Comparative Design Review: An Exercise in Parallel Design

Organizers: Jakob Nielsen, Bellcore, and Heather Desurvire, NYNEX
Moderator: Jakob Nielsen, Bellcore
Designers: Randy Kerr, Microsoft
Dan Rosenberg, Borland Intl.
Gitta Salomon, Apple Computer
Evaluators: Heather Desurvire, NYNEX
Rolf Molich, Baltica Insurance
Tom Stewart, System Concepts Ltd.

Abstract

Three user interface designers were asked to design interfaces for a given problem. These designs were made available to a group of usability specialists for heuristic evaluation. The reviewers will lead off the panel with specific questions to the designers regarding the usability aspects of their designs. The panel will feature a lively discussion of the designers' various approaches and solutions.

Parallel Design

Jakob Nielsen

The goal of parallel design is to explore different design alternatives before one settles on a single approach that can then be developed in further detail and subjected to more detailed usability activities, including iterative design [1].

Typically, one might have about three designers involved in parallel design. For critical products, large computer companies have been known to devote entire teams to developing multiple alternative designs almost to the finial product stage, before upper management decided on which version to release. In general, though, it may not be necessary for the designers to spend more than a few hours or at the most one or two days on developing their initial designs. Also, it is probably better to have designers work individually rather than in larger teams, since parallel design only aims at rough drafts of basic design ideas, in order to explore various parts of the design space.

Heuristic evaluation is a usability inspection method which is aimed at evaluating user interfaces in a fast, inexpensive, and effective manner [1]. The panel will illustrate this evaluation technique, and will highlight its use as an informative technique for evaluating interfaces at an early stage.

Reference: [1]. Nielsen, J. *Usability Engineering*, Academic Press, San Diego, CA, 1993.

Design Problem Given to the Designers

Heather Desurvire and Jakob Nielsen

The designers' task was to design the screen-based graphical user interface for a potential future system that allows users to play music transmitted from a central database through a broad-bandwidth network. To our knowledge, nobody is currently contemplating such a system, so the following description is purely hypothetical.

This interface can be assumed to be available due to an envisioned fiber link to home residential areas as well as businesses. The interface is a personal computer, that allows the users to select any music they want to listen to. Pricing for listening is per hour of listening, and has been calculated to be about what it would cost to purchase 200 CDs and a CD player over 10 years. The fiber link is connected to a set of stereo speakers, that can be purchased with a built in fiber link connection, or can be connected to a typical set of speakers with an adapter. The interface is via a personal computer, which is screen-based, having a graphical user interface and a mouse.

Assume that the database of music is based on the contents of at least ten thousand compact discs.

Design the user interface so that the user can select:

- one song with various artists
- one artist with various songs
- multiple selections for a certain time interval (e.g., 45 minutes of music)
- random songs in a selected genre (by singer, musician, style of music (e.g. rock, motown, classical guitar, south american))

While listening to the music, the user should also be able to manipulate the selections to:

- Pause
- Fast Forward
- Skip to the next song
- Rewind

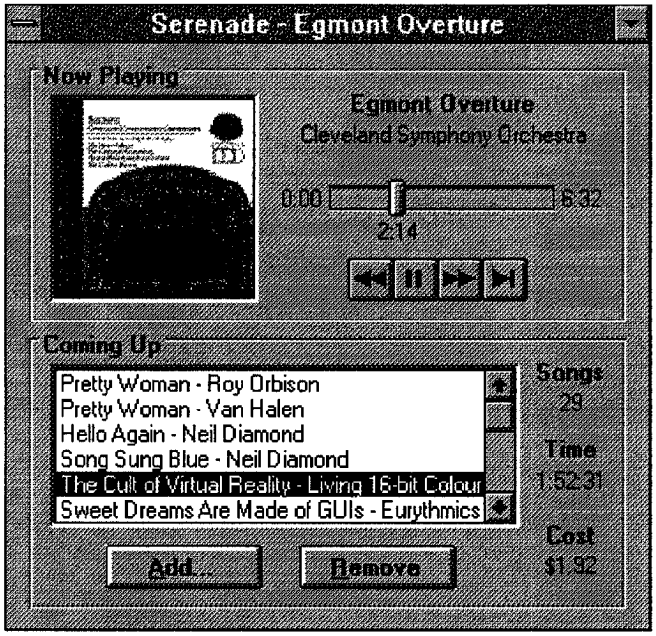
Simplifying assumptions: Only consider music that is structured as "songs." Thus, the interface does not need to accommodate classical music or other music with a more complex structure.

Assume that the user already knows how to use the home computer and its graphical user interface.

Serenade: The Desktop Juke-Box

Randy Kerr

Serenade is a music desktop accessory that operates very similar to a juke-box. The interface factors the user's activity of listening to music primarily as *selecting* and *playing* songs. The main window controls the currently playing song and organizes the upcoming songs. A *Add Songs* sub-dialog is used to filter and select new songs from the database to be added to the play list. This dialog allows very flexible and uniform searching for a specific or random set of songs in a given category. *Serenade* is designed assuming a music service billing rate of $1.00 per hour of listening, and that the music database holds the associated CD cover images.



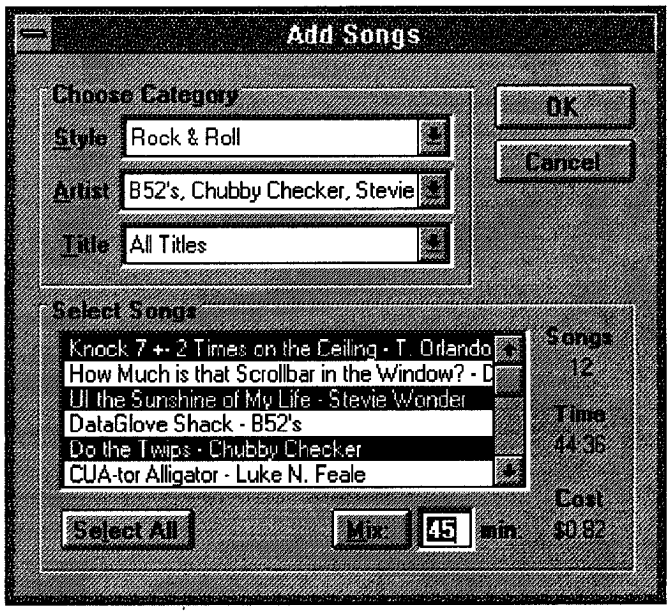
Playing Songs

The main window of *Serenade* posts the current song's name in its title bar and is divided into two cooperating panes labeled *Now Playing* and *Coming Up*. *Now Playing* displays the currently playing song's CD cover (as a memorable visual cue), title, artist and the familiar compact disc controls for rewind, pause/play toggle, fast forward, and skip to next selection. Above these buttons is a slider control which moves from left to right displaying the updating time counter as the song plays. The user may drag the slider to get to a position in the song directly. When the song is finished, the song at the top of the *Coming Up* list assumes the *Now Playing* position and begins to play from its beginning. Songs play without interruption from the rest of the interface: adding new songs or minimizing the *Serenade* window does not interrupt the playing of the song.

The *Coming Up* portion of the window displays the ordered list of songs which will be played next. The user may re-order these songs as (s)he pleases by dragging and dropping songs in different positions in the list with the mouse. The user may remove any song from the list by selecting it and pressing the *Remove* button (disabled when there is no selection) or by hitting the Delete key. The number of songs, the total time, and the cost of the pending music is posted to the right of the *Coming Up* list and are updated dynamically as each song is played.

Selecting Songs

To add songs to the play list, the user presses the *Add...* button which invokes the dialog pictured below.



This dialog allows the user to select songs from as broad or narrow a category as (s)he wishes. Within the *Choose Category* pane the user may type a partial string or select any number of the given items in the drop-down combo boxes labeled *Style*, *Artist*, and *Title*. (The user may also leave the field unspecified by choosing "All".) Songs which match the category specified by these field values appear in the *Select Songs* pane. For instance, the above dialog filters for all rock & roll song titles sung by either the B52's, Chubby Checker, Stevie Wonder, ... (the multiple selection of artists in the drop down list). If the user is unsure of a song's full title, (s)he may type any fragment in the title field and all songs which include that fragment will appear below. To add to the *Coming Up* play list, the user selects (single-clicks) all the desired songs in the *Select Songs* pane and presses the *OK* button. *Select All* will select all songs in the list and *Mix: n min.* will select random songs in the list which amount to the user-specified *n* minutes of time, after which the user may add or remove specific songs from the selection. The total number, duration, and cost of the selected songs appear to the right assisting the time/cost tradeoffs in deciding what to hear.

Interactive Fiber Optic CD System User Interface

Designed by: Daniel Rosenberg

This UI design for a fiber optic home CD music system is characterized by the following architectural features:

1) A traditional desktop UI model is used to blend into the existing windowing environment.

2) Two object types exist, a personal CD object and a Catalog object.

3) Music can be played from either type of object.

4) Their can be as many or as few of each object type open as the user wants.

5) Each object type has two views, a **player view** (that mimics a CD player) and a **song list view**.

6) The catalog object supports query and history lists in its song list view.

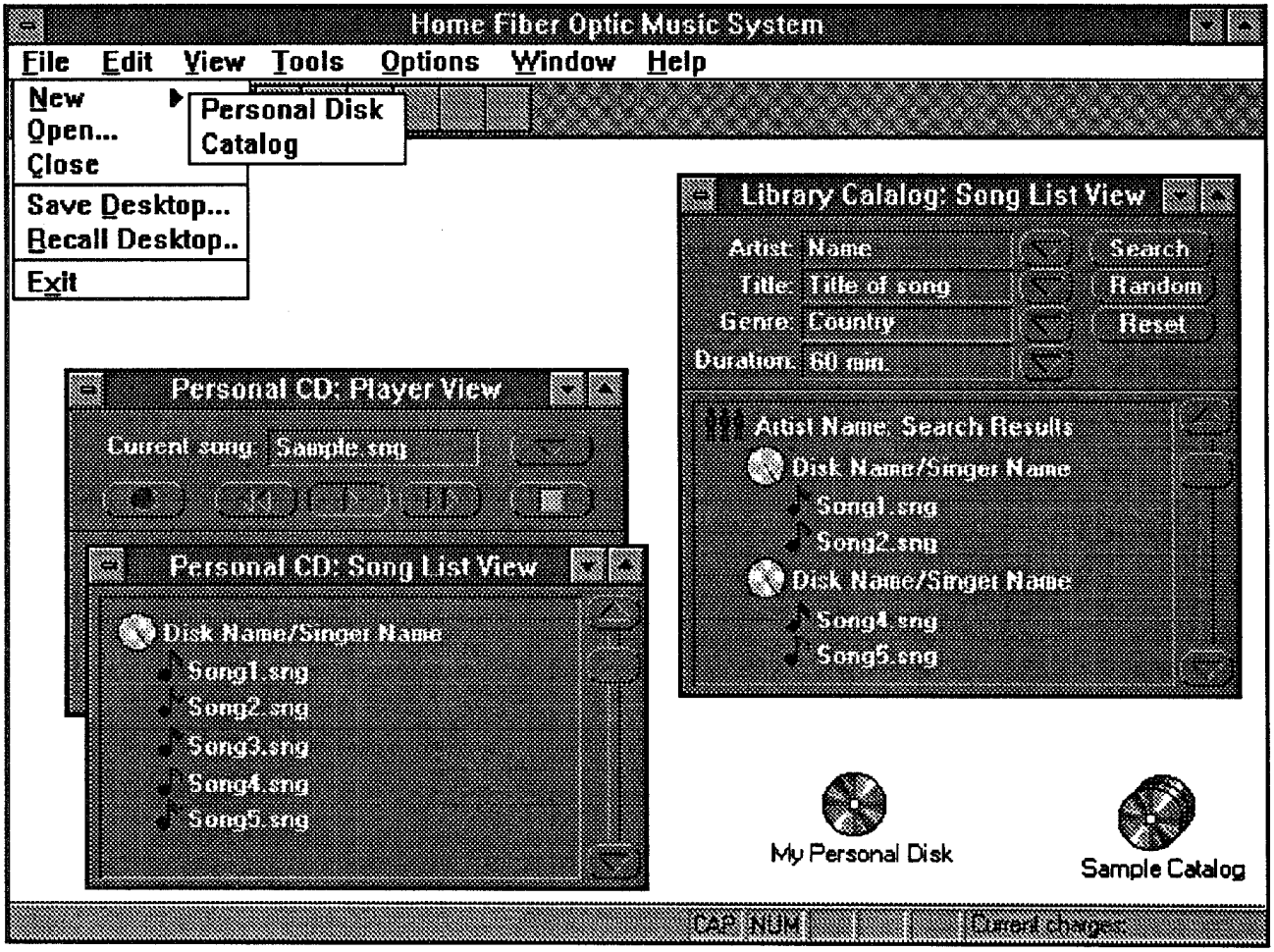
7) Data relationships between author, song and disk title are supported with a built in outliner model. Expand and collapse are supported.

8) A traditional pull down menu structure is provided for editing the disks.

9) Drag-drop semantics are provided between open objects and desktop icons as well. Double-clicking on a song title in either object or view plays the song.

10) The desktop provides a context storage feature so the user can always return to a named desktop state.

The underlying rational for choosing the desktop model was that it can be expanded to other media types. For example video technology could be added to the same interface without overloading the user model. It is also possible to extend this UI with full featured editing capabilities as will be demonstrated paper presentation.



Design of the Personal Jukebox™ Interface

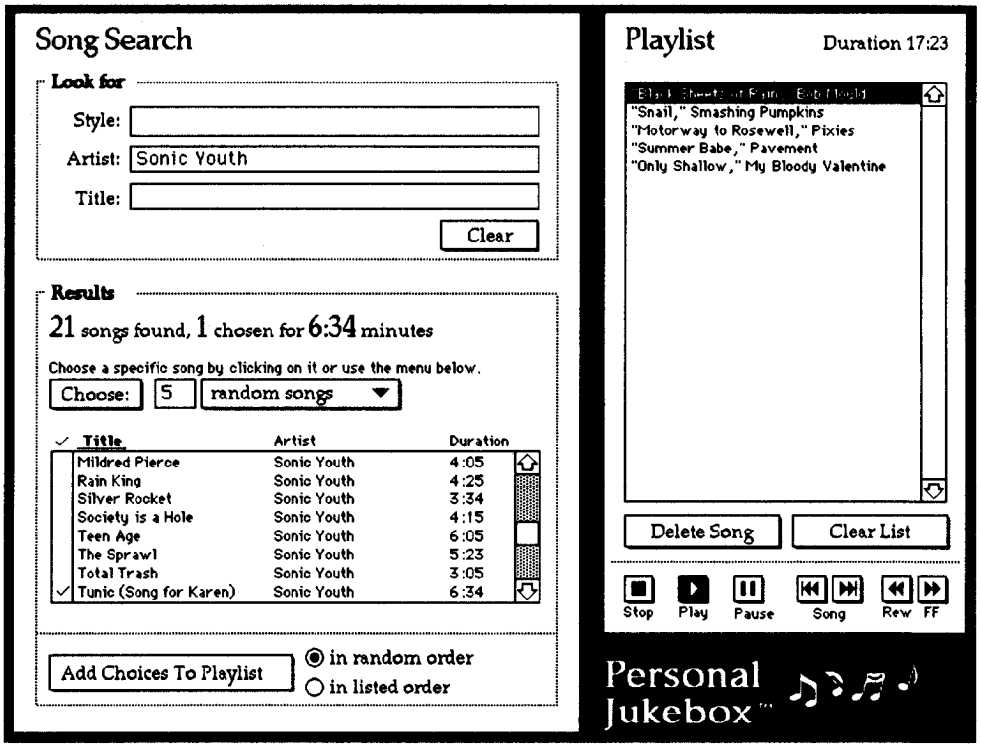
Gitta Salomon

The design shown below (see captions for details) provides users with access to the required functionality in a flexible way. The user can search for songs using several criteria, in both a goal-directed and browsing fashion. Although not shown below, when the user clicks in one of the "look for" fields, a scrolling list of possibilities appears. As the user enters text, this list scrolls to the appropriate place. In this way, the user can enter either an explicit or partial request, or just browse through the possibilities and select one. The interface also provides the user with a number of ways to assemble a playlist. Furthermore, because the search and playlist panels are distinct, the user can look for new material while simultaneously playing audio.

On the role of interface designers. The design problem, as presented to the panelists, already included a specification of the system's functionality. It is my belief that interface designers should be involved much earlier in the product definition process, when the functionality requirements themselves are being determined. Our job should encompass much more than only representing the functionality to users. In my work, we initially conduct informal interviews with potential end users to determine their problems, needs and expectations in a particular domain. Through this user-centered approach, we lead the way in defining what technology should be built. This often produces more desirable and usable technology.

To better make this point, I conducted a few brief interviews concerning the song delivery system and uncovered several interesting ideas for additional functionality. First, I spoke with people about how they search for songs in record stores and how they'd optimally like to find music. Some users want to be able to find songs based on their lyrics when they can't recall the artist's or song's name. If possible, this search criteria should be supported. Another user suggested a 'time machine,' whereby she could queque up an hour of music 'just like the local hit radio station played during February, 1963' (this is not so farfetched – radio stations often maintain playlists, some make them publically available). This is a compelling alternative to the "Song Search" method shown below, and is surely only one of many alternative methods for finding music. I also interviewed a few radio DJs about how listeners request songs and how they decide what to play when they receive incomplete information. DJs often rely on their colleagues' annotations or markers indicating the best, or most requested, songs on an album. If the Personal Jukebox incorporated a community's previous request data, it would allow users to select an artist's most popular songs, even if they couldn't remember or recognize a specific title. Although limited, these interviews serve to point out the role that interface designers can play at early stages in the design process.

*The **Song Search Panel** allows users to look for songs using 3 criteria: the style of music, the artist's name (here, "Sonic Youth") and the song's title. In this session the system returned 21 matches in the scrolling results list. The user has several options for choosing songs from this list: individual songs can be picked by clicking on them ("Tunic" is currently chosen); alternatively, the pulldown menu can be used to choose several songs at once. As shown, the user can click on the "Choose:" button to randomly pick 5 songs. Another menu option allows the user to choose a specific number of minutes of music from the list. Note that the user can reorder the results by clicking on the heading words (The current ordering is alphabetical by title, as indicated by the bold, underlined appearance of the word 'Title'). Users can move their choices to the playlist at any time by clicking on the "Add Choices To Playlist" button. This button either transfers the songs in the order they are shown, or in a random order, depending on the radio button modifier selected.*



*The **Playlist Panel** contains a scrolling list of the songs the user has chosen to play. The list can be altered in several ways: it can be cleared with "Clear List" button; individual songs can be deleted by selecting them and choosing the "Delete Song" button; the order of songs can be changed by clicking on an entry and dragging it to a new position. There is a visual representation for the 'current' song – it is highlighted with inverse video – that cues the user as to which song will be acted on by requests such as 'play', 'pause', 'scan forward (ff),' etc. Buttons that visually resemble those found on standard compact disk players are provided at the bottom of the panel to allow for intuitive control of the various playing states. The current state is shown in inverse video.*