[9] Norman, D. A. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine.* Reading, MA: Addison-Wesley.

[10] O'Conner, R. J. (1988). Apple's view of the future is troubling. *San José Mercury News*, Sunday, November 27, p. 1F.

[11] Salomon, G. (1990). How the look affects the feel: Visual design and the creation of an information kiosk. In *Proceedings of the Human Factors Society 34th Annual Meeting,* Orlando, FL, October 8–12. Santa Monica, CA: Human Factors Society, pp. 277–281.

[12] Schank, R. C. (1990). *Tell Me a Story: A New Look at Real and Artificial Memory.* New York: Charles Scribner's Sons.

[13] Schön, D. A. (1987). *Educating the Reflective Practitioner.* San Francisco: Jossey-Bass.

[14] Schrage, M. (1993). The culture of prototyping. *Design Management Journal,* Winter, pp. 55–65.

[15] Thimbley, H., Anderson, S. & Witten, I. H. (1990). Reflexive CSCW: Supporting long-term personal work. *Interacting with Computers, 2*(3), 330–336.

[16] Tognazinni, B. (1994). The "Starfire" video prototype project: A case history. *Human Factors in Computing Systems: CHI '94 Conference Proceedings.* Reading, MA: Addison-Wesley, pp. 99–105.

[17] Vertelney, L. (1989). Using video to prototype user interfaces. *SIGCHI Bulletin, 21*(2), New York: ACM Press, pp. 57–61.

[18] Wong, Y. Y. (1992). Rough and ready prototypes: Lessons from graphic design. *Human Factors in Computing Systems: CHI '92 Conference, Posters and Short Talks.* New York: ACM Press, pp. 83–84.

[19] Wong, Y. Y. (1993). Layer tool: Support for progressive design. *Human Factors in Computing Systems: CHI '92 Conference, Adjunct Proceedings.* New York: ACM Press, pp. 127–128.

CHAPTER 3

# Scenarios in Discount Usability Engineering

**Jakob Nielsen**
*SunSoft*
*2550 Garcia Avenue, MS MTV 21-225*
*Mountain View, CA 94043-1100*
*USA*

## 3.1 INTRODUCTION

Usability engineering is the systematic approach to improving the usability of user interfaces by applying a set of proven methods throughout the system development lifecycle [15]. A key element of this development philosophy is the observation that user interfaces do not become good merely because the developers wish for it or have the best of intentions. In contrast, it is necessary to employ a variety of usability methods for each step of the lifecycle, from the very first product concept to the postdeployment stage where the product is studied in the field in order to gain knowledge for the next release.

Even though it sounds appealing to be systematic and to rely on proven engineering methods, experience shows that realistic development projects do not have the time, resources, or personnel to do everything that is normally recommended to improve usability. Thus, there is a need for *discount* usability engineering—that is, methods that are cheap, fast, and easy to use but still achieve most of the intended effect. In most projects, if the only available choice was that between nothing and perfection, nothing would win, but discount usability engineering offers a third alternative that can be chosen to achieve substantial improvements in usability while staying within budget and schedule [11, 12, 15, 16, 18].

For use in discount usability engineering, scenarios have the advantage of limiting the usability engineer to looking at a few (hopefully characteristic and/or important) examples of interaction as opposed to the general case where considerably more data and variety have to be considered. Of course, no usability activities can ever be completely general and consider all possible cases and variations in user and task characteristics, but normally one would try to accommodate the main expected sources of variability. Being as general as possible has the advantage of increasing the probability that the resulting conclusions are valid in all important cases. Unfortunately, the likely downside is that too much generality leads to too few conclusions because the usability specialist gets bogged down in handling the complexity resulting from the interplay of the multiple variables. My preferred alternative is to focus on a smaller and more manageable scenario and then be aware of the need to interpret the results with care. One way of assessing the likely scope of scenario-based results is to study a few scenarios that have been chosen to span the expected range of uses and then conduct a sensitivity analysis on the outcomes. If two very different scenarios lead you to basically the same conclusions with respect to the interface, then those conclusions will very likely hold under many additional circumstances.

My definition of the term *scenario* [13] is an encapsulated (that is, self-contained, portable) description of

- an individual *user*
- interacting with a specific set of computer *facilities*
- to achieve a specific *outcome*
- under specified *circumstances*
- over a certain *time interval* (this in contrast to simple static collections of screens and menus; the scenario explicitly includes a time dimension of what happens when).

For example, together with Jan Maurits Faber, I recently investigated user interfaces to various advanced telephone services, including three-way calling. In our studies, we focused on setting up the three-way call and we did not study what happened after the conversation started. Thus, we were able to simulate the other two users with digitized speech on the computer (they always say the same thing when they answer the phone) and run our tests with a single user. In essence, we viewed the other two users as components of the total system with which this individual interacts, which is why the scenario could be seen as the experience of an individual user.

Within the discount framework, I have a bias in favor of single-user scenarios as they are often the cheapest. On the other hand, there are obviously some user interfaces that depend so much on the interaction between multiple users that it would be hard to analyze them solely from a single user's perspective. A study of three-way calling should be seen from a multiuser perspective if one is interested in the group dynamics of negotiations or problem solving mediated through a telecommunications medium; another example could be the use of a shared electronic whiteboard as a conversational prop [2]. In cases like these I would expand the scope of the scenario to include the experience of all the users.

I will discuss seven applications of scenarios in discount usability engineering:

- Diary scenarios as cheap ways of gathering preliminary field data
- Support for feature brainstorming to envision new directions for computers
- Presentation and contrast of user interface ideas as part of the parallel design method
- Allowing heuristic evaluation of highly domain-specific interfaces by usability specialists without domain expertise
- Scenarios as a simplified prototyping technique for user testing
- Scenarios as a unifying background for a set of tasks in user testing
- Stereotypes for data analysis in exploratory studies

This list has been sorted according to the approximate stage of the usability engineering lifecycle in which the various kinds of scenarios are used. The discussion of the scenarios in the following deviates slightly from this sequence since my example of the last of the scenario uses (exploratory data analysis) is related to the electronic door project used as the example of feature brainstorming and is therefore discussed in the context of that project.

## 3.2 DIARY SCENARIOS AS DISCOUNT FIELD STUDIES

One of the first stages of the usability engineering lifecycle is a field study of how users currently behave with respect to the tasks of interest to the project. One discount approach to such studies is to build diary scenarios that list those activities and situations a user encounters during a day that are expected to be relevant for the project. Note that the diary is not supposed to be a complete description of *everything* that happened that day since a complete diary would be very expensive to construct. Some important aspects of the user's day will be left out of the diary, so this discount method should be supplemented with alternative usability methods like direct observation for important projects. The diaries can either be written by users themselves (the cheapest approach), or they can be written by an observer. Figure 3.1 shows a single entry from a diary scenario I wrote as part of a long-term project to assess the needs of personal computing in the generation of systems that will follow the object-oriented operating systems currently under development. One of the expected characteristics of such systems is the need to handle extremely large amounts of information leading to the potential to use information retrieval techniques and attributes of the user's personal interaction history as important aspects of the primary user interface.

The example in Figure 3.1 can be used to derive several lessons, including the following: People may remember information in terms of the time when they accessed it (in this example, the year 1989), but they may not remember all details correctly (here, the conference was

---

8:50am: Received an e-mail message from a colleague in Palo Alto (actually sent the night before, but I happened to not have logged in until now) who urgently needed a reference to one of my papers which she seemed to remember having seen as a demo at the Hypertext'89 conference. Note how she uses me as her personal retrieval system! Her description of the system (including the keywords "netnews," "similarity," and "Hypertext'89" (and of course my name) was more than sufficient to immediately allow me to realize that she wanted my paper "A similarity-based hypertext browser for reading the UNIX network news." I went to my filing cabinet and opened the correct drawer (where I keep reprints of all my hypertext papers), and found the relevant folder and picked out a reprint. I then e-mailed her the reference. She indicated that she needed the paper urgently, so I also found the file with the paper to e-mail it to her: This file was in my directory called "papers" under the subdirectory "archived papers" (for papers that have already been published). I scanned an alphabetical list of files in this subdirectory until I came upon the file "HyperNews similarity browser." Note that HyperNews was the name of the system and was very salient to me when I named the file back in 1989. Also note that this system was in fact never demoed at the Hypertext'89 conference, even though it was a paper published in the *Hypermedia* journal in 1989.

---

**Figure 3.1.** Sample entry from a diary scenario. The diary was written to focus on personal information retrieval in the day of a scientist.



remembered instead of the journal). Also, the information that seemed most salient as a classification term several years ago (here, the application name HyperNews) may not be used during subsequent retrieval attempts. Furthermore, people tend to interact and use each other as personal databases, so retrieval systems might be networked to agents representing close colleagues, friends, and family in case the information is not found on the user's personal system.

There is an obvious danger in relying on personal diaries, especially if they are collected from user interface scientists and others who are not necessarily representative of the general set of intended users: Not only may people describe actions or situations in ways that misstate or miss important aspects, but people willing to take the time to keep a diary are likely to act differently from those users who are not willing or able to participate in the study. Even so, diary scenarios are a fast and cheap way to get some preliminary data before more elaborate field studies are staged. Also, there is some value to collecting information about the habits of advanced users as they may provide hints for ways computers could support more average users and augment their skill.

Diary scenarios differ from a-day-in-the-life scenarios [5, 27] by describing real observations rather than envisionments. Also, they focus on the experience of particular individuals while abstracting away most of the details of that individual's work (since it would otherwise be too burdensome to write the diary). Diaries have also been used for field studies of program development [9] by asking professional programmers to keep track of daily events in their projects as a way to study larger issues than those typically addressed by laboratory studies where student programmers debug 50-line programs. A more structured version of the diary method is to provide participants with preprinted diary log sheets listing various activities of interest to the study. Participants are then asked to make a note in the log every 15 or 30 minutes and check the activities they have been performing since the last note [24]. This structured approach is "quite effortful" [24] and can be used only for a fairly short duration of about one or two weeks. Typically, the log sheets are supplemented with debriefing sessions at the end of each day or each week where the participants are interviewed in more depth about the activities recorded in the diaries.

## 3.3 SCENARIOS IN FEATURE BRAINSTORMS

As part of the *beyond being there* project [3, 6], I have been researching ways of managing an electronic presence that can represent a person, a project, a company, or some other entity to the outside world. The project is called the *electronic door* because the computer facilities that were investigated can be seen as a parallel to the way a physical door represents the occupant of an office to people passing by in the corridor. Many people post items of interest on their doors, such as their name, title, or department; office hours or an indication that they are on vacation and when they will return; as well as more or less whimsical posters, pictures, and cartoons that represent their interests.

As can be seen from this brief description, the electronic door is a futuristic computer system that will probably be significantly different from most currently used computer systems.[1] It is impossible to gather sufficient information about such a system by studies of current systems (though some field studies will be useful), and it is completely infeasible to implement an adequate spectrum of running systems to gain experience from the actual use of electronic doors. We therefore decided to use a scenario-building exercise to clarify our thoughts about the electronic door and its potential features and capabilities. Nine researchers participated in a series of five brainstorming sessions, each lasting between one and two hours. In order to approach the development of the scenarios systematically, we used the method described by Tognazzini [26], where scenarios are gradually built by constructing lists of

- Intended users of the system
- User dichotomies (endpoints of various dimensions on which users differ)
- User experience (the various types of background users can be expected to have)
- Users' goals in performing their tasks
- Sources of information accessed by users when performing the task
- Information generated by the users
- Methods employed by the users while performing the task

[1]Actually, at the time of this writing (1994), many users have started constructing *home pages* on the World Wide Web that seem to satisfy some of the needs we identified in the electronic door project. At the time of the project (1992), the World Wide Web and Mosaic had not yet reached their current level of prominence on the Internet [20] and the home page phenomenon was not studied in the project. It is an interesting exercise to compare the project scenarios for future electronic doors with the actual home pages observed two years later: Even though many of the more advanced features are not (yet) seen in practice, the growth of the World Wide Web happened faster than we had expected.



- User needs
- Possible usage scenarios

Some examples from the various lists are reproduced in Figure 3.2 (the full list contained 212 entries). Each of the list elements was effectively a *microscenario* constructed with very low overhead (just writing down a few words on a whiteboard or an overhead foil) and discussed for a few minutes only to envision some specific aspect of electronic doors. These microscenarios made sense to the brainstorm participants in the context of the original sessions since each gave rise to the next, but some of the original richness was lost upon considering the lists in later sessions where most of the context had been lost. The advantage of the method is that it allows for the exploration of a large number of different microscenarios in a short amount of time, with the corresponding disadvantage that each individual scenario is fleshed out only in minimum detail.

Note that the microscenarios may not make sense to people who did not participate in the meetings. For example, one entry under possible users is "Fans of Don Norman." Norman is a cognitive scientist who is famous for his critiques of bad user interface designs and has a wide following on the Internet. (Every month there will be postings asking "What is the title of Norman's new book?" or some such question.) Since our design team was dominated by user interface specialists and cognitive scientists, we happened to think of Norman when we needed an example of a celebrity whose doings people might be interested in keeping track of. In the world at large, a film star would definitely have been a better example.

From the list of possible usage scenarios, we finally picked three scenarios to develop further. The first scenario concerned the use of electronic doors in a family, including both members of a nuclear family and members of their extended family, such as grandparents. The second scenario was business-oriented and concerned interactions between a company and its clients wanting information about one of its products. The third scenario was referred to as the *ultrafuture scenario* and followed a day in the life of a person in a potential future world where everybody and everything would have electronic doors.

Each detailed scenario was basically a list of situations and system features to react to those situations. For example, the business-oriented scenario started out with assuming that a person at one of Bellcore's client companies (MaryX) had heard that Bellcore was supplying a great hypertext product capable of handling large amounts of text and telephone company documents. In order to find out more infor-

*Who are the people who might want to use electronic doors:*

- Computer science research scientists
- Employees of distributed companies (or even nondistributed companies)
- Historical people
- Composite entities, clubs, joint pseudonyms
- Programs, robots, AIs, Answer Garden
- New people in an organization
- Support people (benefits people, hotline staff)
- Computer dating/personals
- Service advertising which doors to connect to
- Nonindividuals, companies/organizations looking at each other
- Political campaigns
- Fans of Don Norman (wanting to read whatever books he checks out of the library)

*Dimensions along which different classes of users might differ:*

- Zero-time to maintain my door versus "my door is my life," so I will spend endless time maintaining it
- Lots to express versus nothing to say
- Users at fixed location/workstation versus people on the go
- Telling the truth versus lies/half-truths
- Fully automated versus completely manual door construction

*Existing technologies and concepts users know that have similar attributes:*

- Answering machines in movie theaters giving times for shows
- Real doors with things on them
- Bulletin boards
- Business cards
- Clothing selection
- Birth, wedding, death announcements
- Tombstones, epitaphs
- Autobiographies
- Censorship (of information posted on doors)
- Security, privacy, skepticism

*Why would people want to use electronic doors?*

- Make friends
- Influence people
- Keep in touch with friends

**Figure 3.2.** Examples of microscenarios built as part of a brainstorm on possible uses of the electronic door. The notes are basically shown as they were written on the whiteboard during the scenario sessions, meaning that they are sometimes very brief.



- Find out facts without bothering people
- Let others find out about you
- Barrier to prevent people from bothering you
- Create a persona
- Get others' opinions/experience (e.g., movie reviews, recommendations)
- Post and find out reactions/opinions to external events/things
- Surrogate for self on job

*Where would users get the information they would post on the electronic door?*

- Scanned images
- Photos from digitizing camera
- Net articles
- Previously written papers
- Demons watching your computer activities
- Infrared detectors
- Smart badges that transmit information about where the user is

**Figure 3.2.** *(Continued)*

mation about this product, she connected to Bellcore's corporate electronic door, which allowed her to find the electronic door for the specific product. From this door, MaryX could get further information about the product, and a demo, and connect to the electronic door of various Bellcore employees with further expertise. The first Bellcore person she tried to contact was on vacation, but since the electronic door of that person had a message saying so as well as a reference to the electronic door of another person who could handle urgent business, MaryX succeeded in getting the information she needed. The full scenario had several other details, including a discussion of social issues related to some aspects of the envisioned use of the electronic door. For example, we hypothesized that one reason MaryX started by connecting to the electronic door was that she wanted to check out the product without the commitment implicit in asking somebody to come out and show it to her.

Developing these scenarios was a worthwhile exercise, resulting in an increased understanding of the concept of an electronic door. At the same time, the participants in the brainstorming sessions felt that the systematic list-building method did not fit very well with our project and its very undefined nature. Certainly, some good came of trying to be systematic. For example, when listing possible users, we had to consider whether a historic person like Napoleon could have an electronic door (the answer was yes). Having included historic persons on the list of users again led to a user dichotomy between living and dead users, which again led to a discussion of the general issues involved in having electronic doors for dead people who might want to leave an electronic epitaph.

Mostly, though, our feeling was that the list-building format was better suited for the development of scenarios for more constrained types of systems. In our case, there were no real answers, and we constantly felt the urge to develop slightly larger *miniscenarios* for each item added to a list as a microscenario (for example, *how* a dead person's electronic door would look and why people might want to build it). Thus, it may be the case that scenario building for highly unconstrained development projects should start with the detailed specification of a few specific scenarios to clarify the product concept before more detailed lists are constructed.

In any case, scenario building did seem to be a valid method to brainstorm about desired features and possible uses of a new product that had yet to be defined. After the sessions we had a much better idea of the possible uses of electronic doors and possible paths for the project. Unfortunately, it did not seem that the scenarios helped us decide between these possibilities.

The scenario-building exercise utilized the fact that the human imagination is the cheapest multimedia prototyping system around. Just a few words serve to call up reasonably vivid pictures and imagined interfaces. Of course, reliance on human imagination as a prototyping tool leaves the system even more underspecified than more traditional prototyping, so the different participants in a scenario-building session may have somewhat different understandings of the system that is being discussed. These differences emerge over time as more detail is added and more examples are discussed.

### 3.3.1 Field Study of Prototype Design

Given the vagueness of the scenarios, it was decided to implement a limited version of the electronic door called an *electronic business card* to serve as a vertical prototype.[2] A full-featured electronic door would presumably be some kind of highly interactive hypermedia system allowing users to connect to remote servers in real time and navigate their information spaces through a graphical user interface that would

[2]Vertical prototypes have fully working functionality and can thus be used for real but are limited in only supporting a small part of the features in the full system.



be difficult to implement and could only be tested if all users had access to the same platform. In contrast, the electronic business card was envisioned as a more modest version that was accessed through electronic mail (instead of through a real-time dynamic interface) and might initially be text-only (thus making it accessible worldwide without any special equipment). The eventual use of electronic business cards is expected to come with the widespread use of personal digital assistants (PDAs). Assuming that all participants in a meeting bring a PDA and that these PDAs can connect to each other through some kind of wireless network (e.g., infrared beaming), it will be possible for the PDAs to exchange electronic business cards for their owners. Upon leaving the meeting, each participant will thus have a complete list of electronic business cards for all other participants stored in his or her PDA.

Many considerations lead to the conclusion that the information exchanged between the PDAs will not be the complete set of information available for a given participant. Bandwidth and storage limitations might limit each record to a few kilobytes, even though many people will have many megabytes of information available about themselves (when video starts being used as a data medium, most people will have gigabytes of information available for distribution). Also, to avoid information overload most people will not want to have the complete set of available information stored for each person they have ever met, even if it were technologically feasible. Therefore, the electronic business card will be a hypertext link to remote servers (called *electronic business card servers*), which can provide more information over the net upon request. Typical information that may be retrieved from an electronic business card server could be a description of the owner's personal and professional interests, including a publication list in the case of scientists, digitized photos, video, and sound clips, and electronic documents of various sorts (for example, paper reprints for scientists, price lists for salespeople, and policy statements and speeches for politicians).

The electronic business card that was actually implemented was limited to working through standard electronic mail. The advantage of this approach is that it was easily accessible to anybody who had e-mail access and that it was reasonably easy to implement. Also, pointers to the electronic business card could be given simply as an e-mail address that could be printed on paper documents such as letterhead and articles in conference proceedings.

As can be seen from Table 3.1, the author's electronic business card server had been accessed by 556 users over twelve months. In contrast to expectations, most users only used the electronic business card a few times and did not return for further information. Of course, it is pos-

**Table 3.1** Usage Statics for Jakob Nielsen's Electronic Business Card for the Period March 22, 1993 to March 20, 1994

| *Domain* | *Users* | *Commands* | *Commands per Users* |
|---|---|---|---|
| .ar (Argentina) | 1 | 1 | 1.00 |
| .at (Austria) | 1 | 5 | 5.00 |
| .au (Australia) | 13 | 69 | 5.31 |
| .be (Belgium) | 5 | 45 | 9.00 |
| .ch (Switzerland) | 2 | 2 | 1.00 |
| .cl (Chile) | 1 | 1 | 1.00 |
| .de (Germany) | 32 | 116 | 3.62 |
| .dk (Denmark) | 6 | 34 | 5.67 |
| .fi (Finland) | 13 | 63 | 4.85 |
| .fr (France) | 6 | 29 | 4.83 |
| .gr (Greece) | 1 | 1 | 1.00 |
| .hk (Hong Kong) | 1 | 1 | 1.00 |
| .ie (Ireland) | 4 | 25 | 6.25 |
| .il (Israel) | 1 | 1 | 1.00 |
| .it (Italy) | 7 | 26 | 3.71 |
| .jp (Japan) | 13 | 57 | 4.38 |
| .kr (Korea) | 1 | 1 | 1.00 |
| .mx (Mexico) | 1 | 20 | 20.00 |
| .nl (Netherlands) | 20 | 91 | 4.55 |
| .no (Norway) | 6 | 24 | 4.00 |
| .nz (New Zealand) | 1 | 43 | 43.00 |
| .pl (Poland) | 2 | 83 | 41.50 |
| .se (Sweden) | 14 | 146 | 10.43 |
| .sg (Singapore) | 1 | 2 | 2.00 |
| .su (Russia) | 2 | 2 | 1.00 |
| .tw (Taiwan) | 2 | 3 | 1.50 |
| .uk (United Kingdom) | 53 | 271 | 5.11 |
| .za (South Africa) | 1 | 1 | 1.00 |
| **Non-US/Canada total** | **213** | **1,165** | **5.47** |
| .ca (Canada) | 19 | 61 | 3.21 |
| .com (Commercial) | 188 | 483 | 2.57 |
| .edu (Universities) | 97 | 288 | 2.97 |
| .gov (Government) | 9 | 52 | 5.78 |
| .mil (Military) | 3 | 4 | 1.33 |
| .net (Network providers) | 12 | 27 | 2.25 |
| .org (Organizations) | 8 | 18 | 2.25 |
| .us (other USA) | 1 | 1 | 1.00 |
| **US/Canadian total** | **337** | **934** | **2.77** |
| BITNET users | 5 | 23 | 4.60 |
| Userids without domain | 1 | 1 | 1.00 |
| **Grand Total** | **556** | **2,123** | **3.82** |

*Note:* Furthermore, 60 commands were received from anonymous users. While the server was in operation, updated versions of this table could be retrieved by sending it e-mail with the line **get user_stats_by_domain.txt**. This reference was itself an example of the use of the electronic business card to support dead hypertext links.

sible that some users may want to return to the server after an extended period of time not captured by the small window represented by the statistics in Table 3.1. The table shows that users outside North America accessed the service considerably more than did users within North America. This result may indicate that people who are further away from a given locus of information have larger needs for rapid, electronically disseminated information, but the tendency is too weak for any real conclusions. However, given the many studies showing that most electronic communication occurs between people who are geographically proximate [1, 8], the contrasting results in Table 3.1 at least lend some support to the beyond-being-there idea.

A closer look at the actual commands issued by the various users indicates that the overwhelming use of the electronic business card server came from cases where specific instructions had been given for accessing a specific piece of information (similar to the footnote to Table 3.1). In other words, the use of the electronic business card server as a hypertext system with half-dead links [20] was much more pronounced than its use as a primitive electronic door, which people contacted to learn more about its owner. This latter result does not necessarily mean that the electronic door idea is doomed, since the failure of the electronic business card to serve as an electronic door may be a function of its primitive, e-mail-oriented interface. Also, the system did see *some* use that followed the patterns expected from our electronic door scenarios.

The electronic business card project served as a reality check on the electronic door scenarios and provided valuable field data about people's actual use of the kind of facilities we are discussing. It confirmed the feasibility of implementing some of the features and provided data showing that some features were more frequently used than others. Thus, even a simple real system can supplement envisioning exercises and help keep them on track as researchers explore future advanced computer capabilities.

As an aside regarding electronic business cards, I learned after having completed the project that a somewhat similar concept is described in the science fiction novel *Snow Crash* [25]. Of course, since verbal fiction is unconstrained by mere implementation details, the version described in the novel has many interesting and advanced features. In effect, reading a science fiction book can serve as a kind of (entertaining) scenario of future use of a system.

### 3.3.2 Scenarios as Stereotypes for Data Analysis

In the electronic door example, the collected data from the users were analyzed according to a few stereotyped scenarios of system usage. For this particular application, it was deemed unacceptable to increase the users' overhead by sending them questionnaires asking them to explain how and why they used the system. Doing so would quickly have caused people to stop using the system. Instead, usage logs were collected automatically by the system. Eyeballing these logs made it apparent that the system was being used in ways that were slightly different from the main usage as envisioned before the start of the study. Instead of a formal data analysis with previously defined hypotheses, the data were analyzed by a kind of pattern matching, where a human analyst built up a model of stereotypical uses in the form of miniscenarios. For example, one stereotype scenario was a user who had read instructions about how to get a specific file and wanted to get only that one information object, and another stereotype scenario was a user browsing the information base to see what was available. Matching the actual log data to these stereotypes was fairly easy and made it possible to gain a rough understanding of the data in a short amount of time. To some extent, stereotype scenarios are similar to use cases [7] in describing possible sequences of actions, but since the stereotype scenarios are intended to capture user behavior rather than system behavior, they are by necessity less well defined, and many different series of user actions could match the same stereotype scenario.



A key element is that the stereotype scenarios emerged as part of the data analysis they were intended to help. As more user observations became available, the stereotype scenarios crystallized and could then be used to match additional data. Some stereotype scenarios were carried over from the initial design brainstorms (summarized in Figure 3.2) though most of these scenarios turned out not to be well represented in the data. Given that a field trial shows new usage patterns (which will very frequently be the case), it will not be sufficient to simply reuse the scenarios that were developed as part of the envisioning part of the project.

The fact that many of the original envisioning scenarios were not found in the data might lead to two different kinds of conclusions, and one would need to rely on general user interface experience and insights to determine which one was most likely to be true. One possibility is that many of the original scenarios were simply not realistic and that users would not want to behave in the ways the designers envisioned. Another possible explanation is that the vertical prototype implementation was too limited in the range of user activities for which it offered support and that more complex user scenarios would be observed only with more complex software. My personal guess is actually a little of both: More advanced software (like Mosaic and the World Wide Web pages in common use today) does indeed lead to more of the predicted usage scenarios, but there are also cases where the design meetings got carried away and generated scenarios that are unlikely to be seen in real use.

## 3.4 SCENARIOS IN PARALLEL DESIGN

Parallel design [15, 22] is a usability engineering method intended to supplement iterative design in its initial stages. It involves having multiple designers develop initial sketches of a user interface design based on a given functional specification. The goal of parallel design is to explore the design space and to be able to build the first complete interface design on the best aspects of several ideas without the constraints inherent in iterating from a previous design.

Each of the resulting designs from a parallel design project can obviously be described in detail with the use of traditional specification languages, whether formal or as informal as a set of annotated screen mockups. However, we have found usage scenarios to be a particularly effective way of quickly communicating a set of alternative designs to a group (for further discussion). A rough understanding of the designs can be achieved in as little as ten minutes per design by having each designer give a presentation stating what would happen as a hypothetical user progressed through a typical interaction scenario using the interface.

In communicating parallel designs, one can use the same scenario for all designs to make it easier to contrast them, or one can use different scenarios that highlight the particular strengths of each design. The choice between these two options will depend on the available time, since it is easier for the individual designers to use scenarios that match their own design style.

## 3.5 SCENARIOS IN HEURISTIC EVALUATION

Heuristic evaluation [17, 21] is a highly informal usability inspection method, where a set of evaluators (preferably usability specialists [14]) inspect a user interface in order to generate a list of the usability problems in the interface. In contract to other, more systematic usability inspection methods like cognitive walkthroughs [23, 28], heuristic evaluation does not depend on previously defined user tasks, since it is based on checking the various interface elements against the evaluators' usability expertise and a small set of generally applicable heuristics (such as, "be consistent"). Even so, it would be very difficult for a usability specialist to perform a heuristic evaluation of a highly domain-dependent user interface if the usability specialist had no understanding of the meaning of the information contained in the various dialog elements.

We have successfully used the heuristic evaluation method for the evaluation of a highly domain-dependent user interface for a telephone company application. The system will be referred to here as the Integrating System (see [17, 18] for more detail). It is intended to be used by highly trained telephone company technicians and the interface is full of cryptic abbreviations and concepts that are basically meaningless to others. The evaluations were made possible by supplying the evaluators with a usage scenario that had been developed on the basis of a task analysis of the users, resulting in a step-by-step description of the handling of a typical task. At each step, the scenario described what information a real user would extract from the current screen, what inferences the user would make, and what operation the user would perform next. Following these detailed instructions, the evaluators were able to use the system even though they did not truly understand what they were doing or why they were doing it. The result of the heuristic evaluation was a list of forty core usability problems in the part of the interface covered by the scenario as well as four additional problems that were found when some evaluators deviated from the scenario. As could certainly be expected, later user testing revealed additional problems, including some that would never have been found by a heuristic evaluator without any domain knowledge. However, the heuristic evaluation could still be deemed a success, since the usage scenario allowed the usability specialists to evaluate many basic interaction principles and find many usability problems.



Heuristic evaluation was originally not scenario-based. On the contrary, to some extent it is one of the method's strengths that each evaluator approaches the interface differently. When a usage scenario is given, it will constrain the ways the evaluators analyze the interface, and some usability problems will be overlooked. Scenarios have many good aspects, but users always find new ways of using things that would not be covered by the scenarios, so there are some reasons to support a nonscenario inspection of an interface. Heuristic evaluation finds many usability problems not found by user testing, and one reason for this is that it is not tied to a predefined task or use of the system.

Even so, there are also times when it is advantageous to use a scenario for a heuristic evaluation. The example mentioned in my abstract was one such case, where the evaluation could not have been done without a scenario. The integrating system was an interface to a highly specialized task with screens filled with obscure codes (that were second nature to the technicians using the system). We could have done a standard heuristic evaluation and contemplated the nature of these codes, but in order to assess the higher levels of the dialog, the evaluators needed to know what the system was *doing* with the codes, and we deemed a scenario the most efficient way of achieving that goal. Another approach would have been to teach the codes and the related properties of the telephone system, but that would probably have taken several months instead of the two hours needed to teach the scenario and thus would not have been in the spirit of discount usability engineering.

## 3.6 SCENARIOS AS A PROTOTYPING TECHNIQUE

Scenarios can be used as a prototyping technique. All user interface prototypes work by reducing the complexity of the full system in some way, either by eliminating functionality or by eliminating coverage of some parts of the interface. Prototype scenarios eliminate both kinds of complexity by offering a simulation of a single user path through the system. In other words, a scenario is a user interface prototype that can be used to do exactly one thing. As such, they are very cheap to build and are a perfect method for use as part of a discount usability engineering approach [11, 12, 19].

A prototype scenario may be as simple as a textual description of the steps a user would take to achieve the single goal supported by the scenario. This is the kind of scenario produced by the electronic door brainstorm sessions discussed above, though a scenario would need a detailed description of the user interface at each user step to truly count as a prototype. Text-only prototypes rely on the reader's imagination to fill in details of screen design and so forth and only allow an evaluation of the higher-level design issues.

A prototype scenario can also be a set of screen mockups that exist either on paper or as screens displayed by a simple slide-show computer program [10]. Such scenarios can be used for heuristic evaluation [13] and also for some simple user testing. The use of this kind of prototype scenario for user testing is somewhat different than the *scenario machine* described by Carroll and Kay [4]. The scenario machine (often a fully functional interface modified for instructional purposes) blocks the user from performing any actions that are not part of a prespecified scenario in the hope that a novice user will learn faster by being prevented from garden pathing and from spending less time to recover from errors. In a prototype scenario, users will also be unable to complete actions that are not part of the predetermined small action set, but the user is still encouraged to consider all possible actions at each step through the scenario: Instead of just showing the next screen, no matter what the user did (as done by the most extreme version of the scenario machine), the experimenter in a scenario user test will discuss each screen with the user and let the user choose freely what the next action *ought* to be according to that user. Very likely, that next action will not be implemented, and the experimenter will then explain that to the user and encourage further exploration. In this way, a user test of a scenario can collect at least some information about alternative user strategies that do not match the path intended by the designers, even though that path will be the only one to be tested in its entirety.

Common for this form of scenario is that it tries to give as good an impression as possible of the way the user interface will be, even though the interface does not exist yet. See Figure 3.3 for an example of a single screen from one such scenario of the use of a nonexisting graphical user interface for information access. The full scenario contained several such screens and was designed by an iterative design approach, where three early versions were tested with users and modified according to the usability problems discovered. The version in Figure 3.3 is thus the



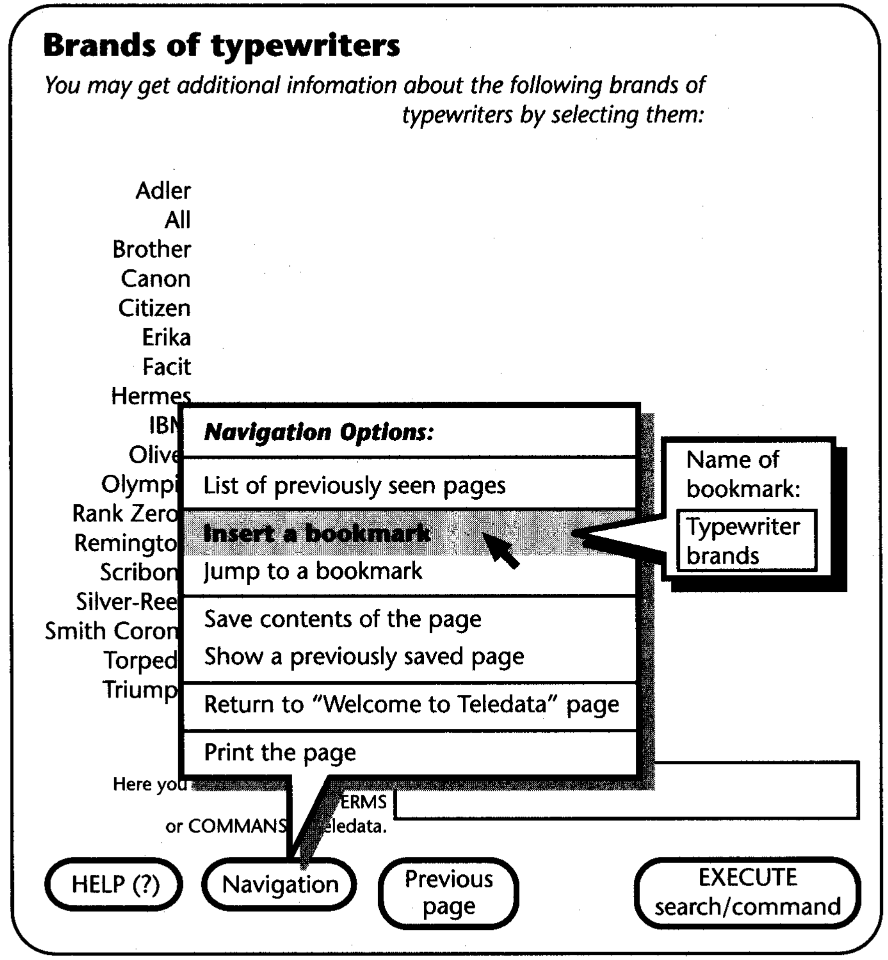


**Figure 3.3.** Example of a screen from a scenario developed to prototype on-line information access using a graphical user interface. The scenario is that the user wants to buy a typewriter. Here, the user has found the listing of the (many) manufacturers of typewriters and may want to return to it later. Therefore the user inserts a bookmark at this screen [10].

fourth version of the scenario. It was possible to produce four versions of the interface in a very short amount of time because the use of scenarios made the prototyping trivial (just draw new screens in a graphics editor—or even easier, modify earlier drawings) and the user testing quick (since only a single interaction stream had to be tested).

Several lessons were learned during the tests of the scenario represented in Figure 3.3. For example, the users had been asked to use the system as if they were interested in buying a typewriter and they often wanted to return to the screen listing all the typewriters. As a result of this observation, the system design was modified to include a bookmark option for users to mark locations in the information space to make them easier to return to. Another observation was that users easily became overwhelmed by the amount of information and wanted a way to filter the information base to show only information of interest to them (for example, only typewriters costing less than a certain amount). Such a filtering mechanism would have been difficult to implement in a general system and might have set back the user testing schedule by several months, but it was easy to try out different options in the scenario where various filtering needs and mechanisms could be presupposed and precomputed.

## 3.7 SCENARIOS TO STRUCTURE USER TESTING

Even in traditional user testing of a more complete system, a form of scenario can be used to improve the test situation. To prevent a user test from being a set of irrelevant and unrelated isolated tasks, it is possible to relate the tasks through a scenario. This form of scenario is a rough outline of a situation in which users might find themselves together with a rationale for why they might want to perform the tasks they are asked to do. By relating the tasks to a larger context, the scenario makes it easier for the users to understand the task descriptions and makes it possible for the user to reuse objects from previous test tasks. Also, having a scenario in mind allows the usability specialist planning the test to construct a coherent and realistic set of test tasks.

For example, we recently performed a series of user tests with alternative user interfaces to a personal communication service. A personal communication service would assign a single telephone number to an individual and then forward calls to that number to different physical telephones, depending on various instructions given by the user. For example, calls could be forwarded to the user's home, car phone, and office phone at different times of day, except that no calls would be



forwarded at night unless the caller knew a special password. The system does not exist in the real world, so we do not know for sure how people are going to use it. To come up with a reasonable set of test tasks, we relied on a scenario centered around a person working in an office on a project with people from another office, and who is going on vacation.

## 3.8 CONCLUSIONS

In earlier work [13], I suggested that scenarios could be classified along three dimensions, purpose, medium of expression, and source of inspiration, giving rise to the taxonomy described in the following and summarized in Table 3.2.

The first dimension of the taxonomy is the purpose of the scenario. The experienced usability professional will want to choose very different kinds of scenarios to achieve different goals. Scenarios can be used as a communication tool, as a thinking or design tool for the scenario developer him- or herself, and they can be used as an artifact in testing (which again can have multiple goals: to test the actual interface design or to test and compare various HCI theories).

The second dimension is the medium of expression and implementation chosen for the scenario. The three main media types are textual

**Table 3.2** The Three Dimensions of the Taxonomy of Scenarios Proposed in this Chapter

- **Purpose**
  - Communicate user interface issues to an audience
    - managers, colleagues
    - users
  - Structure thinking and provide background for refinements
  - Testing
    - interfaces
    - HCI theories
- **Medium of expression and implementation**
  - Textual description
  - Storyboards (screen designs on paper, video, etc.)
  - Running system on an actual computer
- **Source of inspiration**
  - Empirical observations
  - Designers' ideas and analysis

descriptions (leaving the visualization of the scenario to the human imagination), storyboards (leaving the dynamic transitions and behavior to the viewer's imagination), and finally, running systems on actual computer equipment (or whatever implementation platform is intended for the final system). As the scenario progresses along the media type dimension, it is typically necessary to invest more work, but the result will be more precisely specified. Precise specifications are normally an advantage but they can also be a disadvantage early in the design process where one does not want to unduly limit one's imagination.

The final dimension of the taxonomy is the source of inspiration for the elements of the scenario. The two main types of inspiration are empirical observations and the designers' ideas, meaning that scenarios can be built to reflect either the world as it is now or the world as it may become.

Using the taxonomy in Table 3.2 to classify the five types of scenario discussed above gives the following result:

- Diary scenarios: Structure thinking, textual descriptions, empirical observations
- Scenarios in feature brainstorms: Structure thinking, textual description, designers' ideas and analysis
- Scenarios in parallel design: Communicate user interface issues to an audience (colleagues), storyboards, designers' ideas and analysis
- Scenarios in heuristic evaluation: Testing (interfaces), textual description combined with a running system, empirical observations
- Scenarios as a prototyping technique: Testing (interface), any of the three media, designers' ideas and analysis
- Scenarios to structure user testing: Testing (interface), textual description, empirical observation (if derived from field studies or task analysis), or designers' ideas and analysis
- Scenarios as stereotypes to analyze exploratory data: Structure thinking, textual data, empirical observation

The kinds of scenarios discussed here are all tools for use at various stages of the usability engineering lifecycle [15]. By focusing on a single use of an interface, scenarios enable us to understand and communicate the interface better and thus make our work easier and cheaper.

## ACKNOWLEDGMENTS

The electronic door brainstorm project was in collaboration with members of the Bellcore *beyond being there* research project: Steve Abney, David Ackley, Laurence Brothers, George Furnas, Will Hill, Jim Hollan, Joel Remde, Scott Stornetta, and Jeff Zacks. The parallel design project was in collaboration with Heather Desurvire (NYNEX), Bob Durr (Bell Atlantic), Jim Kondziela (NYNEX), Adam Marx (U S WEST), Janet Davidson (Pacific Bell), Randy Kerr (Microsoft), Dan Rosenberg (Oracle), Gitta Salomon (IDEO Product Development), Rolf Molich (Kommunedata), and Tom Stewart (System Concepts Ltd.). The heuristic evaluation of a highly domain-dependent user interface was in collaboration with Gay Norwood from Bellcore's Operations Technology area. The study of personal communication services was in collaboration with Marco Bergman from Delft University of Technology. In all cases, the way these projects have been interpreted in this chapter is solely the responsibility of Jakob Nielsen and does not necessarily represent the positions of the collaborators in the underlying projects.



## REFERENCES

[1] Bikson, T. K. & Eveland, J. D. (1990). The interplay of work group structures and computer support. In J. Galegher, R Kraut. & C. Egido (eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work.* Hillsdale, NJ: Lawrence Erlbaum, pp.245–290.

[2] Brinck, T. & Gomez, L. M. (1992). A collaborative medium for the support of conversational props. In *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work,* Toronto, November 1–4, pp. 171–178.

[3] Brothers, L., Hollan, J., Nielsen, J., Stornetta, S., Abney, S., Furnas, G. & Littman, M. (1992). Supporting informal communication via ephemeral interest groups. In *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work,* Toronto, November 1–4, pp. 84–90.

[4] Carroll, J. M. & Kay, D. S. (1985). Prompting, feedback and error correction in the design of a scenario machine. In *Proceedings of ACM CHI'85 Conference,* San Francisco, CA, April 14–18, pp. 149–153. Also in the *International Journal of Man-Machine Studies, 28,* 1 (1988): 11–27.

[5] Erickson, T. (1995). Notes on design practice: Stories and prototypes as catalysts for communication. *Chapter 2 in the present book.*

[6] Hollan, J. & Stornetta, S. (1992). Beyond being there. In *Proceedings of ACM CHI'92 Conference,* Monterey, CA, May 3–7, pp. 119–125.

[7] Jacobson, I. (1995). The use-case construct in object-oriented software engineering. *Chapter 12 in the present book.*

[8] Kraut, R. E., Egido, C. & Galegher, J. (1990). Patterns of contact and communication in scientific research collaboration. In J. Galegher, R. Kraut & C. Egido (eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work.* Hillsdale, NJ: Lawrence Erlbaum, pp.149–171.

[9] Naur, P. (1983). Program development studies based on diaries. In T. R. G. Green, S. J. Payne & G. C. van der Veer (eds.), *The Psychology of Computer Use.* London: Academic Press, pp. 159–170.

[10] Nielsen, J. (1987). Using scenarios to develop user friendly videotex systems. In *Proceedings of NordDATA'87 Joint Scandinavian Computer Conference,* Trondheim, Norway, June 15–18, pp. 133–138.

[11] Nielsen, J. (1989). Usability engineering at a discount. In G. Salvendy, & M. J. Smith (eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Amsterdam: Elsevier Science, pp. 394–401.

[12] Nielsen, J. (1990a). Big paybacks from "discount" usability engineering. *IEEE Software, 7*, 3 (May): 107–108.

[13] Nielsen, J. (1990b). Paper versus computer implementations as mockup scenarios for heuristic evaluation. In *Proceedings of IFIP INTERACT'90 Third International Conference on Human-Computer Interaction,* Cambridge, U.K., August 27–31, pp. 315–320.

[14] Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proceedings of ACM CHI'92 Conference,* Monterey, CA, May 3–7, pp. 373–380.

[15] Nielsen, J. (1993a). *Usability Engineering*. Boston: Academic Press. Paperback edition, 1994.

[16] Nielsen, J. (1993b). Is usability engineering really worth it? *IEEE Software, 10*, 6 (November): 90–92.

[17] Nielsen, J. (1994a). Heuristic evaluation. In J. Nielsen, & R. L. Mack (eds.), *Usability Inspection Methods*. New York: Wiley, pp. 25–62.

[18] Nielsen, J. (1994b). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. In R. G. Bias & D. J. Mayhew (eds.), *Cost-Justifying Usability*. Boston: Academic Press, pp. 245–272.

[19] Nielsen, J. (1994c). Why GUI panic is good panic. *ACM Interactions Magazine, 1*, 2 (April): 55–58.

[20] Nielsen, J. (1995). *Multimedia and Hypertext: The Internet and Beyond.* Boston: Academic Press.

[21] Nielsen, J. & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of ACM CHI'90,* Seattle, WA, April 1–5, pp. 249–256.

[22] Nielsen, J., Desurvire, H., Kerr, R., Rosenberg, D., Salomon, G., Molich, R. & Stewart, T. (1993). Comparative design review: An exercise in parallel design. In *Proceedings of ACM INTERCHI'93 Conference,* Amsterdam, April 24–29, pp. 414–417.

[23] Polson, P., Lewis, C., Rieman, J. & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies, 36*, 741–773.



[24] Rieman, J. (1993). The diary study: A workplace-oriented research tool to guide laboratory efforts. In *Proceedings of ACM INTERCHI'93 Conference,* Amsterdam, April 24–29, pp. 321–326.

[25] Stephenson, N. (1992). *Snow Crash*. New York: Bantram Spectra.

[26] Tognazzini, B. (1992). *Tog on Interface*. Reading, MA: Addison-Wesley.

[27] Vertelney, L. (1989). Using video to prototype user interfaces. *ACM SIGCHI Bulletin, 21*, 2 (October): 57–61.

[28] Wharton, C., Rieman, J., Lewis, C. & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. L. Mack (eds.), *Usability Inspection Methods*. New York: Wiley, pp. 105–140.