

Condition	*Critical Errors (Mean)*	*Noncritical Errors (Mean)*	*Time on Task (Median Seconds)*
Traditional	0.26	0.73	242
Remote	0.20	1.54	232

errors, mean noncritical, and the median time on task for scenarios that did not produce a critical error (i.e., number of seconds to complete a scenario for all scenarios that were completed correctly).

As suggested by Table 2, the only significant correlation for testing condition is with mean noncritical errors, Pearson $r(282) = .174$, $p < .05$. Participants in the remote condition were more likely to have noncritical errors attributed to them than in the traditional condition. On closer examination, this correlation seems to be due to the fact that one administrator (who ran most of the remote participants) is, across scenarios and case studies presented in this article, more likely to attribute noncritical errors than were the other administrators. For this test, there was a significant correlation, $r(282) = .230$, $p < .05$, between administrator and mean number of noncritical errors. In other words, the correlation observed between traditional and remote testing is likely due to having not counterbalanced administrator across conditions.

The results of this comparison appear to be quite encouraging. We would be pleased to see such a high correspondence between the data collected even if the comparison were between two laboratory tests. The similarity of the mean critical errors, in particular, indicates to us that running sessions remotely does not introduce significant artifacts.

5.2. Case Study 2

The second test was conducted on a new financial application, which was intended to replace three separate methods being used by the customer. In this case, four administrators conducted individual sessions with 24 participants. Eight of the sessions were conducted either in the laboratory or by using a mobile usability lab. Sixteen sessions were conducted remotely. The results of the test, by Testing Methodology, are summarized in Table 3.

For this test, the only variable that yielded a significant correlation, $r(75) = .262$, $p < .05$, with Testing Methodology was time on task. Unlike the prior study, there was no significant correlation between Testing Methodology and noncritical errors. Although there was still a significant correlation, $r(147) = .289$, $p < .05$, between ad-

Table 3: Case Study 2 Summary Data

Condition	*Critical Errors (Mean)*	*Noncritical Errors (Mean)*	*Time on Task (Median Seconds)*
Traditional	0.81	3.98	532
Remote	0.68	4.14	760

ministrator and the number of noncritical errors, the administrators were more evenly distributed across test Testing Methodology. In other words, the confounding of administrator and Testing Methodology was greatly reduced in the second study, in that the administrator who counted more noncritical errors ran both remote and local participants. (The other three administrators counted nearly equivalent numbers of more noncritical errors across the scenarios.)

It has been our experience that time on task is one variable that may be compromised when using the remote approach. The reasons for this are that network performance may be slow or that the connection between participant and administrator may be lost. When these problems occur, the administrator may have to ask the participant to stop their actions until the systems are again synchronized. This presents two problems: inaccuracies are introduced (when stopping and restarting the timer) and the participant may lose track of what they were doing when they were forced to pause. Both of these problems serve to increase time on task. These problems do not occur often, but as the length of the scenario increases, so too does the likelihood that they will have an effect. Consider that the average time to complete a task in Case Study 1 was approximately 4 min, whereas the time required to complete a scenario in Case Study 2 was about 9 min. This difference in absolute time is likely the reason that there was a correlation between condition and time on task only in the second case study.

The similarity between the number of critical and noncritical errors found across the two conditions is, we believe, very supportive of our initial belief that the two methodologies produce functionally equivalent results. The differences between the two conditions are small; and, just as important, we found no qualitative differences between the types of the errors discovered in the two approaches. As is common in traditional testing, the majority of the errors observed with one participant were extremely similar to the errors produced by another participant—regardless of the condition in which they participated.

5.3. Case Study 3

The former case studies focused on the similarity of results produced by the remote and traditional approaches. The final case study focuses instead on whether the remote approach is effective in producing more usable interfaces.

This study involved the iterative design of a Web-based security application, which allows users to request access to sensitive, restricted data. Two walkthroughs and a usability test were conducted in an iterative fashion using NetMeeting (participants were situated across the globe—Europe, Asia Pacific, and North America—which necessitated that the evaluations be performed remotely). Performance on the second walkthrough showed 19% higher ease of use ratings and an 18% improvement in completion rate. The usability test showed a further improvement (relative to the second walkthrough) of 13% in ease of use ratings and 25% in completion rate (which reached 100%). These results indicate to us that the remote approach can certainly be used to implement many parts of the UCD process in an effective fashion.

5.4. Summary of Case Studies

The studies described previously provide converging evidence that the remote approach is worthwhile. The differences in data collected seem quite small (unless time on task is a measure of great importance and the tasks tend to take several minutes to complete), and the approach clearly leads to significant, substantive usability improvements.

6. CONCLUSION

Although there is currently very little research on remote usability evaluation as we discuss it here, we are convinced that this approach is effective; especially in cases of very limited usability budgets, which otherwise might result in little or no iterative design or the exclusion of key user groups. The data collected by the two approaches are very similar for most measures and there is little doubt that the approach can be used to facilitate a UCD process, resulting in better usability of interfaces.

This does not mean that remote techniques can completely replace traditional approaches. First, as noted earlier, we do not feel that these two approaches are mutually exclusive and in fact complement each other nicely. However, some UCD activities (e.g., some methods of task analysis) are difficult or impossible to conduct remotely as described previously. Furthermore, the traditional usability lab offers experimental control and types of data collection (e.g., simultaneous recording of the participant's face, hands, and monitor) that is unmatched by remote approaches. Obviously, those doing more basic research or focused on less commonly reported participant's actions would still want to employ traditional approaches.

Finally, although we provide prima fascia evidence that remote and traditional approaches produce similar results, our work was focused on improving the UI designs of the products tested. More controlled studies should be conducted to compare these approaches. The remote approach will continue to evolve in concert with technology, and controlled studies comparing different remote approaches would be useful to many practitioners.

REFERENCES

Castillo, J. C., Hartson, H. R., & Hix. D. (1997). *Remote usability evaluation at a glance.* (Tech. Rep. No. TR–97–12). Blackburg: Virginia Tech.

Dumas, J. S., & Redish, J. C. (1993). *A practical guide to usability testing.* Norwood, NJ: Ablex.

Hammontree, M., Weiler, P., & Nayak, N. (1994). Remote usability testing. *Interactions, 1,* 21–25.

Hartson, H. R., Castillo, J. C., Kelso, J., Kamler, J., & Neale, W. C. (1996). Remote evaluation: The network as an extension of the usability laboratory. *Proceedings of CHI'96 Human Factors in Computing Systems,* 228–235.

Karat, C. (1993). Usability engineering in dollars and cents. *IEEE Software, 10,* 88–89.

Nielsen, J. (1995). *Usability engineering.* New York: Kaufmann.

Nielsen, J., & Mack, R. L. (Eds.). (1994). *Usability inspection methods.* New York: Wiley.

Preston, A. (1999). Remote usability testing tools. *Usability Interface, 5,* 5.

APPENDIX A

LIMITATIONS AND WORKAROUNDS FOR REMOTE SESSIONS

Conducting a session remotely entails some different requirements than conducting a traditional session. The primary differences and challenges (along with some suggestions for overcoming those challenges) are described.

A portion of the preparation is passed to the participant. It is vital that enough time is allowed for the user to install the necessary software, print scenarios, and so forth. We have prepared a packet of information and instructions, that with minor modification, can be sent to a participant for any session, and since doing so, have had very few problems with the participant being ill-prepared for the session. Typically, the instructions are sent to the participants several days in advance, and a reminder is sent the day before the session. (We provide sample instructions by request.)

A reliable network and stable system are practically essential. If problems arise, the participant may become frustrated and impatient—making it difficult at best to interpret their results. When possible, be sure that no programs are running in the background, and try to schedule participants during times that have less network traffic. Whenever possible, the session should utilize broadband or LAN connections. A fast dial-up (at least 36 kbps) is usually workable, but distance between the participant and administrator may play more of a factor.

If the administrator and participant are separated by one or more firewalls, then it is impossible to conduct some forms of testing (e.g., by using NetMeeting). However, there are some workarounds that have resulted in our turning away remarkably few participants:

- Use an application such as SameTime, running on a server that is accessible to the public.
- Have the participant travel to a nearby corporate facility, where a confederate can give them access to a workstation.
- The administrator may conduct the session using a connection outside the corporate firewall (e.g., at home, where the increasing ubiquity of broadband connections has made the approach more feasible).

The screen resolution of the participant and the administrator might differ. If the administrator's resolution is higher than the participant's (which is most often the direction of the disparity), and the administrator is sharing an application, the participant will not be able to see the entire application screen. In most cases, the simplest solution is to determine whether the participant can see the entire screen at the start of the session (by asking about landmarks in each corner) and altering the administrator or participant's resolution if necessary. In most cases, the participant does not understand the concept of resolution and would need to be guided through the process of changing it (and returning it to “normal” after the session); for this reason we recommend that the session administrator make the modification to their system.

Issues specific to NetMeeting may arise. Due to the variety of Windows versions (in our experience, a single test frequently involves three or more versions of Windows), there are some idiosyncrasies that may occur when the administrator and participant are running different versions of the operating system or NetMeeting. Presented in the following are some general tips for dealing with NetMeeting irregularities:

- If the versions of NetMeeting differ, have the person with the more recent version of NetMeeting initiate the call. The mismatch in versions almost always means that the person with the earlier version of NetMeeting will not be able to control applications on the other system. If you have trouble connecting, try having the other person initiate the call, or restart NetMeeting and try again. If this fails, a reboot will often fix the problem.
- If you are conducting a session with more than one person, the facilitator should ’host’ the meeting. Otherwise, each participant will have to accept new participants into the meeting.