



Demonstration-based Modeling of Driver Performance with In-Vehicle Telematics Tasks

User Guide

December 8, 2003

Copyright © 2003

**Bonnie E. John, Konstantine Prevas,
Peter Centgraf and Sandra Esch**

All rights reserved.

:

Table of Contents

1. Introduction	3
1.1 The tools and how they work together	3
1.1.1 Dreamweaver MX 2004 - Mock-up Creation	3
1.1.2 Netscape 7.0 - Task Demonstration	4
1.1.3 Behavior Recorder - Recording and Playback of Task	4
1.1.4 Model Launcher - Execute model	4
2. Creating a Mock-up	5
2.1 Introduction	5
2.2 Current limitations of HTML mock-ups	6
2.3 How to construct a Mock-up	6
2.3.1 Edit the "FirstPage.html" Template	6
2.3.2 Creating subsequent pages of the mock-up	9
2.4 Possible trouble spots	13
2.5 Example mock-up	14
2.6 Guidelines	16
3. Create a model by demonstration	17
3.1 Introduction	17
3.2 How to demonstrate a task on a mock-up	17
3.3 Exporting demo file - Creating a task model file	19
3.4 Export file options	20
3.5 The model file	20
3.6 Editing the model file	23
3.6.1 Exploring the effects of monitoring driving	23
4. Executing a Stand-alone Task Model	25
4.1 Introduction	25
4.2 Preparing to Execute the Model	25
4.3 Execute the Task Model	29
5. Running a driving plus task model	33
5.1 Introduction	33
5.2 Preparing to Execute the Model	33
5.3 Execute the Task Model	34

:

1. Introduction

This manual describes the usage of the Performance Modeling by Demonstration System (PMDS). The components in this system allow user interface (UI) designers to generate predictive cognitive models of user behavior simply by demonstrating tasks on HTML mock-ups of interfaces. These models can be executed by the theory-backed ACT-R cognitive architecture, producing quantitative predictions of user behavior.

This User Guide will cover:

- How to design and construct an HTML mock-up using specially designed Dreamweaver tools.
- How to demonstrate a task on a mock-up to produce a task model.
- How to execute the model to generate predictions

1.1 The tools and how they work together

Four software applications are used by the PMDS: Dreamweaver, Netscape, Behavior Recorder, and Model Launcher. First, a special extension installed in Dreamweaver is used to create a storyboard-like mock-up of an interface. Tasks are demonstrated by interacting with the mock-ups in Netscape. Behavior Recorder records the actions performed on the mock-up and exports a task model containing a description of these actions. Finally, the Model Launcher executes the model, interacting either with Netscape or a driving simulation to produce human performance measures.

1.1.1 Dreamweaver MX 2004 - Mock-up Creation



Dreamweaver was chosen as the tool to build mock-ups because it provides a WYSIWYG environment familiar to designers, and because its extension mechanism allowed quick adaptation for this purpose. Special extensions are necessary in order for Dreamweaver to create HTML pages that can communicate actions to the Behavior Recorder. These extensions allow HTML elements such as buttons, text fields, and image hotspots to represent interface elements such as a buttons, menus, and voice commands in a proposed design.

For complete instructions on using Dreamweaver to create a mock-up, see the chapter *Creating a Mock-Up*.

1.1.2 Netscape 7.0 - Task Demonstration



The Netscape web browser allows interaction between Netscape and Behavior Recorder through LiveConnect. This connection allows Behavior Recorder to record and replay actions performed within Netscape.

Netscape 7.0 opens the Mac OS 9 Classic Environment and must have an active network connection in order to operate. The network connection may be vestigial merely connecting to a hub is sufficient communication between Netscape and Behavior Recorder to function.

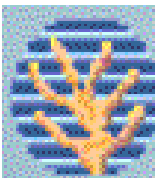
1.1.3 Behavior Recorder - Recording and Playback of Task



The Behavior Recorder records the actions performed on the mock-up in Netscape. The mock-up describes its contents to the Behavior Recorder, which interprets the actions to build the task model. A task model file containing all of the recorded actions is exported from Behavior Recorder at the end of the demonstration.

For complete instructions on how to demonstrate a task using a mock-up, see the chapter *Creating a Task Model by Demonstration*

1.1.4 Model Launcher - Execute model



To produce predictions, the task model file must be executed by the Model Launcher program. Model Launcher uses the ACT-R cognitive architecture to simulate the demonstrated actions and calculate behavioral measures. Predictions can be generated for two different contexts:

- 1) Expert performance of the demonstrated task in isolation. This method uses the Behavior Recorder and Netscape to simulate the task and produce a total time prediction.
- 2) Performance of the task in a driving context. This method uses a model of driver behavior and a driving simulation to produce more varied behavioral measures, such as task time, glance duration, and magnitude of lane deviations.

For complete instructions on how to execute a task model, see the chapters *Executing a Stand-alone Task Model* and *Executing a Driving Plus Task Model*.

2. Creating a Mock-up

2.1 Introduction

An HTML mock-up is a series of web pages that simulates the user interface of a device or piece of software. The goal of a successful mock-up is to correctly simulate the series of actions performed by a user to complete a task. These actions include such things as pressing buttons, looking at display areas, responding to audio stimuli, and making a verbal response.

The primary user input to the PMDS is in the form of a simple mock-up of a proposed interface. A custom-developed Dreamweaver extension is used to create an HTML-based simulation of the graphical software interface or physical device. Although interactions with these two types of interfaces have different physical and cognitive properties, our system can produce valid models for both types by adjusting the instructions provided to the cognitive architecture. For example, a mock-up could simulate a Calculator application on a desktop computer, which uses keystrokes and mouse clicks to register input. The same mock-up could also be used to simulate a handheld calculator, where input would require pressing buttons with a finger. The demonstration of tasks for both devices would involve clicking simulated buttons in the mock-up, but by varying parameters of the output, the performance model produced would be made appropriate to the desired context. For the physical device, the button clicks would be translated into simulated hand movements and given the appropriate motor timings for pressing buttons with fingers. The model for the computer calculator would translate the demonstration into simulated mouse movements and clicks and produce the appropriate motor timings for that method.

In addition to standard graphical interface widgets, special mechanisms are provided for simulating speech input and output using text boxes, for specifying shifts of attention to a driving task, and for simulating a glance at a particular area of an interface. These tools enable the mocking up of devices that have sound capabilities, such as cell phones and talking navigational systems, and for exploring the implications of various interaction styles on driving performance.

2.2 Current limitations of HTML mock-ups

Since the mock-ups are HTML pages, some interfaces may be difficult or impossible to simulate at the level of detail necessary. In particular, interfaces that have highly interactive direct manipulation behaviors, interfaces which rely on animation or precise timing, and interfaces which would require complex Javascript to simulate in an HTML page will not work well. However, it is possible to simulate a wide variety of software and physical devices using images, buttons, menus, and speech.

2.3 How to construct a Mock-up

Mock-ups consist of at least two pages, including a required first page (based on a template called `FirstPage.html`) and at least one page of storyboard for the task.



All mock-ups must include a copy of "FirstPage.html" as the 1st page of the demonstration. This page contains special code needed to accurately simulate the interface. Edit this template only as described below.

The Mock-up Template folder contains a template of "FirstPage.html" along with other support code necessary for a mock-up such as the `BehaviorRecorderApplet.class`.



The file "BehaviorRecorderApplet.class" must be in the same folder as the mock-up.

The location of the Mock-up Template folder in a standard installation is *Macintosh HD>Applications>Behavior Recorder>Mock-up Template*

2.3.1 Edit the "FirstPage.html" Template

- 1) Copy the contents of the Mock-up Template folder into your mock-up folder. BE SURE TO HAVE COPIES OF BOTH "FirstPage.html" and *BehaviorRecorderApplet.class* IN THE FOLDER!
- 2) Open the new copy of "FirstPage.html" in Dreamweaver.



Double-clicking "FirstPage.html" will load it in a browser.

To open it in Dreamweaver, drag it to the Dreamweaver icon on the Dock, or use the "File>Open..." command within Dreamweaver.

FirstPage.html will need to be modified for each storyboard. The picture below shows what the page looks like and where modifications need to be made.



Do not remove or alter the special code in the page, represented in the Design view by the coffee cup icon.

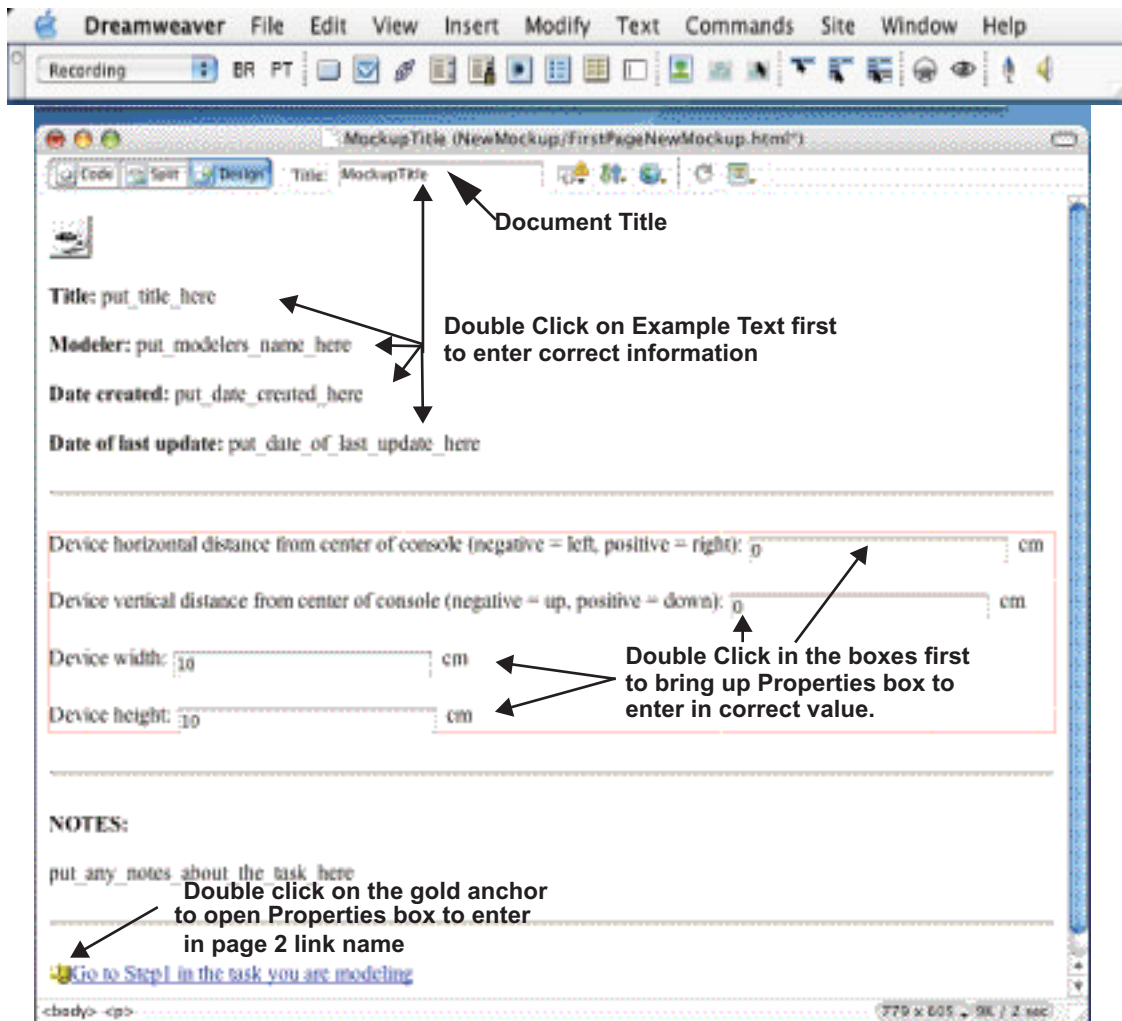


Figure 1 Modifications necessary on FirstPage.html template

3) Modify the first five fields of the template by double-clicking on the example text next to the field's name, and then entering in the correct information.

- **Document Title:** The name that will appear in the Title state of the model file.
- **Title:** The mock-up's name.
- **Modeler:** The name of who constructed the pages.
- **Date created:** Mock-up creation date
- **Date of last update:** Last modification to mock-up

4) Modify the next four fields by double-clicking in the value box next to the text field. This will bring up its Properties window. In the Properties window type the correct value (in centimeters) in the field marked "init val".

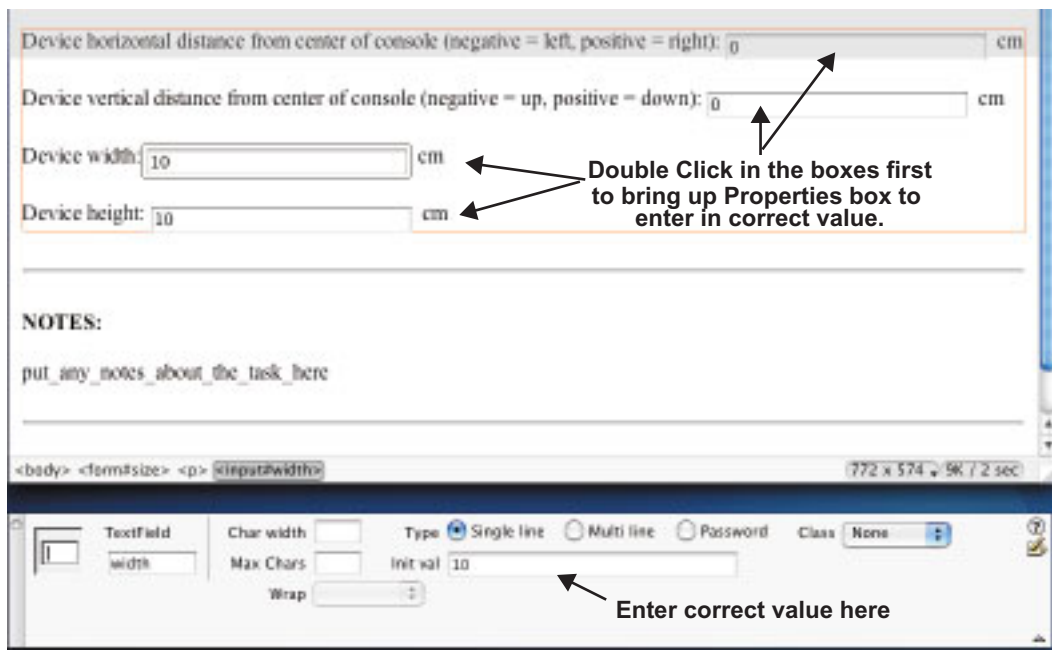


Figure 2 Device dimension information fields.

- **Device horizontal distance from center of console (negative=left, positive=right):** - Enter in the actual physical distance horizontally from the center of the console to the center of the physical device in centimeters.
- **Device vertical distance from center of console (negative=up, positive=down):** - Enter in the actual physical dis-

tance vertically from the center of the console to the center of the physical device in centimeters.

- **Device width:** - Enter in the actual physical width of the device in centimeters.
- **Device height:** - Enter in the actual physical height of the device in centimeters.



If multiple images are being used on a page to construct the image of the device, use the dimensions of the extreme perimeter (including buttons).

- 5) Modify the last field, **NOTES**, if you want. Modification of this field is optional. The contents do not appear anywhere else.
- 6) Double-click on the gold anchor next to the text "Go Step1 in the task you are modeling".
- 7) In the "Link" field in the Properties window, enter the filename of the 2nd page of your storyboard, the 1st image page of the device.



Figure 3 Change link name

- 8) After all fields in the "FirstPage.html" have been modified, save the file in the folder you created in step 1.

2.3.2 Creating subsequent pages of the mock-up

Mock-up pages are created like regular Dreamweaver files except that certain Javascript code must be put in the page at the beginning. This is accomplished by pressing the "BR" button, the leftmost button on the toolbar. An icon will appear on the page that looks like a cup of coffee. This icon represents a Java applet which receives messages from the Behavior Recorder.



Do not alter the Javascript code or remove the applet. If you do so by accident, press the BR button again.

The other buttons on the Recording Toolbar represent interface elements which talk to the Behavior Recorder during the task demonstration.

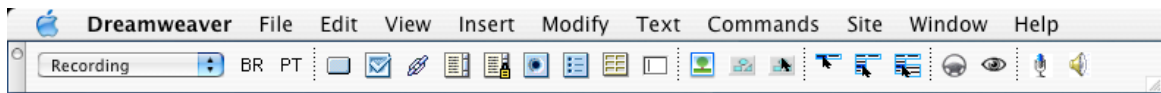























Figure 4 Recording toolbar

Recording toolbar buttons

	Behavior Recorder Header - This button inserts a header which is required on each page of a mock-up. The header will appear on the page as an icon of a coffee cup. If the header has already been inserted, this button will ensure that it has not been modified or deleted.
	Pseudo Tutor Header - This button is not used for predictive performance modelling. If you press this button by accident, press the BR button again.
	Button - When you click on this tool it inserts a button at the insertion point, or immediately following the current selection. Use this tool to simulate a button in your interface.
	Checkbox - When you click on this tool it inserts a checkbox at the insertion point, or immediately following the current selection. Use this tool to simulate a checkbox in your interface.
	Link - When you click on this tool it inserts a hyper link at the insertion point, or immediately following the current selection. This brings up a dialog box that asks the Link name, Link location and Link Text. Use this tool to designate text that will bring up another page of the mock-up when clicked.
	List/Menu - When you click on this tool it inserts a List/Menu at the insertion point, or immediately following the current selection. Use this tool to simulate a drop-down menu in your interface.

	<p>Locked List/Menu - When you click on this tool it inserts a Locked List/Menu at the insertion point, or immediately following the current selection. This brings up a dialog box that asks Please enter a name for this menu. Use this tool to simulate a drop-down menu in your interface which is only enabled after the user clicks on another element of the interface. The element which unlocks the drop-down menu must be selected when this button is clicked.</p>
	<p>Radio Button - When you click on this tool it inserts a radio button at the insertion point, or immediately following the current selection. Use this tool to simulate a radio button in your interface.</p>
	<p>Radio Group - When you click on this tool it inserts a group of radio buttons at the insertion point, or immediately following the current selection. This brings up a dialog box that asks for a group name, the number of buttons desired and whether the buttons should be laid out using line breaks or a table. Use this tool to simulate a group of radio buttons in your interface.</p>
	<p>Table - When you click on this tool it inserts a table at the insertion point, or immediately following the current selection. This brings up a dialog box that asks for a name for the table and the number of rows and columns. Each cell in the table will contain a text field. Use this tool to create a table into which the user can enter text.</p>
	<p>Text Field - When you click on this tool it inserts a text field at the insertion point, or immediately following the current selection. Use this tool to simulate a text field in your interface.</p>
	<p>Image - When you click on this tool it inserts an image at the insertion point, or immediately following the current selection. The image inserted does not communicate with the Behavior Recorder. This tool is identical to the Image tool in Dreamweaver's Common toolbar, and is included here for convenience.</p>
	<p>Hotspot - When you click on this tool it inserts a hotspot over the image currently selected. This brings up a dialog box that asks for a name for the hotspot. Use this tool to designate a section of an image in your interface which can be clicked on by the user. The image, or another hotspot within the image, must be selected before the tool is clicked.</p>
	<p>Rollover Hotspot -When you click on this tool it inserts a rollover hotspot over the image currently selected. This brings up a dialog box that asks for a name for the hotspot. Use this tool to designate a section of an image in your interface which will respond when the user moves the mouse cursor over it. The image, or another hotspot within the image, must be selected before the tool is clicked.</p>

	Menu Name - When you click on this tool it inserts a special hotspot over the image currently selected. This brings up a dialog box that asks for a name. Use this tool to simulate a menu in your interface by creating a special hotspot on an image of a menu system. The image, or another hotspot within the image, must be selected before the tool is clicked.
	Menu Item - When you click on this tool it inserts a special hotspot over the image currently selected. This brings up a dialog box that asks for a name. Use this tool to simulate a menu item in your interface by creating a special hotspot on an image of a menu system. The image, or another hotspot within the image, must be selected before the tool is clicked.
	Submenu Name - When you click on this tool it inserts a special hotspot over the image currently selected. This brings up a dialog box that asks for a name. Use this tool to simulate a submenu name in your interface by creating a special hotspot on an image of a menu system. The image, or another hotspot within the image, must be selected before the tool is clicked.
	Monitor Driving - When you click on this tool it inserts a special button at the insertion point, or immediately following the current selection. The modeler can use this button to insert monitor driving time into the model. Do not change the name of the inserted button.
	Look At - When you click on this tool it inserts a special button at the insertion point, or immediately following the current selection. The modeler can use this button to specify when a user would simply look at an interface element without clicking on it or interacting with it. Do not change the name of the inserted button.
	Audio Input - This is a special text field that represents the user talking to the device (e.g., saying Call Mom to a cell phone). During the task demonstration, the modeler types the words the user would say, and this widget translates the action into an operator in the resulting cognitive model that speaks these words rather than typing them.
	Audio Output - This is a special text field that represents the device talking to the user (e.g., a navigation system saying In 400 feet, turn right to a user). During the task demonstration, the modeler types the words the device would say, and this widget translates the action into an operator in the resulting cognitive model that hears these words rather than typing them.

- 1) Insert your images and widgets from the Recording toolbar on the page. Create the page to look like the device interface in its initial state.



Do not put special characters or spaces in the name of any interface elements, or use a number as the name of an interface element. There must be at least one alphabetic character in the

name. In addition, each interface element must have a unique name.

- 2) Apply your hotspots to sections of the images that will be used to record actions to Behavior Recorder besides those recorded from the buttons inserted in the step above



When hotspots are applied they first appear as small blue rectangles in the upper left corner of the image. You will need to move the hotspot and resize it.

- 3) Add hyperlink addresses to any hotspots that will link to subsequent pages in the storyboard. Double-click on the hotspot to bring up its Properties window, then insert the filename of the page to be linked to in the "Link" field. If a hotspot does not link to another page, the "Link" field should contain a # character.
- 4) Insert the "Monitor Driving" and "Look At" buttons if desired. These buttons should be placed above or below the interface.
- "
- 5) Save the file in the same folder with your modified "FirstPage.html and copy of "BehaviorRecorderApplet.class".

2.4 Possible trouble spots

When an image is inserted, Dreamweaver will sometimes indicate the position of the image in your file system. This may cause the image not to show up in Netscape when the task is demonstrated. If the "Src" field in the image's Properties window contains a full pathname, or anything other than just the image's filename, modify it to contain just the filename. Any images inserted in the page must also be copied to the folder containing the mock-up pages.

Dreamweaver contains a number of bugs related to hotspot manipulation. Hotspots will occasionally disappear when moved or resized. If this happens, press the F5 key on your keyboard to refresh the page. For finer control of hotspot placement, use the arrow keys to move the currently selected hotspot.

2.5 Example mock-up

This section describes the creation of a mock-up for a car radio device for the purpose of modelling a task in which the user selects track 5 of the second CD in his car's CD changer.

First, the "FirstPage.html" file and the BehaviorRecorderApplet.class file are copied into a new folder. The "FirstPage.html" file is opened in Dreamweaver.

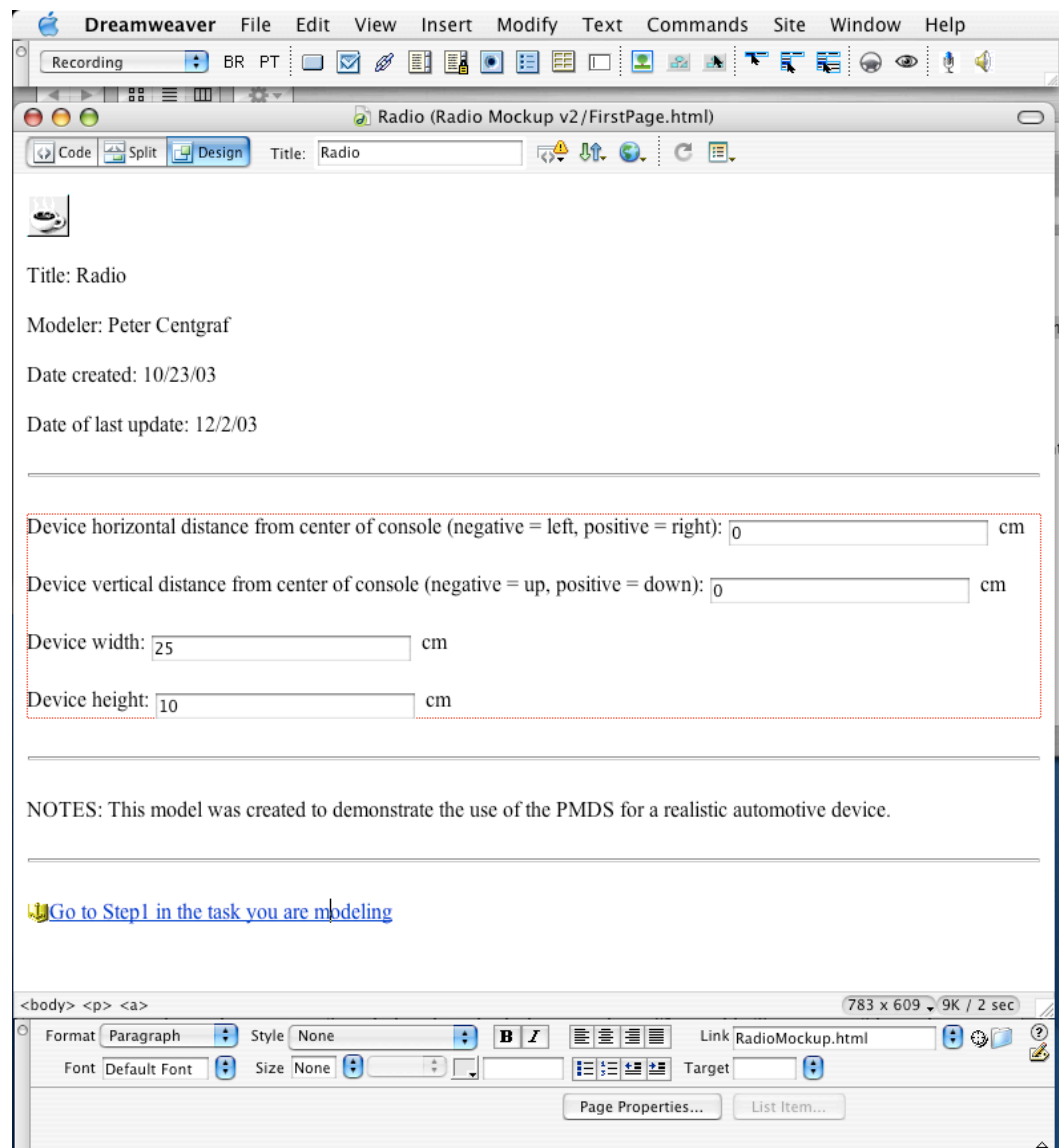


Figure 5 FirstPage.html for Radio Mock-up

The name of the mock-up ("Radio") is inserted in the Document Title field, and the fields at the top of the page are filled out. The device's width and height are measured and inserted in the appropriate fields. Since the

device will be in the center of the console, the distance fields are left at 0. Some notes are inserted in the NOTES section, and the "Link" field in the Properties window of the link on the bottom of the page is modified to "RadioMockup.html", which will be the filename of the second page of the mock-up.

A new document is created in Dreamweaver. The "BR" button is clicked, inserting a coffee cup icon at the top of the page. A file containing an image of the radio s interface is copied to the mock-up folder, and the image tool is used to insert it into the page.

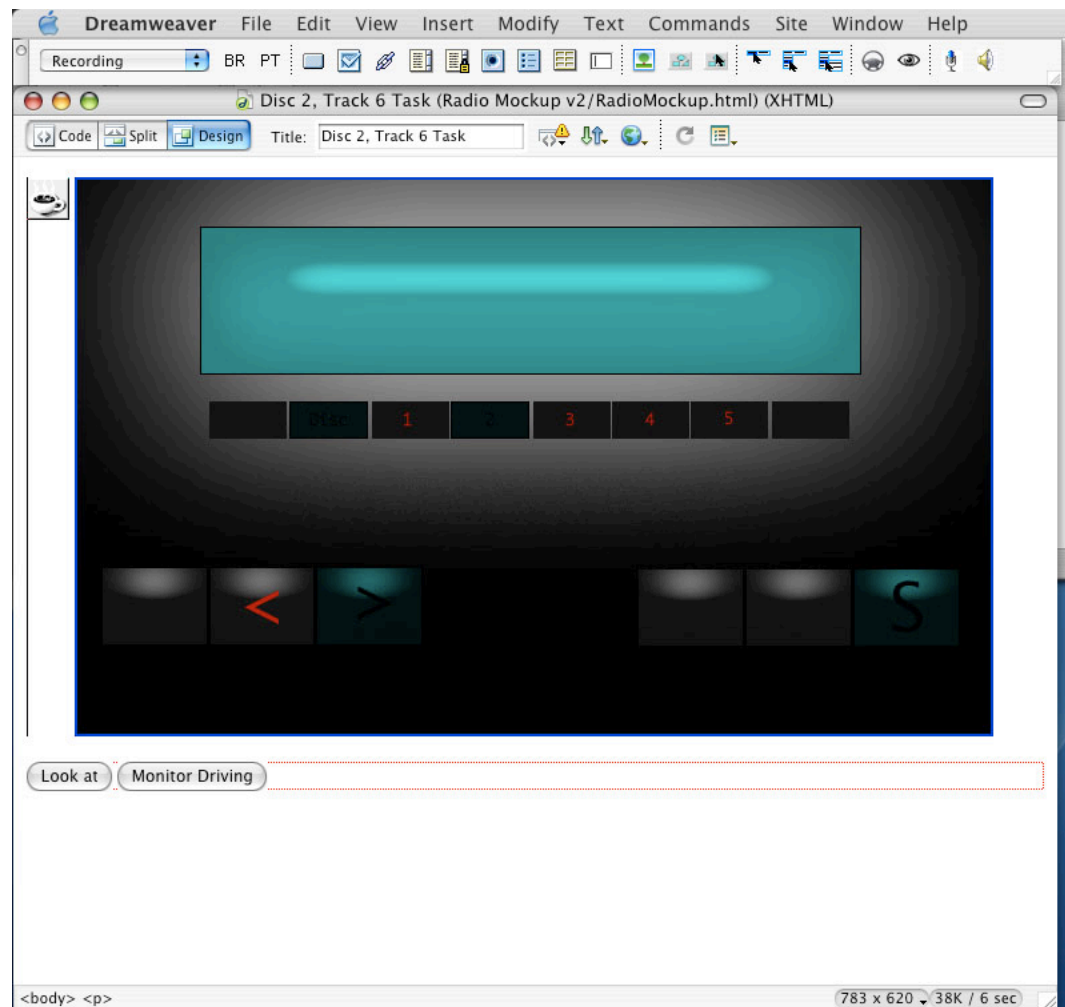


Figure 6 First image page of Radio Mock-up storyboard

Five hotspots are created with the hotspot tool - one for the display, and one for each of the buttons used in the task ("Source", "Disc", "two", and "Next"). The hotspots are named appropriately.

The "Look At" and "Monitor Driving" buttons are inserted at the bottom of the page to allow the modeler to indicate when the user looks at the display, and when the user monitors driving conditions during the task.

This document is saved as "RadioMockup.html", which completes the mock-up.

2.6 Guidelines

- When mocking up a physical device, use hotspots instead of buttons, checkboxes, radio buttons, etc. These widgets can not be resized to match the physical layout of the device. In addition, using hotspots allows you to simulate the effect of a button press by linking the hotspot to a new page.
- Lighter colored images work better because hotspots in Dreamweaver can be hard to see against a dark background.
- Slider bars, twist knobs, and other controls requiring unusual interaction styles can not be mocked up.

3. Create a model by demonstration

3.1 Introduction

Once the instrumented web pages are created in Dreamweaver, tasks can be demonstrated on these pages by opening the Behavior Recorder and the first HTML page in Netscape, and demonstrating the task with mouse movement, clicks, and typing. The web pages use HTML event handlers to send messages to the Behavior Recorder via the LiveConnect feature supported by Netscape.

As the user's actions are sent the Behavior Recorder creates a state-transition diagram, where the state of the webpage is a node and the demonstrated actions are the transitions between nodes. Once a correct procedure for a task is demonstrated, the designer uses the Export item in the File menu to create a file containing ACT-Simple code. The designer has the option of declaring that the mock-up is of a computer-based system where mouse pointing and clicking is a valid interaction technique, or of a physical system where mouse operations are a stand-in for actual physical operations on pressing real buttons (e.g., the HTML represents a cell phone, flight management system, or automobile navigation system). The Export function creates appropriate ACT-Simple code given the choice of this option, e.g., it includes mouse clicks if the mock-up is of a computer-based system, but does not include them for a mock-up of a physical device.

3.2 How to demonstrate a task on a mock-up

- 1) Open the Behavior Recorder
A Window titled Behavior Recorder for file untitled appears with the Demonstrate Mode radio button selected.
- 2) Open Netscape 7.0 - it must be able to connect to the Behavior Recorder through a network connection, either active or inactive.
- 3) Open the FirstPage.html page of the desired mock-up in Netscape by selecting File>Open File... A rectangle will appear in the Behavior Recorder window, with the document title of the FirstPage.html inside it.

This is called the start state.

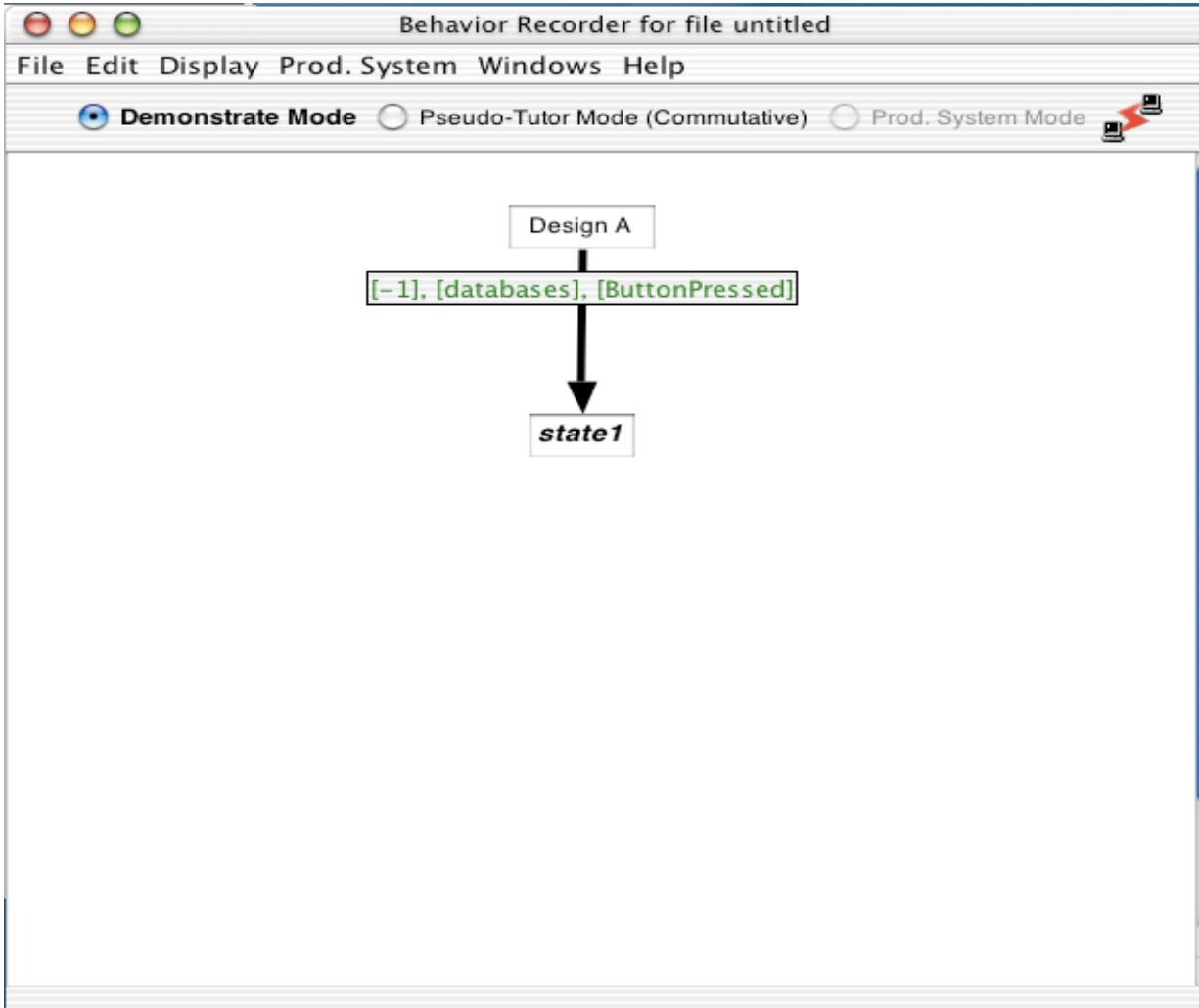


Figure 7 State diagram after first action



Each subsequent action on the mock-up in Netscape will create new states in the Behavior Recorder connected by arrows, with the action that brought about the change in state shown in a label on the arrow.

- 4) Click on the [Go to Step 1 in the task you are modeling](#) hyperlink. This will insert four new states as the sizing and position settings for the mock-up are communicated to the Behavior Recorder.
- 5) The second page of the mock-up will appear. Demonstrate the task as the user would perform it. To designate a point in the task where the user would monitor driving conditions, click the Monitor Driving button you inserted in the mock-up. To indicate that a user would simply look at

an interface element without clicking on it, click the Look At button, then the interface element.

- 6) When the task has been correctly demonstrated, after the last task action, immediately export the demonstration file as described in the next section. Do not perform any further actions in Netscape, since all actions will be recorded by the Behavior Recorder.



Demo CORRECTLY. Start over if you make a mistake!

3.3 Exporting demo file - Creating a task model file

Once you've finished demonstrating the task, select Export... from the File menu in the Behavior Recorder.

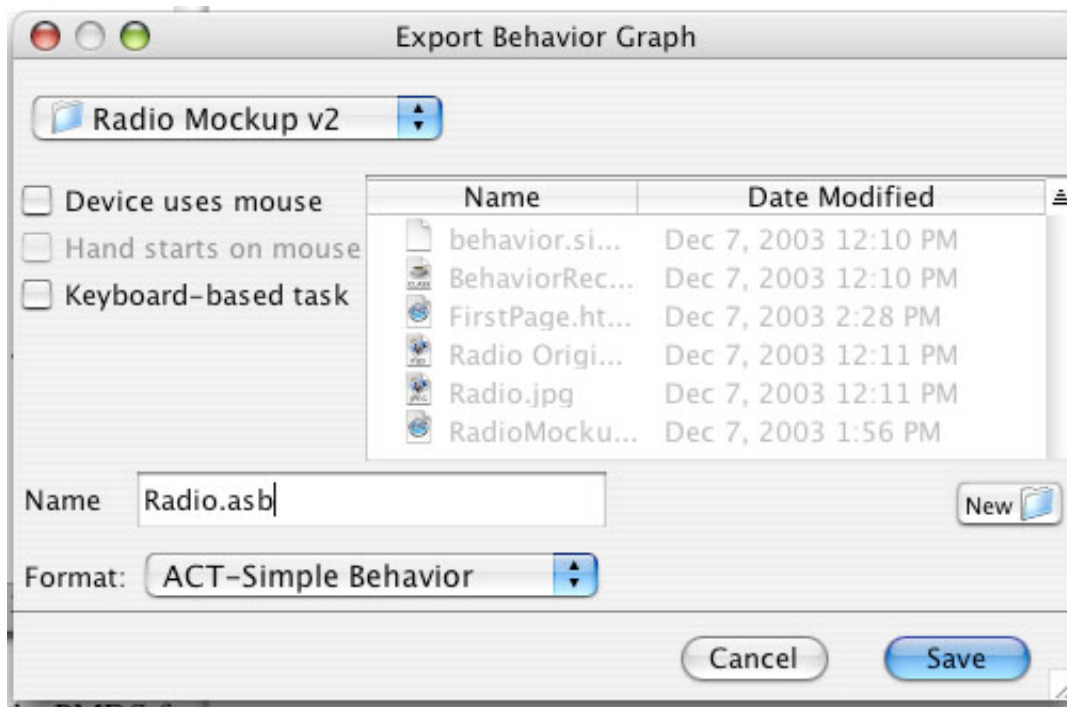


Figure 8 Export Behavior Graph dialog box.

Modify the options if necessary, and enter a name for the file. This will save the task as an ACT-Simple model, which you can view in a text editor if you wish. To help recognize files exported by the Behavior Recorder we've adopted the.asb extension, but this extension is not required.

3.4 Export file options

Device uses mouse: check this box if the interface you are mocking up is manipulated with a mouse or similar pointing and clicking device. Standard desktop computer interfaces should use this setting; devices using a touch-screen or physical buttons should not.

Hand starts on mouse: if the above setting is checked, checking this setting instructs the model to start with its right hand on the mouse. If it is not checked, the model will start with its right hand on the keyboard, and the time it takes the model to move its hand to the mouse will be included in the task execution time.

Keyboard-based task: check this box if the interface you are mocking up is desktop computer-based and the keyboard, rather than the mouse, is the primary means of controlling the interface (e.g., Unix-style text editors).

3.5 The model file

The task model exported by the Behavior Recorder is in the ACT-Simple format, which can be read into the ACT-R cognitive architecture. For a higher level of control over the task model, ACT-Simple can be edited by hand.

In the previous chapter, a mock-up of a radio interface was constructed. Consider the task of selecting the fifth track of the second CD in the car's CD changer using this interface. The steps in this task are:

- 1) Press the source button.
- 2) Press the disc button.
- 3) Press the 2 button.
- 4) Look at the display.
- 5) Press the next button 5 times.
- 6) Look at the display.

The ACT-Simple code exported by the Behavior Recorder is shown below with comments after each section of the code.

```
(set *mouse-clicks* nil)
```

This first line contains the `Device uses mouse` setting in the Export dialog. This line should not be changed by hand.

:

```
(if (fboundp 'create-task-item)
    (setf *task2-code*
        '(progn
```

This section indicates the start of the driving + task model. See Chapter 5, *Executing a Driving + Task Model*. This section must not be edited.

```
(create-task-items '(
    (Next-item Next 534 392 20 9)
    (Source-item Source 637 392 20 9)
    (display-item display 565 357 125 17)
    (two-item two 557 371 14 4)
    (Disc-item Disc 527 371 14 4)
    ))
```

This section lists the interface elements used in the task. This section should not be edited by hand.

```
(klm-goal (task2 drive)
  (move-hand device pop)
  (think)
  (look-at "Source" )
  (press-button "Source" )
  (think)
  (look-at "Disc" )
  (press-button "Disc" )
  (think)
  (look-at "two" )
  (press-button "two" )
  (look-at display )
  (think)
  (look-at "Next" )
  (press-button "Next")
  (think)
  (press-button)
  (think)
  (press-button)
  (think)
  (press-button)
  (think))
```

:

```
(press-button)
)))
```

This is the driving + task model. Each line after the first specifies an action for the model to take. `(move-hand device pop)` tells the model to move its hand from the steering wheel to the device. `(think)` is an action inserted automatically by the Behavior Recorder to account for mental processes taken by the user during the task. `(look-at)` actions are inserted before each button press. The other actions correspond to steps in the demonstration. The second through fifth presses of the Next button are indicated simply with `(press-button)`, which tells the model to press the last button it pressed again.

This part of the model may be edited by hand. Steps may be removed or inserted, as long as any interface elements are specified in the previous section.

```
(klm-p klm (klm-goal klm
(think)
(look-at "Source")
(press-button "Source")
(think)
(look-at "Disc")
(press-button "Disc")
(think)
(look-at "two")
(press-button "two")
(look-at display)
(think)
(look-at "Next")
(press-button "Next")
(think)
(press-button)
(think)
(press-button)
(think)
(press-button)
(think)
(press-button)
)))
```


:

This is the stand-alone task model. See Chapter 4, *Executing a stand-alone task model*. This section functions identically to the driving + task model, except that the names of interface elements must be enclosed in quotation marks, and it may be edited by hand as described above.

```
(pm-start-hand-at-mouse)
(pm-set-params :output-speech t)
```

This section of the model sets some ACT-R parameters and should not be edited by hand.

3.6 Editing the model file

The placement of `(think)` commands by the Behavior Recorder is optimized for desktop software interfaces. It may not be appropriate for other types of interfaces. Fortunately, ACT-Simple's syntax allows for easy manipulation of these commands.

In the model above, the five consecutive presses of the `Next` button are each preceded by a `(think)` command. This is probably not appropriate, so the `(think)` commands before the second through fifth button presses can simply be removed.

The `(think)` command, by default, takes the model 1200 milliseconds to execute, which is an appropriate time for desktop software interfaces. The length of a `(think)` can be changed by specifying a new length in seconds, e.g. `(think .6)` for a 600 millisecond command.

3.6.1 Exploring the effects of monitoring driving

The driving + task model includes `pop` commands where the modeler used the `Monitor Driving` button to indicate a point at which the user would monitor driving conditions. Pops can also be inserted manually by typing `pop` after any command, before the command's closing parenthesis. For example, `(press-button Source)` becomes `(press-button Source pop)`.

Inserting pops by hand allows the modeler to quickly explore the effects of different strategies for monitoring driving. For instance, what if the user monitors driving after every action? After each of the task's subtasks? Only after the entire task is complete? By editing the pops manually, the modeler does not have to demonstrate the task again for each of these scenarios.

:

4. Executing a Stand-alone Task Model

4.1 Introduction

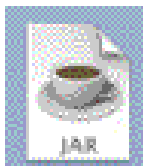
This chapter will describe the procedure for running a stand-alone task model. Stand-alone task models simulate a task that is performed outside of any particular usage context. The result measure produced by a stand-alone model is the total task time. Performance of the task is simulated using Behavior Recorder and Netscape, allowing the modeler to observe the actions of the model in real time.



Netscape 7.0 opens the Mac OS 9 Classic Environment and must have an active network connection in order to operate. The network connection may be vestigial merely connecting to a hub is sufficient for communication between Netscape and Behavior Recorder to function.

4.2 Preparing to Execute the Model

In order for the PMDS to execute a task model, several of the component applications need to be prepared.



- 1) Open the Behavior Recorder by clicking the "BehaviorRecorder.jar" icon in the dock. (You may also navigate to the "Macintosh HD>Applications>Behavior Recorder" folder and double-click the BehaviorRecorder.jar icon there.)

When this step is complete, a blank Behavior Recorder window will appear, as shown below.

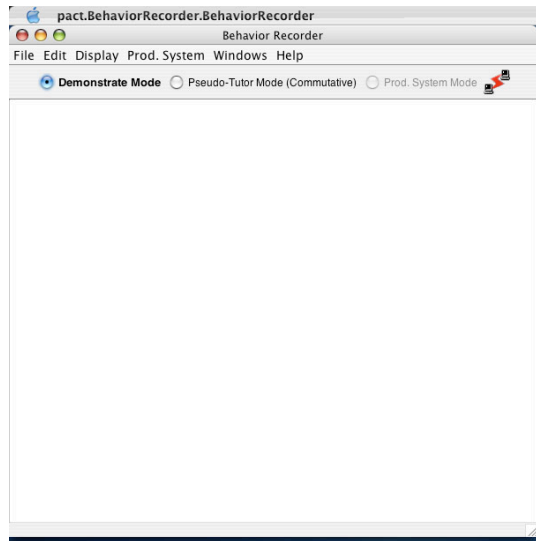
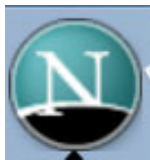


Figure 9 Blank Behavior Recorder window.



- 2) Open the "FirstPage.html" for your model in Netscape. You may either drag the file's icon to the Netscape icon on the dock or use the "File>Open File..." menu option within Netscape.

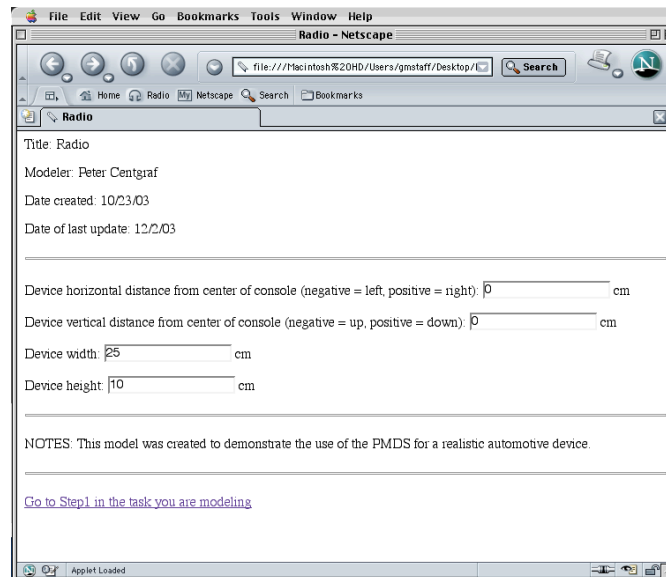


Figure 10 FirstPage.html for the CD player example loaded in Netscape.

After the "FirstPage.html" has loaded, the Behavior Recorder should display the title of your mock-up in its main window, as shown below.

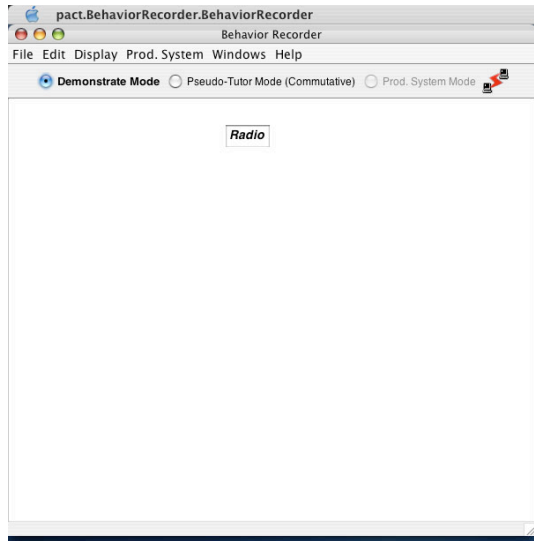


Figure 11 Behavior Recorder with the model title displayed.

If the title does not appear, there may be a problem establishing a connection between Netscape and the Behavior Recorder. Make sure that the mock-up contains the "BehaviorRecorderApplet.class" file, and that a network connection of some kind is available on the computer.

Note: The Behavior Recorder must be open before the mock-up is loaded in order for the connection to be made. If "FirstPage.html" was opened before the Behavior Recorder, simply refresh the browser window to establish the connection.

- 3) Click on the link at the bottom of the page, labeled with the text "Go to Step1 in the task you are modeling". The Behavior Recorder will register the size information from the model as states in the diagram.

:

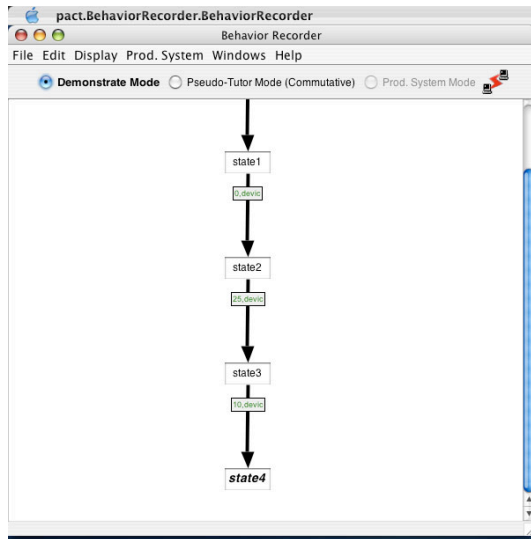


Figure 12 Behavior Recorder window with sizing information.



- 4) Open the Model Launcher by double-clicking its icon in the dock.
- 5) Press the "Stand-alone Task Model" button to load the support for this type of model.

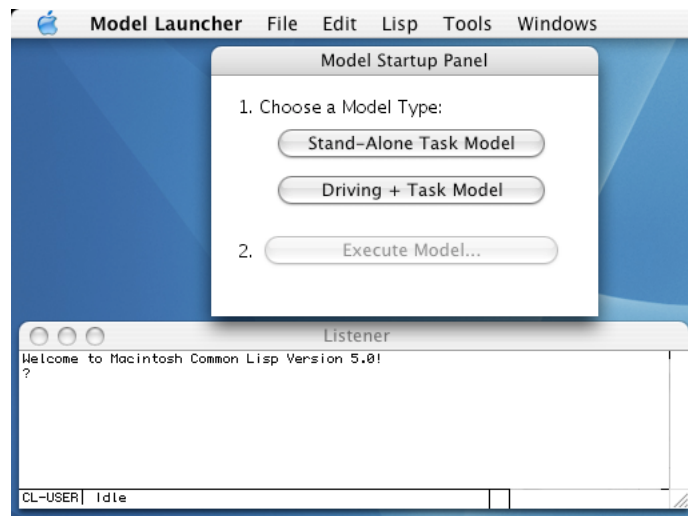


Figure 13 Initial Model Launcher windows.

The Model Launcher will load the necessary support files and display output to a new "Listener 1" window. This output is related to the ACT-R cognitive architecture implementation and may be ignored. When preparation has finished, the "Execute Model..." button will become active, as shown below.

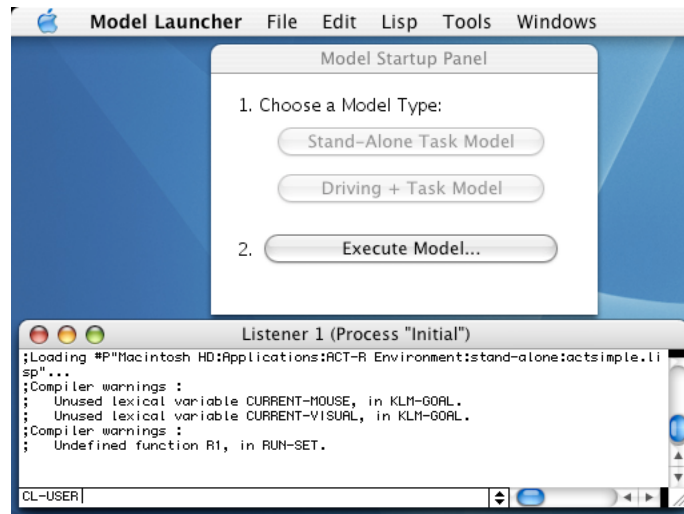


Figure 14 Model Launcher after preparing to execute a Stand-alone model.

The PMDS is now ready to execute a task model.

4.3 Execute the Task Model

To execute a task model and generate the total task time, follow these steps:

- 1) Make sure that the Behavior Recorder has a connection to the mock-up, and Netscape is displaying the first step of the task storyboard. If you are not sure that the Behavior Recorder is ready, reset to a new diagram by using the File>New menu item and clicking "No" when prompted to save. Finally, reload the page containing the first step of the task storyboard. The Behavior Recorder window will display the title of the model.
- 2) Press the "Execute Model..." button. The "Choose a File" dialog box will appear.

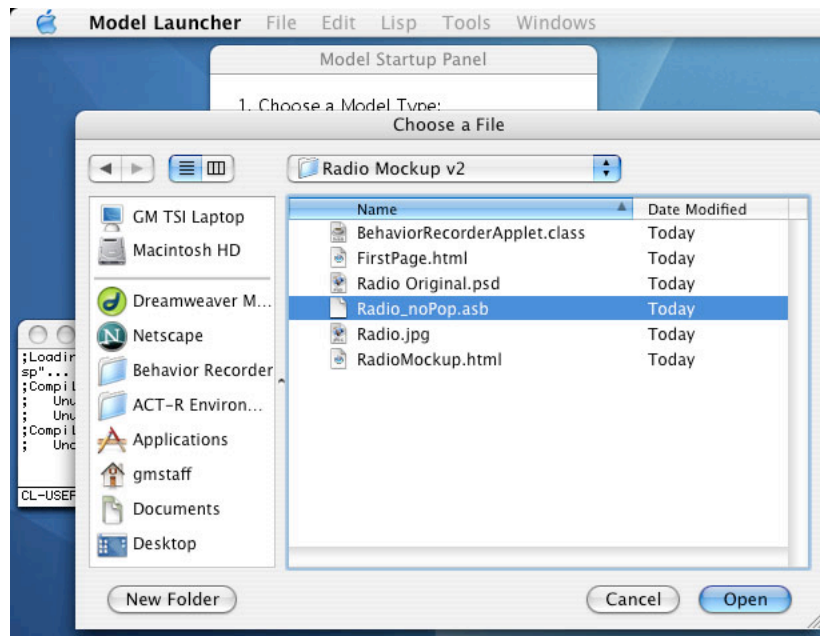


Figure 15 Choose a File dialog in Model Launcher.

- 3) Navigate to the task model file that was exported from the Behavior Recorder earlier. The convention for this type of file is to use an "asb" file extension.
- 4) Double-click the task model file to select it. The Model Launcher application will execute the task model using the ACT-R cognitive architecture. The output displayed in the "Listener 1" window describes the steps of execution carried out by the ACT-R architecture. You may safely ignore this unless detailed timeline data is required.

While the task is being executed, the Behavior Recorder will simulate actions within Netscape. The modeler can observe the actions by watching for page loading within Netscape and changes in the state diagram in the Behavior Recorder window.

- 5) When the task execution has completed, a "Results" window will appear, containing the total task time in seconds. The time may be copied for reuse elsewhere by using the "Copy to Clipboard" button, the Edit>Copy menu item, or the Command-C keyboard shortcut.

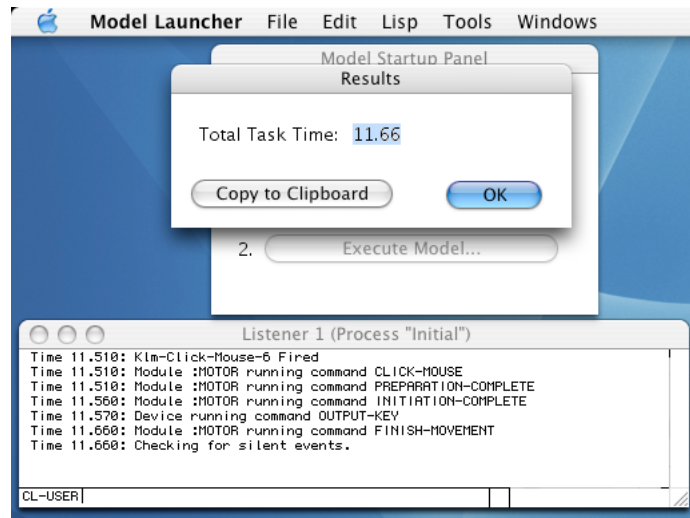


Figure 16 Results dialog for a stand-alone task in Model Launcher.

- 6) Press the "OK" button to dismiss the window, so that more models can be executed.
- 7) When finished executing models, quit the Model Launcher using the "File>Quit" menu item or the Command-Q keyboard shortcut.

:

5. Running a driving plus task model

5.1 Introduction

This chapter will describe the procedure for running a task model with a driving model. This allows the modeler to examine the effects of performing a task while driving on driving performance, and vice versa. Unlike a stand-alone model, the driving plus task model is executed in a special driving environment created by the Model Launcher, so the Behavior Recorder and Netscape are not required to be open. The output of a driving plus task model includes several measures of both task performance and driving performance. In addition, the model is required to react to the brake lights of a lead car, and data on the model's reaction time before and during the task is collected.

5.2 Preparing to Execute the Model



- 1) Open the Model Launcher by double-clicking its icon in the dock.
- 2) Press the "Driving + Task Model" button to load the support for this type of model.

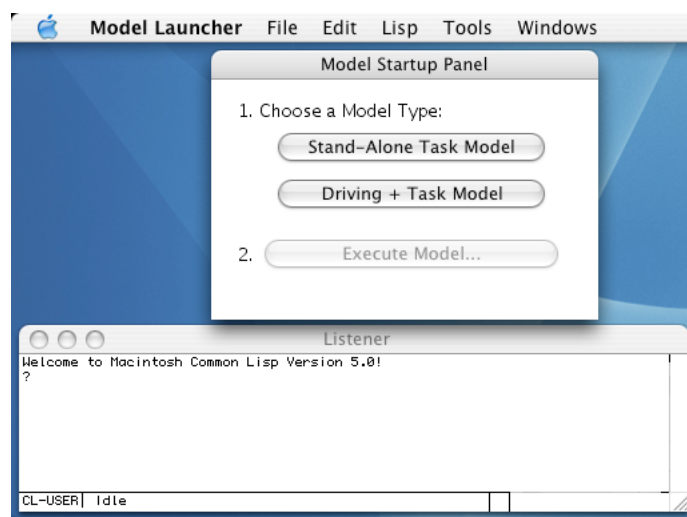


Figure 17 Initial Model Launcher windows.

The Model Launcher will load the necessary support files and display output to a new "Listener 1" window. This output is related to the ACT-R cognitive architecture implementation and may be ignored. When preparation has finished, the "Execute Model..." button will become active, as shown below.

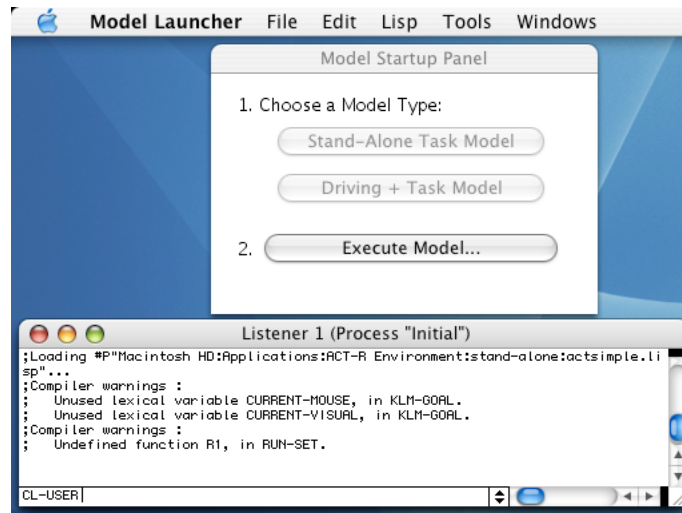


Figure 18 Model Launcher after preparing to execute a Driving model.

The PMDS is now ready to execute a task model.

5.3 Execute the Task Model

To execute a driving plus task model and generate the analyses, follow these steps:

- 1) Press the "Execute Model..." button. The "Choose a File" dialog box will appear.

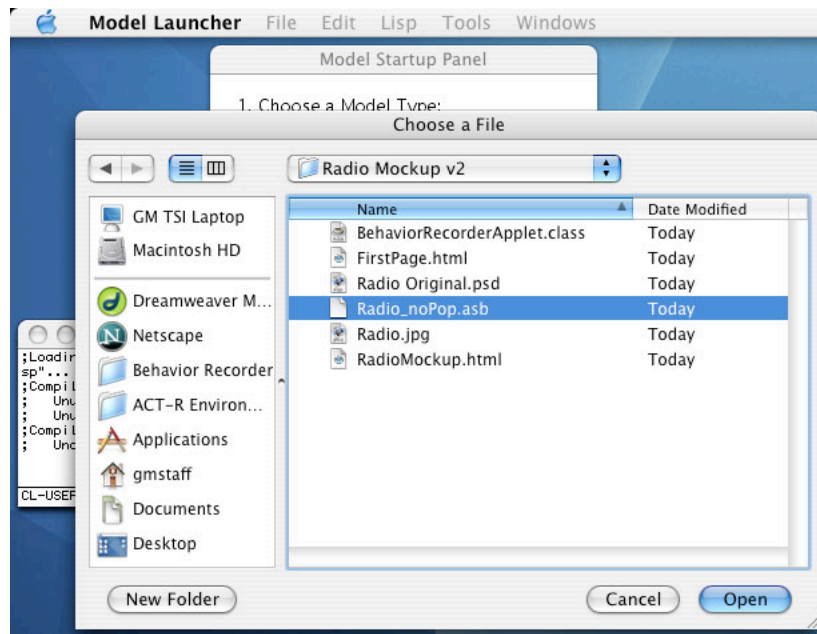


Figure 19 Choose a File dialog in Model Launcher.

- 2) Navigate to the task model file that was exported from the Behavior Recorder earlier. The convention for this type of file is to use an "asb" file extension.
- 3) Double-click the task model file to select it.

The Model Launcher application will open a new window to display the driving environment, which will include the relevant interface elements from the task interface. It will first execute the task model several times without driving to generate a baseline. The driving environment will then be closed and re-opened, and the model will drive for approximately 40 seconds to generate baseline driving performance data. After 40 seconds, the model will execute the task several times while driving. The output displayed in the "Listener 1" window describes the steps of execution carried out by the ACT-R architecture. You may safely ignore this unless detailed timeline data is required.

The driving environment includes several indicators of the model's activity. A green circle is displayed to represent the position of the model's right hand. This circle starts on the box in the center of the window, which represents the right side of the steering wheel. Once the model begins to perform the task, this circle will move accordingly. A red ring and a blue circle are used to show where the model is looking. A white circle represents the position of the lead car.

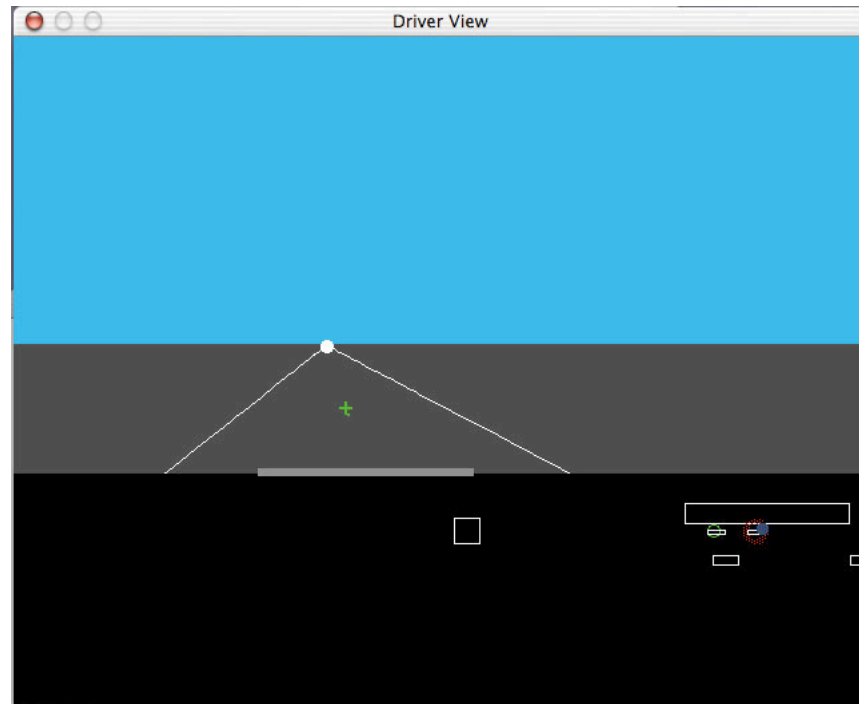


Figure 20 The driving simulation window.

- 4) When the task execution has completed, the Model Launcher compiles the data for the following measures:
- Average task time in seconds, for both driving and non-driving conditions
 - Average lateral deviation in meters, both before and during task
 - Average braking reaction time, both before and during task
 - Average number of glances to device per task
 - Average glance duration

These measures are displayed in the "Listener 1" window, and can be copied for reuse elsewhere by using the Edit>Copy menu item or the Command-C keyboard shortcut.

Listener 1 (Process "Initial")

Analyzing...

Task Time:

Subject	NoDrive	Drive
Model	10.7500	18.5500

Lateral Deviation:

Subject	NoTask	Task
Model	0.3453	0.6051

Brake Reaction Time:

Subject	NoTask	Task
Model	0.8750	1.3000

Glance Number:

Subject	NoTask	Task
Model	0.0000	4.0000

Glance Duration, Device:

Subject	NoTask	Task
Model	NIL	0.4375

Glance Duration, Forward:

Subject	NoTask	Task
Model	NIL	1.3000

CL-USER |

Figure 21 Sample output from a Driving + Task Model.

- 5) When finished executing models, quit the Model Launcher using the "File>Quit" menu item or the Command-Q keyboard shortcut.

: