

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TORONTO

CSC318S

**THE DESIGN OF
INTERACTIVE COMPUTATIONAL MEDIA**

Lecture 7 — 2 February 1998

**MULTIDISCIPLINARY DESIGN, ENVISIONMENT,
AND PROTOTYPING**

7.1 Ideas for new media and their interfaces.....	2
7.2 Multidisciplinary design.....	3
7.3 “Task analysis” of the future — Envisionment.....	4
7.4 Envisionment tools for visualization.....	5
7.5 Card-based prototyping tool — HyperCard.....	6
7.6 Time-based prototyping tool — Director	6

Ronald Baecker
Professor of Computer Science,
Electrical and Computer Engineering, and Management
University of Toronto

Copyright © 1991-1995, 1998, Ronald Baecker.
All rights reserved.

7.1 Ideas for new media and their interfaces

Where do ideas come from? They come from.....

- Observations of current work practice

- Observations of current systems

- Insights from other media and disciplines (7.2), e.g.,

 - Film

 - Animation

 - Theatre

 - Architecture

 - Industrial design

 - Information display

- Imagination and envisionment (7.3, 7.4)

 - Scenarios, prototypes

 - Prototyping media and tools (7.5, 7.6)

- From analogies and metaphors (Lecture 8)

Techniques for generating new ideas (Mountford, BGBG, 138-9)

- New uses for the object (e.g., “the desktop”)

- Adapt the object to be like something else

 - (e.g., a playroom, a kitchen)

- Modify the object for a new purpose

 - (e.g., exploration, access to environment)

- Add features to the object (e.g., other tools)

- Subtract features from the object

 - (i.e., leaving bare essentials only)

- Substitute a similar object (e.g., playing field)

- Rearrange aspects of the object (e.g., top to bottom)

- Change point of view

 - (e.g., windows view *from files to desktop*)

- Combine the data into an ensemble (e.g., virtual office)

7.2 Multidisciplinary design

Why multidisciplinary design?

- Need for understanding of users and tasks
- Need for communication with users
- Need for envisionment of design possibilities
- Need for playing with design ideas (7.1)
- Need for creation of rich, sensual, interactive media
- Need for evaluation of system and interface success

These needs require many skills and perspectives not normally possessed by most computer scientists

Disciplines contributing to user interface design

- Computer science
- Psychology
- Sociology, organizational behaviour
- Anthropology, linguistics
- Graphic design
 - See Vertelney, Arent, & Lieberman, (Laurel, pp. 45-55)
- Industrial design
 - See Vertelney & Booker, (Laurel, pp. 47-63)
- Task specialists, “users”

Need for clash, synergy of perspectives and priorities

- Example: Computer science and graphic design
 - Computer scientists value program and how it works
 - Graphic designers value picture, screen, interaction, and how it looks and feels
- See Kim article, in Laurel book, pp. 31-44

Designers and users can help with envisionment (7.3, 7.4)

Key aspect is creating prototypes (7.5, 7.6), so prototyping media must be accessible to non-computer specialists

7.3 “Task analysis” of the future — Envisionment

Envisionment is the production of scenarios or other artifacts such as scripts, storyboards, or interactive prototypes for interface visualization

Envisionment for designing interfaces

- Visualizing concepts

- Exploring alternatives

- Resolving feature details

- Developing interaction scenarios (e.g., “Day in the Life”)

Envisionment for pre-testing interfaces

- Can you read or interpret this?

- Can you follow this?

- Can you make this work?

- Do you understand what is going on?

- Is this the way you would do this?

- Does this suggest alternate approaches to you?

Envisionment for presenting interface ideas

- To interface designers, for discussion

- To programmers, to guide implementation

- To marketing and management, to guide product design

- To users, to get early feedback

Another term is “smoke and mirrors” !

- But don’t let yourself be fooled

7.4 Envisionment tools for visualization

Scripts, scenarios... “A Day in the Life”

Role-playing exercises

Drawings, sketches

Screen shots

Storyboards

Flipbooks

Puppets

Collections of post-its and cut-outs

Physical models

Computer animation

Interactive software prototypes (e.g., with HyperCard, as in 7.5; Director, as in 7.6)

We have already seen examples associated with the Apple and Interval Design Competitions

7.5 Card-based prototyping tool — HyperCard

Features of HyperCard (Example in Fig. 7.1)

- A system accessible to non-programmers
- An interface builder
- A direct manipulation system
- A hypertext system
- A prototyping tool
- An extensible system
- An object-oriented system

Some limitations to HyperCard

- Interface limitations
 - Best for card-based interfaces
 - Not good for interactive text manipulation
 - Not good for sketching, gesture-based interfaces
- HyperTalk language limitations
 - No arrays
 - Little (weird) program structure
 - Few debugging tools

7.6 Time-based prototyping tool — Director

Features of Director

- System for computer animation, structuring images and their changes over time
- Theatre as the unifying metaphor — actors, stage, etc.
- Scripting language — Lingo
- An extensible, object-oriented system

Director can be used for prototyping, but.....

- Very low-level specification of interfaces
- Lingo language not much better than HyperTalk

Figure 7.1 A Telephone Logbook in HyperCard (Goodman, The Complete HyperCard Handbook, 3rd Edition, Bantam, 1990, pp. 801-5)

A Telephone Logbook

MANY EXECUTIVES AND SELF-EMPLOYED PROFESSIONALS NEED TO RECORD THEIR OUTGOING telephone calls—the date and time, the person called, the phone number called, the content of the call, and to what account or project the time and phone call charges should be billed. What we'll show you here is the beginning of what could become an elaborate system. It ties directly into the Addresses stack supplied with HyperCard.

Overview

Whenever you are in the Addresses stack (Figure 57-1) and dial a call, a modified dial button script not only dials the number as it always did but also goes to a different stack of telephone logbook pages (Figure 57-2) and generates a new card. When a new card is made, four items in the card are automatically filled in for you: the current date, the time, the name of the person appearing in the Addresses stack card you just dialed, and the phone number just dialed.

You may also generate a new card while in the Telephone Log stack by choosing New Card from the Edit menu. When you do this, the date and time are placed into the card. Because the card doesn't know whom you're calling, you'll have to type in the name and phone number.

At any time while you are in the Log stack, you may check the person's Addresses stack card by clicking on the cards button in the upper right. That button's script goes to the Addresses stack and performs a search on the name from the Log card. If no match is found, you are given the choice to make a new Addresses stack card for this person or not. If you choose yes, a new Addresses card is made, placing the person's name in the Name spot and the phone number into the Phone Number field on the Addresses card.

When you are finished with a call, you may click the Done button near the field labeled Call Finished at. This action places the current time into the adjacent field. If there is already an ending time in that field, a click of that button won't accidentally overwrite the time.

Addresses	
Name	Fred Yoshioka
Company	
Street	5330 W. 87th Street
City & State	Pacific, WA
Zip Code	98092
Telephone	555-6741

Figure 57-1 We modify the script connected to the Dial buttons to log of each dialed call.

Telephone Log	
Date:	Friday, May 22, 1987
Call Started at:	7:14 PM
Call Finished at:	7:20 PM
Who I Called:	Fred Yoshioka
Phone Number Called:	555-6741
Comments:	Has a great HyperCard stack he thinks I should see.
Client Charge:	Yoshioka Intl.

Figure 57-2 A record of each call goes into the Telephone Log card and the fields are automatically filled in by the script.

In practice, a professional person would perform a Message Box File a project or client name when it comes time for billing. Also, if you telephone bill and don't recognize a number, search for that number to find the phone call that generated the call.

Scripts

We start with the dial button script in the Addresses stack, as revised for use with the Log stack. No other changes were made to the button or the rest of the Addresses stack. From there we move onto the scripts and properties of the Log stack. Refer to Figure 57-2 to get your bearings when we describe fields and buttons.

```
stack "Addresses"
background "Body"
  on dialNumber phoneNumber
    [add the following lines to the end of this handler]
    put field "Name" into personCalled
    lock screen
    go stack "Telephone Log" in a new window
    makeNewCard -- handler in Log stack
    unlock screen
    put personCalled into field "Person"
    put phoneNumber into field "Number"
  end dialNumber
```

Comment: After dialing the number, this handler places the name and number into their respective variables (personCalled and numberCalled). Then we go to the last card of the Telephone Log stack to keep the new card in chronological order. When the new card is created, the Telephone Log stack's own newCard handler performs some things (you'll see in a moment). Then the name and number are posted to the new log page. Unlocking the screen is optional, because it automatically reverts to false at idle time, but the user will see the name and number being placed into the fields.

```
stack "Telephone Log"
bkgrnd "Log Sheet"
  on makeNewCard
    go to last card of bkgrnd "Log Sheet" -- keep in chrono ~
    order
    doMenu "New Card"
  end makeNewCard

  on newCard -- time and date stamp triggered by makeNewCard
    put the long date into field "Date"
    put the time into field "Start"
  end newCard

  on openStack
    hide msg
  end openStack

  on openCard
```

```
hide msg -- in case you use Message 1
end openCard
```

```
button "First"
  on mouseUp
    visual effect scroll right
    go to first card
  end mouseUp
```

```
button "Prev"
  on mouseUp
    visual effect scroll right
    go to prev card
  end mouseUp
```

```
button "Next"
  on mouseUp
    visual effect scroll left
    go to next card
  end mouseUp
```

```
button "Last"
  on mouseUp
    visual effect scroll left
    go to last card
  end mouseUp
```

```
button "To Home"
  on mouseUp
    visual effect iris close
    go home
  end mouseUp
```

```
button "To Addresses"
  on mouseUp
    put field "Person" into personCalled
    put field "Number" into numberCalled
    get long id of this card
    put it into saveCard -- in case we
    set lockScreen to true
    go to stack "Addresses"
    send "find whole personCalled" to R
    if the result is not empty then --
      answer "No card found. Add one?"
      with "Yes" or "No" or "Cancel"
      if it is "Cancel" then go to sav
```

```
-- glad we saved it
else if it is "Yes" then
  go to last card of background 1
  doMenu "New Card" -- new Address card
  put personCalled into field "Name"
  put numberCalled into field "Telephone"
end if
end if
and mouseUp
```

Comment: This handler must send the Find command to HyperCard to bypass the Find handler already in the Addresses stack. If we let that handler take care of the Find command, this handler wouldn't know if the search was successful nor would it know when to add a new card.

```
button "Done"
  on mouseUp
    if field "End" is empty
      then put the time into field "End"
    end mouseUp
```

Further Ideas

As you use this stack, you'll find many things you can add that will make its application more carefree. For instance, you'll quickly discover that you leave frequent notes in the Comments field about getting a busy signal or no answer. You might want to add buttons that automatically place those notes into the Comments field with the click of the mouse button.

Another button you can add is one that redials the number in the current card. The handler for this will generate a new card and carry over the name and number items into the new card's fields. You can always add buttons that link to other stacks you need while on the phone, like your appointment book.

If you have other stacks that dial phone numbers, consider modifying those dial scripts to come to the Telephone Log stack and make a new card. Eventually, you can build a highly integrated empire of telephone-related stacks.